(19)
Bundesrepublik Deutschland
Deutsches Patent- und Markenamt

(10) **DE 10 2006 045 224 A1** 2008.01.24

(12) # Offenlegungsschrift

(21) Aktenzeichen: **10 2006 045 224.0**
(22) Anmeldetag: **26.09.2006**
(43) Offenlegungstag: **24.01.2008**

(51) Int Cl.[8]: ***H04L 9/32*** (2006.01)

(71) Anmelder:
dl-DATEN GmbH, 30982 Pattensen, DE

(72) Erfinder:
Werner, Herbert-Heinz Maria, 30982 Pattensen, DE; Cwienk, Georg-Viktor, Dr., 44797 Bochum, DE

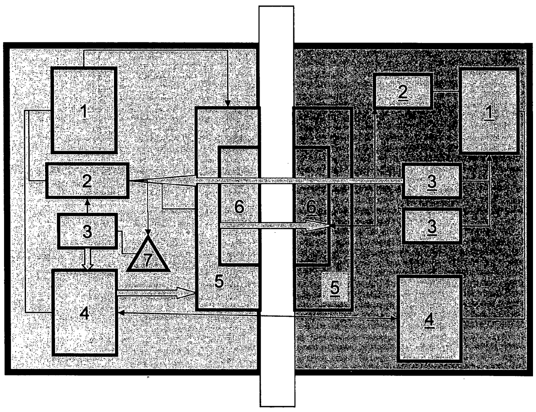
(56) Für die Beurteilung der Patentfähigkeit in Betracht gezogene Druckschriften:
US2003/00 76 960 A1
US2002/01 86 837 A1
WO 01/08 350 A1

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

Prüfungsantrag gemäß § 44 PatG ist gestellt.

(54) Bezeichnung: **Verfahren für eine hochgradige verbindungsorientierte Authentisierung und Überprüfung der Integrität der Nutzerdatenübertragung (SynD)**

(57) Zusammenfassung: Das nachfolgend beschriebene Verfahren beschreibt die Authentisierung und Authentifizierung zwischen zwei oder mehreren Partnern, zwischen einer oder mehreren Hierarchien und damit abgestuften Rechten.
Auf eine einfache Weise wird dokumentiert, dass die für die Authentisierung und Authentifizierung genutzten Verfahren ebenso für eine Contentverschlüsselung genutzt werden, die mit gängigen statistischen Methoden oder anderen Dechiffrierverfahren nicht zu entschlüsseln sind, und zugleich die Verknüpfung erkannter und grundlegend neuer Zerlegungsmechanismen der Primzahlwelten nutzt.
Der in der Content-Security vorgehaltene Dechiffrierraum an statischen Tabellen zuvor generierte gesetzmäßig gebildeter Zerlegung hat den Charakter eines komplexen Buchschlüssels und setzt die Kenntnis beider kommunizierender Partner voraus.



Beschreibung

[0001] Das nachfolgend beschriebene Verfahren beschreibt die Authentisierung und Authentifizierung zwischen zwei oder mehreren Partnern, zwischen einer oder mehreren Hierarchien und damit abgestuften Rechten.

[0002] Auf eine einfache Weise wird dokumentiert, dass die für die Authentisierung und Authentifizierung genutzten Verfahren ebenso für eine Contentverschlüsselung genutzt werden, die mit gängigen statistischen Methoden oder anderen Dechiffrierverfahren nicht zu entschlüsseln sind, und zugleich die Verknüpfung erkannter und grundlegend neuer Zerlegungsmechanismen der Primzahlwelten nutzt.

[0003] Der in der Content-Security vorgehaltene Dechiffrierraum an statischen Tabellen zuvor generierter gesetzmäßig gebildeter Zerlegung hat den Charakter eines komplexen Buchschlüssels und setzt die Kenntnis beider kommunizierender Partner voraus.

Das Implementierungsverfahren in der Hardware-Implementierung

[0004] Das Verfahren nutzt bislang nicht bekannte Kryptomethoden, die sich aus der Zerlegung beliebig großer Zahlenräume von Primzahlen bzw. nicht primen Zahlen ergeben.

[0005] Beschreibung Fig. 1:
Quelle (A) und Senke (B) werden durch einen Kommunikationsbereich von einander abgetrennt.

[0006] Quelle-Komponenten werden mit arabischen Zahlen dargestellt, Senke-Komponenten werden mit unterstrichenen arabischen Zahlen abgebildet.

[0007] Die Hauptprogrammeinheit = CPU (**1**) kommuniziert permanent mit dem flüchtigen Speicher (**2**) und dem Permanentspeicher (**4**) in dem Anwendungen vorgehalten werden und bei Bedarf in (**2**) hochgeladen werden.

[0008] Ein programmierbarer Baustein (**3**) kommuniziert bei Bedarf mit (**2**) und lädt das Programm SynD hoch, zugleich veranlasst es (**4**) Programme auf der Netzwerkfilterebene zu aktivieren und mit der Software-Einbettung der CPU (zentrale Prozessoreinheit), das BIOS (Binäres Eingabe- und Ausgabesystem) (**5**) über Netzwerkinterfacekarten (**6**) zu kommunizieren. Die Gegenstelle (<u>**6**</u>) gibt an (<u>**2**</u>) und veranlasst (<u>**3**</u>) das Programm SynD zu starten, die generierte Primzahl/nicht prime Zahl wird durch ein eigenes Programm in (<u>**3**</u>) in einen Internet Kontroll Management Programm (ICMP)-Stack als Pattern (Muster) eingeschrieben und von (<u>**6**</u>) zu (**6**) gesendet. Ein Unterprogramm (**7**) von SynD wird von (**1**) gestartet und führt zu der Prüfung, ob die Senke autorisiert ist. Die unter (<u>**3**</u>) generierte Primzahl/nicht prime Zahl wird mit (**4**) gegengelesen und akzeptiert oder verworfen. Wird die Authentisierung akzeptiert, so gilt für diese Senke und als Markierung als überprüft angenommene Information = Acknowledge (ACK) unter Transport Kontroll Protokoll (TCP) informiert und ein Sicherheitstunnel (VPN, virtueller Netzwerkpfad) aufgebaut. Wird von der Senke keine Primzahl/nicht prime Zahl via Pattern und ICMP erkannt (identifiziert), dann bleibt diese Stelle rufbereit in einer (ihrer) Warteschleife und die ICMP-Anforderung = ICMP-Request wird als Standard-Ping vom Betriebssystem weiter gesendet, bis die eingetragene Wiederholung beendet ist. Nach außen hin wirkt dieser ICMP-Request wie ein Standard Echo-Request. Die Senke (B), die die gesandte prime bzw. nicht prime Zahl erkennt, ist aktiviert und sendet rollierend die komplementäre prime bzw. nicht prime Zahl an die Quelle (A).

[0009] In dieser Figur (Fig. 2) wird beispielhaft eine Portierung der Grundfunktionalitäten nach verketteten Goldbachschen Zerlegungen dargestellt.

[0010] Die Vielzahl von Anwendungsmöglichkeiten, insbesondere die Anwendung im Rahmen mobiler Kommunikationseinrichtungen, lässt sich daraus paradigmatisch ableiten.

[0011] Fig. 2 beschreibt prinzipiell den primären Austausch von Informationen, der durch das Prinzip der synchron rollierenden Verschlüsselung ergänzt wird.

[0012] (A) und (B) stellen zwei Computersysteme dar, die über das öffentliche Netz miteinander kommunizieren. (B) empfängt die Anfrage von (A) und antwortet mit einem „alive". Dadurch weiß (A), dass die Kommunikationsstrecke über ICMP und später über TCP aufbaubar ist.

[0013] Prozess (a) führt nach Feststellung, dass (B) mittels ICMP erreichbar ist, dazu, dass Prozess (b) bei der Senke – einem beliebigen Client, der sich gegenüber der Quelle (A) authentisieren muss, aufgerufen wird. Dieser Prozess (b) löst die Rücksendung einer Identifizierungszahl an (A) aus.

[0014] Der Initialprozess wird per Hand oder automatisch von (A) ausgelöst. Dabei sendet (A) an die zu authentifizierende Stelle (B) eine Ziffernkombination. Diese Ziffernkombination wurde für den Hauptprozess (d) (SynD) ermittelt und in einem Zentralspeicher gesichert abgelegt. Die Senke (C), in der diese Zahl/Zahlen nicht gespeichert ist/sind, wird durch die/diese Zahl/Zahlen nicht angesprochen (aktiviert). Diese Zahlen und weitere Zahlen, die in der Senke gespeichert sind, können z.B. über ICMP-Tunnel gesichert übertragen werden. Durch den Algorithmus SynD werden zwei Zahlen generiert, die gültig sind und nach der Goldbachschen oder Anti-Goldbachschen Zerlegung gebildet wurden. [MNda-Cwienk2006]

[0015] Die als Zahlenwerte Z_1 ... Z_n implementierten Zahlen sind Ausgangspunkt für das Prinzip der synchron rollierenden Verschlüsselung (siehe dazu Fig. 3) und können auch in Zeitabständen geändert werden.

[0016] Diese Zahlen werden von (B) an (A) und vice versa gesendet. Die Bildung dieser Zahl geschieht im Mikrosekunden-Bereich und hat keinen Einfluss auf die Performance eines Gesamtsystems. Nachdem (A) diese Zahl von (B) erhalten hat, gilt (B) als authentisiert und vice versa. Zur anschließenden Übertragung eines Nutzdaten-Blocks wird eine mit der Authentisierung gekoppelte Zahl (x) vorgeschaltet, der nach Ende des Datenblocks (DBL) eine gekoppelte Zahl (y) folgt.

[0017] Dieses Prozedere kann beliebig oft (nutzerabhängig) fortgesetzt werden (siehe das Beispiel im NDA-Teil).

[0018] Nach demselben Prinzip können (geeignete) Nutzdaten-Übertragungen von einzelnen Senken (sogar zeitgleich) an die Quelle erfolgen.

[0019] Die für Fig. 3 beispielhaft genutzten Zahlenwerte (die implementierten rollierenden) Zahlenwerte Z_1 ... Z_n können in Zeitabständen geändert werden.

[0020] Die Genese dieser Zahlen Z_1 ... Z_n obliegt dem Lizenznehmer, dem eine Menge dieser Zahlen n-Tupels zur Verfügung gestellt wird.

[0021] Hinsichtlich des Algorithmus wird auf die Monographien (unter NDA stehend) hingewiesen. [MNda-Cwienk2006]

Prinzip der synchron rollierenden Veschlüsselung

[0022] Die unter (3) gespeicherte Primzahl/nicht prime Zahl wird an (4) gesandt und akzeptiert oder verworfen. Die Quelle transportiert daraufhin die zyklisch nachfolgende Primzahl/nicht prime Zahl an die Senke. Dort wird die zyklisch folgende Primzahl/nicht prime Zahl Z_3 aufgerufen und darauf wird die Nutzinformation an die Senke übertragen. Der Nutzinformation wird eine „gekoppelte" Zahl GZW 1 (x1) vorangestellt und das Ende dieser Nutzinformation wird durch die gekoppelte Zahl GZW 2 (y2) vermerkt. Die Senke quittiert den Empfang dieses Nutzteilblocks mit Sendung der zyklisch nachfolgenden Primzahl/nicht prime Zahl Z_4 an die Quelle, woraufhin die Quelle die nachfolgende Zahl (prime oder nicht prime Zahl) Z_5 generiert und an die Senke damit verbunden den zweiten Nutzteilblock sendet. Der Empfang dieses Blocks wird durch Sendung der Zahl Z_6 bestätigt. Dieser Zyklus kann in nachfolgenden Schritten analog fortgesetzt werden, sofern der Nutzer eine derart hochgradige Verschlüsselung als notwendig fordert.

Prinzipzusammenfassung

(1) A sendet ICMP und Z_1 als Pattern an B.
(2) B empfängt Z_1 und rolliert sofort durch die mittels ICMP an A gesendete Zahl Z_2.
(3) A empfängt Z_2, rolliert auf Z_3 und sendet Nutzdaten über TOP, die Nutzdaten werden in Blöcke fragmentiert und markiert.
(4) B empfängt Nutzdaten Block 1 bis zur Markierung. Nach Erhalt der Markierung wird auf Z_4 rolliert und das Authentisierungsprozedere neu angestoßen.

Stand der Technik

[0023] Die gängigen Verschlüsselungsverfahren stützen sich im wesentlichen auf den kleinen Fermatschen Satz und die Primzahlfaktorzerlegung, die mit RSA initiiert worden war. Versatz, Ersatz, Buchschlüssel und andere gängige Methoden zur Umwandlung von Klartextverfahren in Chiffre haben stets dort die Grenze, wo Übertragungsverfahren ohne Authentisierung und Blocktextverschlüsselung erfolgen.

[0024] In Tabelle 1 werden die letzten Patente, die beim DPMA angemeldet wurden, angeführt. Im Schwerpunkt modifizieren diese Patente die bereits gängigen Schlüsselalgorithmen und -verfahren.

Tabelle 1: Die letzten 10 Patentanmeldungen auf dem Gebiet der Datenverschlüsselung (Stand: 13.09.2006)

Nr.	Veröffentlichungs-Nummer Titel
1	AT000000333732E [DE] VERSCHLÜSSELUNG MITTELS ÖFFENTLICHEM SCHLÜSSEL MIT EINEM DIGITALEN ...
2	AT000000317579E [DE] VERFAHREN ZUR AUSFÜHRUNG EINES VERSCHLÜSSELUNGSPROGRAMMS ZUR VERSCHLÜSSELUNG ...
3	AT000000316313E [DE] DYNAMISCHE REKONFIGURATION EINER VERSCHLÜSSELUNG BEIM FESTSTELLEN ...
4	AT000000314701E [DE] METHODE UND VERFAHREN ZUR FÄLSCHUNGSRESISTENTEN VISUELLEN VERSCHLÜSSELUNG
5	AT000000282915E [DE] VORRICHTUNG ZUR VERSCHLEIERUNG UND VERSCHLÜSSELUNG
6	AT000000281724E [DE] VERFAHREN UND EINRICHTUNG ZUR EINMALIGEN VERSCHLÜSSELUNG EINER VIELZAHL ...
7	AT000000199198E [DE] VERFAHREN ZUR BLOCKWEISEN VERSCHLÜSSELUNG/ENTSCHLÜSSELUNG UNTER ...
8	AT000000185035E [DE] VERSCHLUESSELUNG UND ENTSCHLUESSELUNG VON MULTIMEDIADATEN
9	DE000006928982U [DE] KAMERA MIT EINER FUEHLEINRICHTUNG ZUM ABFUEHLEN EINER VERSCHLUESSELUNG ...
10	DE000002600635C2 [DE] Verfahren zur Verschlüsselung eines Analogsignals und Übertragungsvorrichtung, ...

[0025] Die Authentifizierung bezeichnet den Vorgang der Überprüfung der Identität eines Gegenübers, z.B. einer Person, eines Computerprogramms oder einer Organisationseinheit (z.B. Firma).

[0026] Die Authentisierung definiert den Vorgang des Nachweises der eigenen Identität.

[0027] Für beide Verfahren gibt es bereits unterschiedliche technische Ausprägungen. Diese Verfahren dominieren das Kreditkartenwesen genauso wie die Herstellung einer gesicherten Verbindung über das Internet (VPN). Zur Integritätssicherung können unterschiedliche Schlüsselverfahren zum Tragen kommen, die sowohl gleiche Schlüssel zum Ver- und Entschlüsseln benutzen (symmetrisch) als auch Verfahren, die unterschiedliche Schlüssel zum Ent- und Verschlüsseln benutzen (asymmetrische Schlüssel).

[0028] Den Verfahren ist das folgende Problem immanent:
Beide miteinander kommunizierende Stellen müssen über Schlüssel verfügen. Die Art der Schlüsselverteilung ist ein kritischer Ansatzpunkt zu deren Offenlegung.
Wird ein öffentlicher Schlüssel genutzt, bleibt die Problematik der Generierung der zugehörigen asymmetrischen Schlüssel zur Öffnung des kryptierten Textes.

[0029] Der gängige Schlüsselalgorythmus RSA beruht auf dem kleinen Fermatschen Theorem (Uhrenrechner mit mod p, wobei p eine Primzahl/nicht prime Zahl ist. Man erhält für jede Zahl, die man p-fach mit sich selbst multipliziert, die Ausgangszahl). Dieses System wurde durch Rivest, Schamir und Adleman weiter entwickelt. Kryptographiesysteme mit öffentlichem Schlüssel (also asymmetrische Systeme) haben Bekannterweise Schwierigkeiten, einen Vorgang mit einer Bit-Zahl auszuführen, die größer als 512 Bit ist. Aufgrund dieser Tatsache wurde zusätzlich eine „wiederholende Kundenstruktur" eingeführt. Unabhängig davon entsteht seit einiger Zeit als scharfer Mitbewerb zum RSA-Schlüssel ein Verfahren, das insbesondere vor dem Hintergrund der Mobile IP Ideen verwertet, wo elliptische Kurven, die durch Gleichung identifiziert werden und auf diese Weise sich dem letzten fermatschen Theorem annähern.

[0030] Mit den Übertragungsprotokollen TCP/IP und der zunehmenden Bedeutung von IPsec des heutigen Internets wurden Standards geschaffen, die die Implementierung von Quelle-Senke-Zuordnungen und Authentisierungsverfahren einerseits ermöglicht, andererseits containerisierte Aufnahmekapazitäten für unterschiedliche Schlüsselverfahren gewährleistet.

[0031] Eine weitere Perspektive bietet das neue Internetprotokoll IP next Generation (IPnG/IPv6). Für die bereits beschriebenen Verfahren wird die Erweiterung der Adressierungsräume und damit einhergehend die Erweiterung des Overheads zur Aufnahme komplexer Schlüsselalgorythmen, seien diese symmetrischer oder asymmetrischer Natur, ermöglicht.

[0032] Das bekannte RADIUS-Protokoll zur Authentisierung und Authentizifierung, im Langtitel heißt es auch Remote Authentication Dial in User Service, ist ein Protokoll nach einem Industriestandard, das im RFC2865 und RFC2866 beschrieben ist. Ein RADIUS-Client, normalerweise ein Default-Client oder VPN-Server oder drahtloser Zugriffspunkt, sendet an Benutzer Anmeldeinformationen und Verbindungsparameter in Form einer RADIUS-Meldung an den RADIUS-Server. Der RADIUS-Server authentifiziert und autorisiert die Anfrage des RADIUS-Clients und sendet die RADIUS-Antwortmeldung zurück. RADIUS-Clients senden außerdem RADIUS-Kontoführungsmeld-ungen an RADIUS-Server. Darüber hinaus unterstützen die RADIUS-Standards die Verwendung von RADIUS-Proxys. Ein RADIUS-Proxy ist der Computer, der die RADIUS-Meldungen zwischen RADIUS kompatiblen Computern sendet. Die RADIUS-Meldungen werden grundsätzlich als UDP-Meldungen (User Datagram Protocol) gesendet. Der UDP-Port 1812 wird für die Authentifizierungsmeldungen und der UDP-Port 1813 für die RADIUS-Kontoführungsmeldungen verwendet. Bei Nutzung von UDP werden Sicherheitsrisiken in Kauf genommen. UDP hat weder Sequenznummern noch eine eigene Verbindungsaufbauprozedur. Damit sind Tür und Tor geöffnet für eine verfälschte Einbringung von UDP-Paketen. Da zudem die Angabe des Sourceports optional ist, lässt sich keine verlässliche Identifizierung des Absenders anhand der IP-Adresse oder des Ports durchführen. Alle Applikationen, die UDP nutzen, müssen geeignete Authentifizierungsmechanismen selbst einbringen. Generell gilt, dass UDP eine Erweiterung des IP-Protokolls ist. Die Erweiterung besteht in der Übermittlung von Source- und Destinationport, über die eine Zuordnung zu den Prozessen geschieht. UDP-Pakete werden im Datenfeld von IP-Paketen übertragen. UDP enthält keine weiteren Maßnahmen zur Verbindungssteuerung, ist damit verbindungslos und gilt als ein ungesicherter Übertragungsdienst.

Vergleichende Betrachtung

[0033] In der Zusammenfassung werden die Internetsicherheitsprotokolle zueinander in Beziehung gesetzt.

[0034] Hier S-HTTP, ein Protokoll, das aus der Sicht des Nutzers sehr einfach zu handhaben ist, das nur horizontal und innerhalb der Anwendungen eine Granulierbarkeit von Sicherheitsfunktionen anbietet, vertikal bei Hosts und Subnetzen ist diese Möglichkeit nicht vorhanden. Außerdem kann hier nur eine Ende-zu-Ende-Sicherheit gebildet werden. Beim S-HTTP werden vorher Sicherheitszertifikate angeboten, die vom Initiator als vertrauenswürdig eingestuft werden. Dies setzt das Prozedere des Austausches von Informationen voraus, die im ungesicherten Umfeld erfolgen. Das Verfahren ist daher kein Authentisierungsverfahren, sondern prüft lediglich die Authentizität eines Eintrages.

PGP (Pretty Good Privacy)

[0035] Bei diesem Verfahren wird nicht die ganze Nachricht asymmetrisch verschlüsselt, sondern die Nachricht symmetrisch und der verwendete Schlüssel asymmetrisch (Hybride Verschlüsselung). Möglichkeit der 1:n Verschlüsselung, keine ONLINE Authentisierung, die Kommunikationsbasis bleibt ungesichert, CONTENT-Security ist die Zielrichtung. Keine Authentisierung, wohl aber Feststellung der Authentizität und Nachricht mit dem Nachteil, dass das Einschleusen einer nicht authentisierten Stelle erfolgen kann (Nutzung RSA, DH/DSS-Algorithmen).

[0036] PGP ist aus der Sicht des Nutzers einfach bis schwierig zu handhaben, je nach der entsprechenden Client-Integration. Die Anwendungstransparenz ist hier nicht erforderlich. Die Anwendungsabhängigkeiten sind sehr hoch, Signaturfunktionen sind vorhanden, die Granulierbarkeit der Sicherheitsfunktionen sind weder horizontal (gleichwertige Stationen) noch vertikal (Servertechnik, hierarchisch höher angeordnete Technik) existent. Die Administrierbarkeit ist relativ aufwändig und auch hier geht es nur um eine Ende-zu-Ende-Sicherheit.

[0037] S/MIME ist Mailverschlüsselung, eine Authentisierung (im Prinzip eine Weiterentwicklung von PGP). E-Mails werden überschlüsselt übertragen. Sender- und Empfängerinformationen bleiben offen. S/MIME nutzt Public Key Verfahren. Nur durch Abrufen der existenten öffentlich zusätzlich und dediziert versteckten Public Keys lässt sich eine bedingte Quelle-Senke-Identifikation realisieren.

[0038] S/MIME ist für den Nutzer einfach bis schwierig zu handhaben. Die Anwendungstransparenz ist hier nicht notwendig, die Anwendungsabhängigkeit aber sehr hoch, Signaturfunktionen sind vorhanden, die Granulierbarkeit der Sicherheitsfunktionen sind weder horizontal noch vertikal möglich. Eine relativ einfache Administrierbarkeit vermittelt auch hier nur eine Ende-zu-Ende-Sicherheit.

[0039] SSH gilt als sehr einfach zu handhabendes Protokoll. Eine Anwendungstransparenz ist nicht nötig, die Anwendungsabhängigkeiten sind durch Tunnelmöglichkeiten mittelhoch. Eine ganz entscheidende Sache ist, dass Signaturfunktionen nicht vorhanden sind. Die Granulierbarkeit der Sicherheitsfunktionen ist horizontal nicht vorhanden und vertikal nur sehr gering durch Tunnellinkfunktionen gegeben. Die Administrierbarkeit ist sehr einfach mit einer Initialisierung der Schlüsselverteilung und nur Ende-zu-Ende-Sicherheitsfunktionen in mehrstufigen komplexen Netzwerkstrukturen sind möglich.

[0040] SSH ermöglicht den Austausch zwischen Quelle und Senke über eine weitgehend sichere authentifizierte und verschlüsselte Verbindung zwischen zwei Netzwerkrechnern. Es kann nur über TCP/IP genutzt werden. Damit ist SSH für Punkt-zu-Punkt-Verbindungen ausschließlich über das Internet interessant.

n:n Fähigkeit bei der Abbildung von Authentisierungen bei Nutzung von SynD.

[0041] Nachfolgend werden die Fähigkeiten von SynD zur Herstellung von n:n Verbindungen dargestellt und zu häufig genutzten Kryptoprogrammen abgegrenzt.

[0042] Das häufig genutzte Programm SSH kann keine n:n Verbindungen authentisieren und nicht so granuliert werden, dass gerade hierarchisch abgestufte Nutzungsrechte abgebildet werden können, wie es SynD ermöglicht.

[0043] SSL ist in der Handhabung sehr einfach, die Anwendungstransparenz hoch, die Anwendungsabhängigkeit relativ hoch, die Signaturfunktionen sind jedoch nicht vorhanden, die Granulierbarkeit der Sicherheitsfunktionen ist horizontal und vertikal gering. Unabhängige Integrationsfähigkeiten in Hard- und Software sind schwierig und dort, wo es geht, ist es sehr inflexibel. Die Administrierbarkeit ist in Abhängigkeit der Infrastruktur komplex bis einfach. Die mehrseitigen Sicherheitsanforderungen in komplexen Netzstrukturen sind relativ schwierig aufzubauen und erlauben nur Ende-zu-Ende-Verschlüsselungen.

[0044] IPsec plus IKE sind extrem einfach zu handhaben. Granulierbarkeiten der Sicherheitsfunktionen sind horizontal nicht existent, vertikal jedoch hoch. Es gibt eine unabhängige Integrationsfähigkeit in Hard- und Software. Die Administrierbarkeit ist komplex bis einfach, je nach der Policy und mehrseitigen Sicherheitsanforderungen. In komplexen Netzstrukturen zeigt sich IPsec und IKE sehr flexibel.

[0045] Bei der Darstellung der vorhandenen Sicherheitsmechanismen zeigt sich, dass insbesondere die Ende-zu-Ende-Verbindungsmöglichkeiten bei den Internetprotokollen zu kurz kommen. Weiterhin werden hier mit IPsec und IKE Schlüsselverfahren angewandt, die mit 3DES oder IDEA realisiert werden oder mit AH. Zu IPsec lässt sich generell sagen, dass IPsec die Daten im Wesentlichen durch zwei Mechanismen schützt: Dem Authentification Header (AH) und den Encapsulating Security Payload (ESP). Die kryptographischen Verfahren zur Nutzung sind nicht festgelegt. Definiert werden hier nur Mindestanforderungen (DES/CBC für ESP, HMAC/SHA für AH). Obergreifende Sicherheitsarchitekturen werden in anderen Veröffentlichungen spezifiziert. Es hat sich gezeigt, dass auf Paketebene keine asymmetrisch kryptographischen Verfahren verwendet werden können. Dies hat seinen Grund darin, dass es offenkundig eine schlechte Performance der Algorithmen gibt. Bei den heutigen Netzwerkgeschwindigkeiten von mehr als 100 MB/sec kann selbst von der leistungsfähigsten Workstation nicht für jedes Paket eine RSA bzw. Ver- oder Entschlüsselung vorgenommen werden. Selbst hoch optimierte Hardwareimplementierungen kommen heute noch nicht über die Grenze von 1 MB/sec hinaus. Deshalb werden als Stand der Technik auf IP-Ebene nur symmetrische kryptographische Verfahren zum Einsatz gebracht. Das zum Patent angemeldete SynD-Verfahren behebt diesen Engpass.

[0046] ESP dient der Nutzdatenüberschlüsselung, wird derzeit nur mit symmetrischen Schlüsseln durchgeführt und spielt bei der Authentisierung/Authentifizierung des beschriebenen und zum Patent angemeldeten Verfahrens keine Rolle.

Firewallunabhängigkeit von SynD

[0047] SynD nutzt bei der ersten Verbindung mittels IP bereits die Authentisierung und Feststellungsmöglichkeit der Authentizität portunabhängig. Wird der ICMP-Request durch eine Firewall geblockt, erfolgt keine Authentisierung. Die Möglichkeit der Diskriminierung des Patterns innerhalb des Pings lässt sich mit geringem Aufwand bereits auf der Firewallebene ggf. mit Datenbasis im Corporate LAN realisieren.

Unabhängigkeit von SynD vom genutzten Übertragungsprotokoll

[0048] SynD ist prinzipiell unabhängig von der Art des genutzten Übertragungsprotokolls, auch wenn in dieser Schrift die Anwendung mittels IP/ICMP beschrieben ist.

[0049] Auch im Bereich von synchroner Datenübertragung oder der Nutzung von Cellorientierten Verfahren (ISDN, ATM) lässt sich SynD anwenden. Gerade die Plattformunabhängigkeit und leichte Portierbarkeit ist ein entscheidender Vorteil dieses Authentifizierungs-/Authentisierungsverfahrens. SynD bietet die Kombinationsmöglichkeit mit allen anderen gängigen Schlüsselmitteln (RSA, DH ...).

Sich aus den Alleinstellungsmerkmalen ableitende Aufgabenstellung für SynD

Aufgabenstellung:

[0050] Die angemeldete Erfindung bietet eine technische Problemlösung eines Kryptographiesystems und eines Datenverschlüsselungsverfahrens, die in der Lage sind, bereits erwähnte Unzulänglichkeiten des Standes der Technik auszuschalten und damit die Sicherheit der Datenübertragung massiv zu erhöhen.

Kurzzusammenfassung für SynD:

[0051] In einer mikrocomputerbasierten technischen Implementierung werden innovative Verfahren zur sicheren Kommunikationsauthentisierung bereitgestellt. Die Verwendung dieser Überschlüsselung, der hochkarätigen Verschlüsselung von Kommunikationsendstellen erfolgt ausgehend von einem Trustcenter über eine dedizierte und auf beliebige Nutzer verteilbare hierarchisch oder nicht hierarchisch geordnete Verschlüsselungsstelle zu realisieren. Abhebend von den bisher bekannten Verfahren zur Überschlüsselung/Verschlüsselung bietet diese Lösung einen sehr hohen Performancegewinn sowie die Möglichkeit zur Miniaturisierung der Hardware und damit eine hohe Einbruchssicherheit.

Grenzen und Abhängigkeiten von SynD in Bezug auf den Stand der Technik

Anforderungen an das SynD-System und Abgrenzung:

[0052] Das hier vorgelegte Patent beschreibt den Bereich der Authentisierung mit seinen Schnittstellen sowie die extrem gesicherte Blockdatenübertragung.

[0053] Eine Kombination zwischen asymmetrischen Schlüsselverfahren und symmetrischen Schlüsselverfahren wird eingeführt. Die asymmetrischen Schlüsselverfahren werden genutzt, um den Datentransport kryptographisch abzusichern und über eine eigene Trust-Hierarchie Verifikationsprozesse zu realisieren.

[0054] Die Bereitstellung der Schlüssel, die Übertragungswege und die Form der Überschlüsselung werden vorgegeben in Algorithmen, die asymmetrisch die einzelnen Partner erreichen.

Prinzip auf der Grundlage einer Übertragung von Quelle zu Senke im Internet, hier: Nutzung eines bestimmten Protokolltyps

Beschreibung der prinzipiellen Funktionsweise von SynD:

[0055] SynD ist das Akronym für System nicht dechiffrierbarer Authentisierung und basiert auf einer neuen Form der Primzahlzerlegung. Es nutzt das ICMP-Protokoll. ICMP (Internet Control Message Protocol) dient dazu, Fehlermeldungen, Steuerinformationen oder Statusmeldungen zu übertragen. ICMP ist kein selbständiges Transportprotokoll, sondern nutzt das Datenfeld von IP zur Übertragung. Auf diese Weise ist es möglich, dass die IP-Software auf verschiedenen Systemen und Endsystemen Informationen und Nachrichten übermittelt. Auch höhere Protokolle und Anwendungen können damit über Fehlerzustände informiert werden. Die Tatsache, dass Daten und Informationen übertragen werden, sieht man beim einschlägigen RFC 792, der den Echoreply beschreibt. Unabhängig davon ist im RFC ICMP (letztes Update vom 25.07.2006) eine Mehrzahl von Informationsfeldern als so genannte Typfelder identifiziert. Im Typbereich 19 ist eine Reservierung für Sicherheitsanfragen gegeben. In den Feldern 20 bis 29 besteht die Möglichkeit der Reservierung für so genannte robustness experiments. Beide Typen der Felder 19 und 20 bis 29 spielen bei SynD eine bedeutende Rolle. So wird auf diese Weise mit einem ICMP-Request an eine oder mehrere Stationen ein Autorisierungsschlüssel gesandt, der dann über den Echorequest bearbeitet zurückgesandt wird. Dies kann sowohl ein System sein, das in einer Punkt-zu-Punkt-Beziehung mit der Quelle steht. Es kann aber auch ein ganzes Netzwerk oder mehrere Stationen gleichzeitig bedienen. Hier werden die Mechanismen genutzt, die ICMP und IP bieten, z.B. ist es möglich, einen generellen Ping auf eine Broadcast-Adresse zu senden, worauf alle Stationen, die innerhalb dieses Netzwerks liegen, mit ihrem Echorequest antworten, es sei denn, dass dieser entsprechend nicht aufgeschaltet wurde. Als Träger des beschriebenen Authentisierungsverfahrens bietet sich daher das ICMP-Protokoll an. Durch die Nutzung des ICMP-Protokolls sind zusätzliche Authentisierungsprotokolle wie RADIUS-Protokoll überflüssig. Das ICMP-Protokoll bedient sowohl Nutzer (Useranfragen) bzw. Client-Server-Beziehungen konventioneller Art, wie auch Client-Server-Anfragen, in denen Maschinenadressen abgerufen werden. Insbesondere durch die Möglichkeit des Mobile IP und der Erweiterung der Adressräume im Bereich IPv6 bietet sich dieses Protokoll an. Die Verbindung des ICMP-Requests mit Security Features, die bereits auf der verbindungsorientierten Ebene, bevor in irgendeiner Weise Datenströme gelaufen sind, genutzt werden können, verhindert ein schlechtes Zeit-Antwort-Verhalten. Die gesendeten Informationen innerhalb des Feldes 19 oder der Felder 20 bis 29 werden automatisiert von den Maschinen abgearbeitet und die Authentifikation und die Authentisierung erfolgen in ähnlicher Geschwindigkeit wie mit einem Standardping.

[0056] Zur Absicherung aller Verfahren sind entsprechende kryptographische Hash-Schlüsselfunktionen, die ein Komprimat im Sinne einer Prüfsumme ermitteln, erforderlich.

[0057] Die unter RC4 veröffentlichte, von der RSA Data Security Inc. vermarktete Verfahrensart, die sowohl bei Lotus Notes, Oracle oder SQL Features eine bedeutende Rolle hat, nutzt eine so genannte typische Stromchiffrierung und ist damit unabhängig von Klartextinformationen. Die einzelnen Schlüsselstrombytes werden mit geheimem Schlüssel erzeugt. Auch hier ist die Einbringung des Cwienk'schen Algorithmus durchaus möglich und sinnvoll. Die Größe des vorgegebenen Schlüsselraums mit 2^{1700} zerlegt auch $256^2 \times 256!$, lässt sich in der Cwienk'schen Aufbereitung der entsprechenden Algorithmen ohne Schwierigkeiten overroulen.

[0058] Das Volumen der Schlüssel kann nach diversen Verfahren geordnet und beliebig erweitert werden. Ein Verschlüsselungssystem (VS) als Bestandteil von SynD kann unterschiedliche Eigenschaften der Primzahl und der nicht primen Zahlen nutzen, wodurch der Verschlüsselungsraum unbegrenzt erweiterbar ist.

[0059] Zur Definition der Folgen der Primzahlen und nicht primen Zahlen, die in die nicht entschlüsselbaren Zerlegungen eingehen, wird auf bekannte und bisher unbekannte Gesetzmäßigkeiten zurückgegriffen, auf die in den unter NDA zugestellten Monographien hingewiesen wird.

[0060] Die Schlüsselstruktur und die Schlüsselmenge wird zunächst von der dI-DATEN GmbH festgelegt. Teilmengen dieses Schlüssels können den Unternehmen zur Verfügung gestellt werden, die dieses Patent als Lizenz zur Nutzung erwerben.

[0061] Die Verteilung der hardwareimplementierten Teilmengen dieser Schlüssel obliegt dem Lizenznehmer.

Gewerbliche Anwendbarkeit:

Sichere Authentisierung von beliebigen Endstellen (Nutzern) mit einer Zentrale oder einer Hierarchie von Zentralen
Sicherstellen von Überprüfungen der Datenübertragungsintegrität
Abbilden zusätzlicher Absicherungsmaßnahmen bei der verbindungsbezogenen Authentisierung durch Einsatz von additiven Datenflussintervallverschlüsselungen
Bei Banken, Sparkassen, Telekommunikation jeglicher Art, großer und mittlerer Industrie, Pharmaindustrie, Versicherungen, Gesundheitswesen, Krankenkassen, Behördensicherheit (Ministerien, Verfassungsschutz, Polizei, Justiz, Bundeswehr etc.)

[0062] In zunehmender Entwicklung erhält die Miniaturisierung im Rahmen der Mobilfunk-Telefonie eine besondere Bedeutung. Die enorme Freiheit der Kommunikationsmöglichkeiten findet heute dort Grenzen, wo die Sicherheit in der Kommunikation bei Verbindungsaufbau und Übertragungsinitiierung nicht hinreichend ist. Die Sicherstellung der Verbindungsintegrität bildet damit die Grundlage für die Minimierung der genannten Restrisiken und erlaubt eine Miniaturisierung der zu implementierenden Hardware.

[0063] Mit Einsatz von mobiler IP und IPnG/IPv6 werden idealtypische Einsatzszenarien und -möglichkeiten für SynD geschaffen. Die Anteile von IPSec können dadurch in erheblichem Maße ausgeweitet werden.

Zusatzinformationen:

[0064] Die im Eigentum der dI-DATEN GmbH befindlichen, nicht veröffentlichte Monographien von Dr. Georg Viktor Cwienk, 2006, auf deren Basis implementierte Verschlüsselungsverfahren in großer Menge generierbar sind, die nach diversen Ordnungsschlüsselverfahren strukturiert sind.

[0065] Diese Monographien (beigefügt) bilden die deklarative Basis dieses Patents und beweisen die nachfolgend erwähnten Gesetzmäßigkeiten.

[0066] Darüber hinaus wird darauf hingewiesen, dass der Grad der Verschlüsselungssicherheit durch eine Implementierung mehrerer Sicherheitsschlüssel auf Basis wohl definierter Primzahl/nicht prime Zahlen und wohl definierter gerader Zahlen noch wesentlich erhöht werden kann (siehe die beigefügten Figuren).

[0067] Das angemeldete Patent zeigt in der Abgrenzung auf, dass bestimmte Verfahren, die in der Vergangenheit angewendet wurden und teilweise zu eigenständigen Patenten geführt haben, durch die neue Patentschrift zu einem weit komplexeren, nicht mehr entschlüsselbaren System führt.

[0068] Insbesondere sind nachfolgend aufgeführte Arbeitsprämissen, die in der Vergangenheit zu probabilistischen Verfahren geführt haben, durch neu gefundene Gesetzmäßigkeiten wesentlich erweitert worden. Die These, dass es kein Muster in den Primzahlen gibt, ist durch die Erweiterung der Gleichung für den Beweis des Hauptsatzes der Primzahltheorie und durch die Ergänzung der Goldbach'schen Theorie wesentlich erweitert worden. Primzahlen, die in den addierten bzw. subtrahierten geraden bzw. ungeraden Zahlen als Faktoren auftreten, dürfen in den vorgegebenen geraden bzw. ungeraden Zahlen nicht auftreten. Diese Prämisse führt zu Gleichungssystemen, die prime Zahlen oder aber Zahlen in den Primzahlfaktoren mit größeren Primzahlen als die Primzahlen, die in den Zahlen n_g bzw. n_n auftreten, enthalten.

[0069] In der Anlage befinden sich die Monographien des Dr. Cwienk „Neue Aspekte der Primzahltheorie", die u. a. in Teil 1 und Teil 2 zu einer Erweiterung der Gleichung des Hauptsatzes der Primzahltheorie führen. Es wird insbesondere auf die Seiten 115 ff., Überschrift "20 Zusammenfassung und Ausblick" verwiesen sowie auf die Zusammenfassung der Gleichung des Hauptsatzes Seite 122 (siehe dazu auch Zeilen 495/496).

[0070] Die Erkenntnisse aus diesen mathematischen Beweisen, die immanenter Bestandteil der Patentschrift und des hier angemeldeten Patents sind, führen zu einer wesentlich höheren Absicherungsmöglichkeit, als die Verfahren, die unter RSA Anwendung gefunden haben.

Ergänzender deklarativer Patentteil

[0071] Erläuterung zu Fig. 4:
1) Authentisierung
Definierte Tsunamiwellen (ZW)
2) Abgeleitete gekoppelte Zahlen zu den ZW = GZW
Diese eröffnen die Textblockverschlüsselung und beenden diese.
3) Text-Verschlüsselung mit definierten Primzahlwellen (PW):
jedem Buchstaben,
jeder Ziffer,
jedem Satzzeichen
werden im Textblock unwiederholbare Primzahlen (PZ) zugeordnet.

Die Überschneidungsvermeidung (Ausschließung)

[0072] Eine Identität der Blockwellen (Zahlen) mit dem GZW wird dadurch ausgeschlossen, dass das Abfrageintervall höher gesetzt wird, wodurch sich große GZW-Werte ergeben.

[0073] Die der Patentanmeldung beigefügten mathematischen Grundlagen der „additiven und subtraktiven" Datenverschlüsselungen mit Goldbach-Zerlegungen, Quasi-Goldbach-Zerlegungen sowie Anti-Goldbach-Zerlegungen und Quasi-Anti-Goldbach-Zerlegungen und ihren definierten (wählbaren) Kombinationen fußen auf den folgenden unveröffentlichten Monographien und den in ihnen beschriebenen neuen Erkenntnissen von Gesetzmäßigkeiten in der Primzahltheorie, die die Grundlage des SynD-Verfahrens darstellen:

1. Die unveröffentlichte Monographie zu dem Thema:
„Neue Aspekte der Primzahl-Theorie – Teil I"/April 2006
Verfasser: Dr. Georg Cwienk
[MNda- Cwienk2006]
2. Die unveröffentlichte Monographie zu dem Thema:
„Neue Aspekte der Primzahl-Theorie – Teil II"/August 2006
Verfasser: Dr. Georg Cwienk
[MNda- Cwienk2006]

[0074] Auf Anforderung kann ein entsprechendes Beispiel bereitgestellt werden.

Ergänzende Erläuterungen zur Patentschrift SynD hier: Ergänzendes deklaratives Grundlagenmaterial

[0075] Für die Generierung der Primzahlwellen und der Primzahl-Tsunamiwellen sind von ausschlaggebender Bedeutung die Gesetzmäßigkeiten in der Monographie „Neue Aspekte der Primzahltheorie" Teil I, Blockschema I, Blockschema II und Blockschema III. Diese Gesetzmäßigkeiten sind auf dem Deckblatt der zitierten Monographie angeführt. In dieser Monographie wurde ebenfalls der Nachweis erbracht, dass die Quasi-Goldbach-Zerlegungen und die Quasi-Anti-Goldbach-Zerlegungen in gleichen Intervallen in einer gleichen Mächtigkeit (Anzahl) auftreten.

[0076] Die Gesetzmäßigkeit in Block I ist eine Erweiterung der euklidischen Gesetzmäßigkeit, dass eine natürliche Zahl, die faktoriell aus Primzahlen zusammengesetzt ist, durch Addition der Zahl 1 zu einer neuen Primzahl führen kann bzw. zu neuen Primzahlen. Diese euklidische Gesetzmäßigkeit, auf der der Beweis aufbaut, dass es unendlich viele Primzahlen gibt, ist im Block I dadurch erweitert worden, dass sowohl die Addition von ungeraden Zahlen: +1, +3, +5 etc. bzw. die Subtraktion ungerader Zahlen: –1, –3, –5 usw. zu neuen Primzahlen führen kann, wie aber auch die Addition gerader Zahlen: +2, +4, +6 bzw. die Subtraktion gerader Zahlen: –2, –4, –6 usw. neue Primzahlen ergeben kann.

[0077] In Block II sind die Progressionen erweitert und zwar zu zwei Obergruppen: Obergruppe I, Obergruppe II.

[0078] In der Obergruppe I treten vier Klassen von Primzahlen auf mit den Endziffern 1, 3, 7, 9, die bei Teilung durch die Zahl 4 den Rest 1 ergeben.

[0079] In der Obergruppe II sind wiederum vier Klassen von Primzahlen angegeben, die bei Teilung durch die Zahl 4 den Rest 3 ergeben.

[0080] Im Block III sind die Bildungsgesetze für die Folgen der Zwillinge Mod 2n n>= 1 angegeben.

[0081] Die Gesetzmäßigkeiten, die in Block I dargestellt sind, führen dazu, dass Primzahlwellen (durch Goldbach'sche oder Anti-Goldbach'sche Zerlegungen) generiert werden können und zwar in ziemlich großen, vorgegebenen Intervallen. Diese Gesetzmäßigkeiten von Block I führen aber auch dazu, dass man Primzahl-Tsunamiwellen generieren kann, ebenfalls in vorgegebenen Intervallen, und diese Gesetzmäßigkeiten führen dazu, dass man die mit diesen Primzahl-Tsunamiwellen gekoppelten Zahlen ermitteln kann, die den Textblöcken vor- und nachgeschaltet werden – diese eröffnen und abschließen.

Musterschreiben

Textblock 1

das testschreiben vom 12.09.06
herrn
dr. georg cwienk
auf der egge 20
44797 bochum
wh/sj 05101/8 54 64-10 12.09.2006
die auf der basis ihrer theoretischen untersuchungen inzwischen
implementierten datenverschlüsselungen – patentsache 1
sehr geehrter herr dr. cwienk,
wir freuen uns, ihnen mitteilen zu können, dass wir die datenverschlüsselung

Textblock 2

1. die authentisierung
2. die textverschlüsselung

inzwischen implementiert haben.
sie finden beiliegend die cd, die ihnen die entschlüsselung dieses an sie
gerichteten schreibens ermöglicht.
mit freundlichen grüßen
ihr herbert werner

Zw I

Z → E	Z ← E	Z → E
278891	321243	479275571

GZw I

6139039

Pw I

d	a	s		t	e	s
7008	1572	68322	281598	71850	9718	68371
t	s	c	h	r	e	i
71899	68443	5230	17088	65118	9767	20080
b	e	n		v	o	m
3712	9839	40650	281647	92262	43492	35848
	1	2	.	0	9	.
281719	60	142	147058	58	862	147107
0	6			h	e	r
107	522	281767	281887	17173	9887	65167
r	n		d	r	.	
65239	40699	281959	7057	65287	147179	284608
g	e	o	r	g		c
14200	10007	43541	65407	14249	284697	5279
w	i	e	n	k		
93202	20129	10079	40771	25180	284729	284777
a	u	f		d	e	r
1621	84688	12468	284897	7129	9780	65479
	e	g	g	e		2
284897	9829	14321	14369	9901	284969	191
0		4	4	7	9	7
179	284969	298	347	652	911	701
	b	o	c	h	u	m
28678	3761	43613	5351	17209	84737	35897
		u	n	s	e	r
286927	286999	84809	40819	68491	9949	65668
	z	e	i	c	h	e
287047	119802	10069	20201	5399	17257	10141
n		t	e	l	e	f
40939	287167	71971	10102	34182	10151	12517
o	n		d	a	t	u
43661	41011	287239	7177	1693	72019	84857

m			w	h	/	s
35969	288880	288929	93251	17377	296080	68611
j		0	5	1	0	1
22738	289001	227	450	109	347	181
/	8	5	4	6	4	-
296129	708	499	419	571	467	177972
1	0		1	2	.	0
229	419	289049	349	263	147227	14172
9	.	2	0	0	6	
983	147347	311	14221	14293	643	289169
	d	i	e		a	u
289169	7297	20249	10223	289241	1741	84977
f		d	e	r		b
12589	309730	7369	10271	65717	309779	3833
a	s	i	s		i	h
1861	68683	20369	70260	309851	20441	17449
r	e	r		t	h	e
65789	10391	65837	309899	72139	17788	10463
o	r	e	t	i	s	c
43781	65957	12252	72211	21058	70309	5519
h	e	n		u	n	t
17837	12301	41728	310019	85049	41777	74722
e	r	s	u	c	h	u
12373	66029	70381	85049	5591	17909	85548
n	g	e	n		i	n
41849	14369	12421	41897	310091	21107	42017
z	w	i	s	c	h	e
119851	93323	21179	70429	5350	17957	12541
n		i	m	p	l	e
42089	312960	21227	36017	50478	34231	12613
m	e	n	t	i	e	r
36137	317542	42282	74771	21347	317591	67530
t	e	n		d	a	t
74843	317663	42331	313009	7162	1933	74891
e	n	v	e	r	s	c
317711	42403	92311	317831	67579	70549	5399
h	l	ü	s	s	e	l
18077	34303	326958	70621	70800	317903	34351
u	n	g	e	n		-
85597	42451	14489	319230	42571	313081	178021

	p	a	t	e	n	t
313129	50527	2188	75011	319279	42643	75083
s	a	c	h	e		1
70849	2237	5471	18149	319351	313249	421
		s	e	h	r	
313321	313938	70921	319399	17928	67651	313987
g	e	e	h	r	t	e
14561	319519	319591	17977	67699	77422	324130
r		h	e	r	r	
67819	314059	18049	324179	67891	68160	314107
d	r	.		c	w	i
7211	68209	147419	314227	5519	93371	21419
e	n	k	,			w
324251	42288	25229	155488	314299	316908	93491
i	r		f	r	e	u
21988	68281	316908	12637	68329	324299	85597
e	n		u	n	s	,
324419	42337	316957	85669	42409	70969	155537
	i	h	n	e	n	
317029	22037	18097	42457	324491	42577	317077
m	i	t	t	e	i	l
36209	22109	77471	77543	326890	22157	34471
e	n		z	u		k
326939	42649	317197	119923	85717	317269	25301
ö	n	n	e	n	,	
332872	347440	347489	327011	347561	155537	353508
d	a	s	s		w	i
7211	2309	71089	71161	353557	93563	22277
r		d	i	e		d
68449	353629	7283	22349	327059	353667	7331
a	t	e	n	v	e	r
2357	77591	327178	347609	92383	327251	68521
s	c	h	l	ü	s	s
70830	5639	18217	34543	327007	70879	70951
e	l	u	n	g		
365080	34318	85837	347729	14470	353797	353869

GZw I
6139111

Zw II		
Z → E	Z ← E	Z → E
2850737	4297813939	4503599630217143

GZw II
2846641

Pw II

1	.		d	i	e	
2325	5834193	9446967	144765	659583	214425	9446981
a	u	t	h	e	n	t
72693	3293373	2765505	599667	214439	1389027	2765519
i	s	i	e	r	u	n
659597	2522367	659609	214451	2356983	3293387	1389221
g			2	.		d
404955	9446993	9447041	3975	5834207	9447233	144779
i	e		t	e	x	t
659657	214499	9448001	2765531	214691	4312467	2765579
v	e	r	s	c	h	l
3847563	215459	2356997	2522381	135183	599681	1012215
ü	s	s	e	l	u	n
9049185	2522393	2522441	218551	1012229	3293399	1389233
g			i	n	z	w
404969	9451093	9460617	659849	1389281	5081553	3906783
i	s	c	h	e	n	
660617	2522633	135197	599693	229473	1389473	9460631
i	m	p	l	e	m	e
663709	1271583	1800693	1012241	229487	1271597	229499
n	t	i	e	r	t	
1390241	2765771	668673	229547	2357009	2766539	9460643
h	a	b	e	n	.	
599741	72707	114735	229739	1393333	5834219	9460691
	s	i	e		f	i
9460883	2523401	668687	230507	9461651	305397	668699
n	d	e	n		b	e
1423437	144791	233599	1423451	9464743	114749	240477
i	l	i	e	g	e	n
668747	1012289	668939	240491	404981	240503	1423463
d		d	i	e		c
144839	9473337	145031	669707	240551	9473351	135209

d 145799	, 6068385	9473363	d 148891	i 672799	e 240743	9473411
i 680223	h 599933	n 1423511	e 241511	n 1423703	9473603	d 145017
i 680237	e 244603	9474371	e 275973	n 1424471	t 2769631	s 2526493
c 135257	h 600701	l 1012481	ü 9049199	s 2623143	s 2623157	e 275987
l 1013249	u 3293447	n 1427563	g 405029	9477463	d 145031	i 680249
e 275999	s 2623169	e 276047	s 2623217	9609837	a 72719	n 1427487
s 2623409	i 680297	e 276239	9609851	g 405221	e 277007	r 2357057
i 680489	c 135449	h 603793	t 2866197	e 280099	t 2866211	e 5599215
n 1427501	9609863	s 2624177	c 136217	h 648243	r 2357249	e 5599229
i 681257	b 114761	e 5599241	n 1427513	s 2627269	9609911	e 5599289
r 2358017	m 1271609	ö 9744783	g 405989	l 1016341	i 684349	c 139309
h 648257	t 2866223	. 5834267	9610103	961081	m 1271657	i 734397
t 2866271	9613963	f 305411	r 2361109	e 5599481	u 3293639	n 1427561
d 145043	l 1149513	i 734411	c 137373	h 648269	e 5600249	n 1427753
9699537	g 409081	r 2376345	ü 9049211	ß 5300937	e 5603341	n 1428521
9699551	9699563	i 734423	h 648317	r 2376359	9699611	h 648509
e 5616843	r 2376371	b 114809	e 5616857	r 2376419	t 2866463	9699537
w 3906797	e 5616869	r 237661	n 1431613	e 5616917	r 2377379	

GZw II
2846643

278891 321243 479275571

6139039

7008	1572	68322	281598	71850	9718	68371
71899	68443	5230	17088	65118	9767	20080
3712	9839	40650	281647	92262	43492	35848
281719	60	142	147058	58	862	147107
107	522	281767	281887	17173	9887	65167
65239	40699	281959	7057	65287	147179	284608
14200	10007	43541	65407	14249	284697	5279
93202	20129	10079	40771	25180	284729	284777
1621	84688	12468	284897	7129	9780	65479
284897	9829	14321	14369	9901	284969	191
179	284969	298	347	652	911	701
28678	3761	43613	5351	17209	84737	35897
286927	286999	84809	40819	68491	9949	65668
287047	119802	10069	20201	5399	17257	10141
40939	287167	71971	10102	34182	10151	12517
43661	41011	287239	7177	1693	72019	84857
35969	288880	288929	93251	17377	296080	68611
22738	289001	227	450	109	347	181
296129	708	499	419	571	467	177972
229	419	289049	349	263	147227	14172
983	147347	311	14221	14293	643	289169
289169	7297	20249	10223	289241	1741	84977
12589	309730	7369	10271	65717	309779	3833
1861	68683	20369	70260	309851	20441	17449
65789	10391	65837	309899	72139	17788	10463
43781	65957	12252	72211	21058	70309	5519

2850737 4297813939 4503599630217143

2846641

2325	5834193	9446967	144765	659583	214425	9446981
72693	3293373	2765505	599667	214439	1389027	2765519
659597	2522367	659609	214451	2356983	3293387	1389221
404955	9446993	9447041	3975	5834207	9447233	144779
659657	214499	9448001	2765531	214691	4312467	2765579
3847563	215459	2356997	2522381	135183	599681	1012215
9049185	2522393	2522441	218551	1012229	3293399	1389233
404969	9451093	9460617	659849	1389281	5081553	3906783
660617	2522633	135197	599693	229473	1389473	9460631
663709	1271583	1800693	1012241	229487	1271597	229499
1390241	2765771	668673	229547	2357009	2766539	9460643
599741	72707	114735	229739	1393333	5834219	9460691
9460883	2523401	668687	230507	9461651	305397	668699
1423437	144791	233599	1423451	9464743	114749	240477
668747	1012289	668939	240491	404981	240503	1423463
144839	9473337	145031	669707	240551	9473351	135209
145799	6068385	9473363	148891	672799	240743	9473411
680223	599933	1423511	241511	1423703	9473603	145017
680237	244603	9474371	275973	1424471	2769631	2526493
135257	600701	1012481	9049199	2623143	2623157	275987
1013249	3293447	1427563	405029	9477463	145031	680249
275999	2623169	276047	2623217	9609837	72719	1427487
2623409	680297	276239	9609851	405221	277007	2357057
680489	135449	603793	2866197	280099	2866211	5599215
1427501	9609863	2624177	136217	648243	2357249	5599229
681257	114761	5599241	1427513	2627269	9609911	5599289
2358017	1271609	9744783	405989	1016341	684349	139309

648257	2866223	5834267	9610103	961081	1271657	734397
2866271	9613963	305411	2361109	5599481	3293639	1427561
145043	1149513	734411	137373	648269	5600249	1427753
9699537	409081	2376345	9049211	5300937	5603341	1428521
9699551	9699563	734423	648317	2376359	9699611	648509
5616843	2376371	114809	5616857	2376419	2866463	9699537
3906797	5616869	237661	1431613	5616917	2377379	
2846643						

Tabelle Zw I

```
for i from 0 to 10^7 do:
 if isprime (7!+700+i)
   and isprime (8!+772+i)
   and isprime (12!+820+i)
   and isprime (14!+940+i)
   and  isprime (17!+1012+i)
   and isprime (21!+1180+i)
  then print (i,7!+700+i,8!+7772+i,12!+820+i,14!+940+i,17!+1012+i,21!+1180+i)
 end_if:
```

273151	278891	321243	479275571	87178565291	355687428370163	51090942171709714331
319201	324941	367293	479321621	87178611341	355687428416213	51090942171709760381
1504381	1510121	1552473	480506801	87179796521	355687429601393	51090942171710945561
1508439	1514179	1556531	480510859	87179800579	355687429605451	51090942171710949619
2364541	2370281	2412633	481366961	87180656681	355687430461553	51090942171711805721
2949367	2955107	2997459	481951787	87181241507	355687431046379	51090942171712390547
3068941	3074681	3117033	482071361	87181361081	355687431165953	51090942171712510121
3320881	3326621	3368973	482323301	87181613021	355687431417893	51090942171712762061
4299159	4304899	4347251	483301579	87182591299	355687432396171	51090942171713740339
4691971	4697711	4740063	483694391	87182984111	355687432788983	51090942171714133151
5081847	5087587	5129939	484084267	87183373987	355687433178859	51090942171714523027
5981427	5987167	6029519	484983847	87184273567	355687434078439	51090942171715422607
6138339	6144079	6186431	485140759	87184430479	355687434235351	51090942171715579519
6610479	6616219	6658571	485612899	87184902619	355687434707491	51090942171716051659
6654321	6660061	6702413	485656741	87184946461	355687434751333	51090942171716095501
6689719	6695459	6737811	485692139	87184981859	355687434786731	51090942171716130899
7039341	7045081	7087433	486041761	87185331481	355687435136353	51090942171716480521
7112779	7118519	7160871	486115199	87185404919	355687435209791	51090942171716553959
7188367	7194107	7236459	486190787	87185480507	355687435285379	51090942171716629547
7464741	7470481	7512833	486467161	87185756881	355687435561753	51090942171716905921
7716591	7722331	7764683	486719011	87186008731	355687435813603	51090942171717157771
8236369	8242109	8284461	487238789	87186528509	355687436333381	51090942171717677549
8726677	8732417	8774769	487729097	87187018817	355687436823689	51090942171718167857
8945581	8951321	8993673	487948001	87187237721	355687437042593	51090942171718386761
9392329	9398069	9440421	488394749	87187684469	355687437489341	51090942171718833509

Tabelle GZw I

```
for i from 6*10^6 to 10^7 do:
if
isprime (7!+700+i) and
isprime (8!+772+1) and
isprime (12!+820+i) and
isprime (14!+940+i) and
isprime (17!+1012+i) and
isprime (21!+1180+i)
then print (i,700+i,772+i,820+i,940+i,1012+i,1180+i)
end_if:
end_for:
```

6138339	6139039	6139111	6139159	6139279	6139351	6139519
6610479	6611179	6611251	6611299	6611419	6611491	6611659
6654321	6655021	6655093	6655141	6655261	6655333	6655501
6689719	6690419	6690491	6690539	6690659	6690731	6690899
7039341	7040041	7040113	7040161	7040281	7040353	7040521
7112779	7113479	7113551	7113599	7113719	7113791	7113959
7188367	7189067	7189139	7189187	7189307	7189379	7189547
7464741	7465441	7465513	7465561	7465681	7465753	7465921
7716591	7717291	7717363	7717411	7717531	7717603	7717771
8236369	8237069	8237141	8237189	8237309	8237381	8237549
8726677	8727377	8727449	8727497	8727617	8727689	8727857
8945581	8946281	8946353	8946401	8946521	8946593	8946761
9392329	9393029	9393101	9393149	9393269	9393341	9393509

Tabelle Pw I

```
for i from 0 to 10^8 do:
if
isprime (7^2+i) and
isprime (11^2+i) and
isprime (13^2+i) and
isprime (17^2+i) and
isprime (19^2+i)
then_print ("i",i,7^2+i,11^2+i,13^2+i,17^2+i,19^2+i)
end_if:
end_for:
```

0	58	107	179	227	347	419
1	60	109	181	229	349	421
2	142	191	263	311	431	503
3	232	281	353	401	521	593
4	298	347	419	467	587	659
5	450	499	571	619	739	811
6	522	571	643	691	811	883
7	652	701	773	821	941	1013
8	708	757	829	877	997	1069
9	862	911	983	1031	1151	1223
	928	977	1049	1097	1217	1289
	1138	1187	1259	1307	1427	1499
	1500	1549	1621	1669	1789	1861
a	1572	1621	1693	1741	1861	1933
	2188	2237	2309	2357	2477	2549
	2608	2657	2729	2777	2897	2969
	3222	3271	3343	3391	3511	3583
b	3712	3761	3833	3881	4001	4073
	4470	4519	4591	4639	4759	4831
	5058	5107	5179	5227	5347	5419
	5212	5261	5333	5381	5501	5573
c	5230	5279	5351	5399	5519	5591
	5350	5399	5471	5519	5639	5711
	5692	5741	5813	5861	5981	6053
	6400	6449	6521	6569	6689	6761
d	7008	7057	7129	7177	7297	7369
	7162	7211	7283	7331	7451	7523
	9172	9221	9293	9341	9461	9533
	9612	9661	9733	9781	9901	9973
e	9718	9767	9839	9887	10007	10079
	9780	9829	9901	9949	10069	10141
	10102	10151	10223	10271	10391	10463
	12252	12301	12373	12421	12541	12613
f	12468	12517	12589	12637	12757	12829
	12622	12671	12743	12791	12911	12983
	13050	13099	13171	13219	13339	13411
	13710	13759	13831	13879	13999	14071
0	14172	14221	14293	14341	14461	14533
g	14200	14249	14321	14369	14489	14561
	14470	14519	14591	14639	14759	14831
	14578	14627	14699	14747	14867	14939
	15150	15199	15271	15319	15439	15511

h	17088	17137	17209	17257	17377	17449
	17788	17837	17909	17957	18077	18149
	17928	17977	18049	18097	18217	18289
	19980	20029	20101	20149	20269	20341
i	20080	20129	20201	20249	20369	20441
	21058	21107	21179	21227	21347	21419
	21988	22037	22109	22157	22277	22349
	22108	22157	22229	22277	22397	22469
j	22738	22787	22859	22907	23027	23099
	23248	23297	23369	23417	23537	23609
	23620	23669	23741	23789	23909	23981
	23808	23857	23929	23977	24097	24169
k	25180	25229	25301	25349	25469	25541
	26128	26177	26249	26297	26417	26489
	32020	32069	32141	32189	32309	32381
	32202	32251	32323	32371	32491	32563
l	34182	34231	34303	34351	34471	34543
	34318	34367	34439	34487	34607	34679
	34792	34841	34913	34961	35081	35153
	35232	35281	35353	35401	35521	35593
m	35848	35897	35969	36017	36137	36209
	38872	38921	38993	39041	39161	39233
	39502	39551	39623	39671	39791	39863
	40338	40387	40459	40507	40627	40699
n	40650	40699	40771	40819	40939	41011
	41728	41777	41849	41897	42017	42089
	42282	42331	42403	42451	42571	42643
	42288	42337	42409	42457	42577	42649
o	43492	43541	43613	43661	43781	43853
	46278	46327	46399	46447	46567	46639
	46390	46439	46511	46559	46679	46751
	47688	47737	47809	47857	47977	48049
p	50478	50527	50599	50647	50767	50839
	52200	52249	52321	52369	52489	52561
	56310	56359	56431	56479	56599	56671
	62022	62071	62143	62191	62311	62383
q	63078	63127	63199	63247	63367	63439
	63298	63347	63419	63467	63587	63659
	63438	63487	63559	63607	63727	63799
	63568	63617	63689	63737	63857	63929
r	65118	65167	65239	65287	65407	65479
	65668	65717	65789	65837	65957	66029
	67530	67579	67651	67699	67819	67891
	68160	68209	68281	68329	68449	68521
s	68322	68371	68443	68491	68611	68683
	70260	70309	70381	70429	70549	70621
	70800	70849	70921	70969	71089	71161
	70830	70879	70951	70999	71119	71191
t	71850	71899	71971	72019	72139	72211
	74722	74771	74843	74891	75011	75083
	77422	77471	77543	77591	77711	77783
	81678	81727	81799	81847	81967	82039
u	84688	84737	84809	84857	84977	85049
	85548	85597	85669	85717	85837	85909
	88848	88897	88969	89017	89137	89209
	92058	92107	92179	92227	92347	92419

v	92262	92311	92383	92431	92551	92623
	92338	92387	92459	92507	92627	92699
	92502	92551	92623	92671	92791	92863
	92632	92681	92753	92801	92921	92993
w	93202	93251	93323	93371	93491	93563
	94152	94201	94273	94321	94441	94513
	95022	95071	95143	95191	95311	95383
	96618	96667	96739	96787	96907	96979
x	98130	98179	98251	98299	98419	98491
	98548	98597	98669	98717	98837	98909
	99318	99367	99439	99487	99607	99679
	104658	104707	104779	104827	104947	105019
y	104710	104759	104831	104879	104999	105071
	106632	106681	106753	106801	106921	106993
	110560	110609	110681	110729	110849	110921
	119760	119809	119881	119929	120049	120121
z	119802	119851	119923	119971	120091	120163
	123258	123307	123379	123427	123547	123619
	124182	124231	124303	124351	124471	124543
	124860	124909	124981	125029	125149	125221
	125910	125959	126031	126079	126199	126271
ß	129000	129049	129121	129169	129289	129361
	129172	129221	129293	129341	129461	129533
	133270	133319	133391	133439	133559	133631
	136140	136189	136261	136309	136429	136501
	136830	136879	136951	136999	137119	137191
	140508	140557	140629	140677	140797	140869
	144120	144169	144241	144289	144409	144481
	145468	145517	145589	145637	145757	145829
.	147058	147107	147179	147227	147347	147419
	149202	149251	149323	149371	149491	149563
	150600	150649	150721	150769	150889	150961
	155332	155381	155453	155501	155621	155693
,	155488	155537	155609	155657	155777	155849
	156892	156941	157013	157061	157181	157253
	163612	163661	163733	163781	163901	163973
	164532	164581	164653	164701	164821	164893
;	166798	166847	166919	166967	167087	167159
	173380	173429	173501	173549	173669	173741
	175312	175361	175433	175481	175601	175673
	175660	175709	175781	175829	175949	176021
-	177972	178021	178093	178141	178261	178333
	178392	178441	178513	178561	178681	178753
	181668	181717	181789	181837	181957	182029
	183402	183451	183523	183571	183691	183763
"	189102	189151	189223	189271	189391	189463
	190422	190471	190543	190591	190711	190783
	201532	201581	201653	201701	201821	201893
	207208	207257	207329	207377	207497	207569
!	209428	209477	209549	209597	209717	209789
	209500	209549	209621	209669	209789	209861
	209598	209647	209719	209767	209887	209959
	209808	209857	209929	209977	210097	210169
(	210550	210599	210671	210719	210839	210911
	213852	213901	213973	214021	214141	214213
	214020	214069	214141	214189	214309	214381
	222930	222979	223051	223099	223219	223291

)	225460	225509	225581	225629	225749	225821
	227098	227147	227219	227267	227387	227459
	228792	228841	228913	228961	229081	229153
	229728	229777	229849	229897	230017	230089
@	231120	231169	231241	231289	231409	231481
	234882	234931	235003	235051	235171	235243
	244498	244547	244619	244667	244787	244859
	245928	245977	246049	246097	246217	246289
?	250930	250979	251051	251099	251219	251291
	259330	259379	259451	259499	259619	259691
	264972	265021	265093	265141	265261	265333
	268920	268969	269041	269089	269209	269281
Daten-pause	281598	281647	281719	281767	281887	281959
	284608	284657	284729	284777	284897	284969
	286878	286927	286999	287047	287167	287239
	288880	288929	289001	289049	289169	289241
/	296080	296129	296201	296249	296369	296441
	305160	305209	305281	305329	305449	305521
	305232	305281	305353	305401	305521	305593
	306898	306947	307019	307067	307187	307259
Daten-pause	309730	309779	309851	309899	310019	310091
	312960	313009	313081	313129	313249	313321
	313938	313987	314059	314107	314227	314299
	316908	316957	317029	317077	317197	317269
e	317542	317591	317663	317711	317831	317903
	319230	319279	319351	319399	319519	319591
	324130	324179	324251	324299	324419	324491
	326890	326939	327011	327059	327179	327251
ü	326958	327007	327079	327127	327247	327319
	327748	327797	327869	327917	328037	328109
	330738	330787	330859	330907	331027	331099
	331720	331769	331841	331889	332009	332081
ö	332872	332921	332993	333041	333161	333233
	342010	342059	342131	342179	342299	342371
	343540	343589	343661	343709	343829	343901
	345412	345461	345533	345581	345701	345773
n	347440	347489	347561	347609	347729	347801
	348100	348149	348221	348269	348389	348461
	351682	351731	351803	351851	351971	352043
	352060	352109	352181	352229	352349	352421
Daten-pause	353508	353557	353629	353677	353797	353869
	353578	353627	353699	353747	353867	353939
	356910	356959	357031	357079	357199	357271
	363780	363829	363901	363949	364069	364141
e	365080	365129	365201	365249	365369	365441

Tabelle Zw II

```
for i from 0 to 5*10^7 do:
 if
   isprime (2^12+10+i) and
   isprime (2^32+12+i) and
   isprime (2^52+16+i) and
   isprime (2^62+18+i) and
   isprime (2^82+28+i) and
   isprime (2^212+30+i)
  then print (i,2^12+10+i,2^32+12+i,2^52+16+i,2^62+18+i,2^82+28+i,2^212+30+i)
 end_if:
end_for:
```

2846631	2850737	4297813939	4503599630217143	4611686018430234553	4835703278458516701671363	6582018229284824168619876730229402019930943462534319453397282757
24376095	24380201	4319343403	4503599651746607	4611686018451764017	4835703278458516723200827	6582018229284824168619876730229402019930943462534319453418812221
29386581	29390687	4324353889	4503599656757093	4611686018456774503	4835703278458516728211313	6582018229284824168619876730229402019930943462534319453423822707

Tabelle GZw II

```
for i from 0 to 5*10^7 do:
if
isprime (2^12+10+i) and
isprime (2^32+12+i) and
isprime (2^52+16+i) and
isprime (2^62+18+i) and
isprime (2^82+28+i) and
isprime (2^212+30+i)
then print ("i",i,10+i,12+i,16+i,18+i,28+i,30+i)
end_if:
end_for:
```

2846631	2846641	2846643	2846647	2846649	2846659	2846661
24376095	24376105	24376107	24376111	24376113	24376123	24376125
29386581	29386591	29386593	29386597	29386599	29386609	29386611

Tabelle Pw II

```
for i from 0 to 10^7 do:
if
isprime (2^2+10+i) and
isprime (2^4+10+i) and
isprime (2^6+10+i) and
isprime (2^8+10+i) and
isprime (2^10+10+i) and
isprime (2^12+30+i)
then_print ("i",i,2^2+10+i,2^4+10+i,2^6+10+i,2^8+10+i,2^10+10+i,2^12+30+i)
end_if:
end_for:
```

[0082]

0	27	41	53	101	293	1061	4153
1	2325	2339	2351	2399	2591	3359	6451
2	3975	3989	4001	4049	4241	5009	8101
3	4677	4691	4703	4751	4943	5711	8803
4	7827	7841	7853	7901	8093	8861	11953
5	16857	16871	16883	16931	17123	17891	20983
6	26655	26669	26681	26729	26921	27689	30781
7	38625	38639	38651	38699	38891	39659	42751
8	45267	45281	45293	45341	45533	46301	49393
9	45303	45317	45329	45377	45569	46337	49429
	46425	46439	46451	46499	46691	47459	50551
	47955	47969	47981	48029	48221	48989	52081
	53583	53597	53609	53657	53849	54617	57709
a	72693	72707	72719	72767	72959	73727	76819
	104523	104537	104549	104597	104789	105557	108649
	111567	111581	111593	111641	111833	112601	115693
	111753	111767	111779	111827	112019	112787	115879
b	114735	114749	114761	114809	115001	115769	118861
	120903	120917	120929	120977	121169	121937	125029
	120993	121007	121019	121067	121259	122027	125119
	133365	133379	133391	133439	133631	134399	137491
c	135183	135197	135209	135257	135449	136217	139309
	137373	137387	137399	137447	137639	138407	141499
	138375	138389	138401	138449	138641	139409	142501
	140097	140111	140123	140171	140363	141131	144223
d	144765	144779	144791	144839	145031	145799	148891
	145017	145031	145043	145091	145283	146051	149143
	174747	174761	174773	174821	175013	175781	178873
	214017	214031	214043	214091	214283	215051	218143
e	214425	214439	214451	214499	214691	215459	218551
	229473	229487	229499	229547	229739	230507	233599
	240477	240491	240503	240551	240743	241511	244603
	275973	275987	275999	276047	276239	277007	280099
f	305397	305411	305423	305471	305663	306431	309523
	317703	317717	317729	317777	317969	318737	321829
	319023	319037	319049	319097	319289	320057	323149
	346065	346079	346091	346139	346331	347099	350191
g	404955	404969	404981	405029	405221	405989	409081
	431355	431369	431381	431429	431621	432389	435481
	517203	517217	517229	517277	517469	518237	521329
	556767	556781	556793	556841	557033	557801	560893
h	599667	599681	599693	599741	599933	600701	603793
	648243	648257	648269	648317	648509	649277	652369
	648705	648719	648731	648779	648971	649739	652831
	654513	654527	654539	654587	654779	655547	658639

i	659583	659597	659609	659657	659849	660617	663709
	668673	668687	668699	668747	668939	669707	672799
	680223	680237	680249	680297	680489	681257	684349
	734397	734411	734423	734471	734663	735431	738523
j	777405	777419	777431	777479	777671	778439	781531
	782115	782129	782141	782189	782381	783149	786241
	817137	817151	817163	817211	817403	818171	821263
	854097	854111	854123	854171	854363	855131	858223
k	900747	900761	900773	900821	901013	901781	904873
	942243	942257	942269	942317	942509	943277	946369
	983505	983519	983531	983579	983771	984539	987631
	997935	997949	997961	998009	998201	998969	1002061
l	1012215	1012229	1012241	1012289	1012481	1013249	1016341
	1149513	1149527	1149539	1149587	1149779	1150547	1153639
	1200435	1200449	1200461	1200509	1200701	1201469	1204561
	1200975	1200989	1201001	1201049	1201241	1202009	1205101
m	1271583	1271597	1271609	1271657	1271849	1272617	1275709
	1354575	1354589	1354601	1354649	1354841	1355609	1358701
	1375743	1375757	1375769	1375817	1376009	1376777	1379869
	1378833	1378847	1378859	1378907	1379099	1379867	1382959
n	1389207	1389221	1389233	1389281	1389473	1390241	1393333
	1423437	1423451	1423463	1423511	1423703	1424471	1427563
	1427487	1427501	1427513	1427561	1427753	1428521	1431613
	1441287	1441301	1441313	1441361	1441553	1442321	1445413
o	1457847	1457861	1457873	1457921	1458113	1458881	1461973
	1534713	1534727	1534739	1534787	1534979	1535747	1538839
	1618293	1618307	1618319	1618367	1618559	1619327	1622419
	1765275	1765289	1765301	1765349	1765541	1766309	1769401
p	1800693	1800707	1800719	1800767	1800959	1801727	1804819
	1863657	1863671	1863683	1863731	1863923	1864691	1867783
	1878813	1878827	1878839	1878887	1879079	1879847	1882939
	1897677	1897691	1897703	1897751	1897943	1898711	1901803
q	1915215	1915229	1915241	1915289	1915481	1916249	1919341
	1965363	1965377	1965389	1965437	1965629	1966397	1969489
	2182923	2182937	2182949	2182997	2183189	2183957	2187049
	2308857	2308871	2308883	2308931	2309123	2309891	2312983
r	2356983	2356997	2357009	2357057	2357249	2358017	2361109
	2376345	2376359	2376371	2376419	2376611	2377379	2380471
	2422425	2422439	2422451	2422499	2422691	2423459	2426551
	2451783	2451797	2451809	2451857	2452049	2452817	2455909
s	2522367	2522381	2522393	2522441	2522633	2523401	2526493
	2623143	2623157	2623169	2623217	2623409	2624177	2627269
	2633415	2633429	2633441	2633489	2633681	2634449	2637541
	2694147	2694161	2694173	2694221	2694413	2695181	2698273
t	2765505	2765519	2765531	2765579	2765771	2766539	2769631
	2866197	2866211	2866223	2866271	2866463	2867231	2870323
	2979285	2979299	2979311	2979359	2979551	2980319	2983411
	3091947	3091961	3091973	3092021	3092213	3092981	3096073
u	3293373	3293387	3293399	3293447	3293639	3294407	3297499
	3338847	3338861	3338873	3338921	3339113	3339881	3342973
	3436137	3436151	3436163	3436211	3436403	3437171	3440263
	3664233	3664247	3664259	3664307	3664499	3665267	3668359
v	3847563	3847577	3847589	3847637	3847829	3848597	3851689
	3853137	3853151	3853163	3853211	3853403	3854171	3857263
	3865833	3865847	3865859	3865907	3866099	3866867	3869959
	3881277	3881291	3881303	3881351	3881543	3882311	3885403
w	3906783	3906797	3906809	3906857	3907049	3907817	3910909
	3940413	3940427	3940439	3940487	3940679	3941447	3944539
	4076463	4076477	4076489	4076537	4076729	4077497	4080589
	4234707	4234721	4234733	4234781	4234973	4235741	4238833

x	4312467	4312481	4312493	4312541	4312733	4313501	4316593
	4328073	4328087	4328099	4328147	4328339	4329107	4332199
	4407243	4407257	4407269	4407317	4407509	4408277	4411369
	4477353	4477367	4477379	4477427	4477619	4478387	4481479
y	4481175	4481189	4481201	4481249	4481441	4482209	4485301
	4599267	4599281	4599293	4599341	4599533	4600301	4603393
	4671735	4671749	4671761	4671809	4672001	4672769	4675861
	5074833	5074847	5074859	5074907	5075099	5075867	5078959
z	5081553	5081567	5081579	5081627	5081819	5082587	5085679
	5171535	5171549	5171561	5171609	5171801	5172569	5175661
	5270523	5270537	5270549	5270597	5270789	5271557	5274649
	5290587	5290601	5290613	5290661	5290853	5291621	5294713
ß	5300937	5300951	5300963	5301011	5301203	5301971	5305063
	5413203	5413217	5413229	5413277	5413469	5414237	5417329
	5414037	5414051	5414063	5414111	5414303	5415071	5418163
	5446797	5446811	5446823	5446871	5447063	5447831	5450923
e	5599215	5599229	5599241	5599289	5599481	5600249	5603341
	5616843	5616857	5616869	5616917	5617109	5617877	5620969
	5765817	5765831	5765843	5765891	5766083	5766851	5769943
	5771277	5771291	5771303	5771351	5771543	5772311	5775403
.	5834193	5834207	5834219	5834267	5834459	5835227	5838319
	5859603	5859617	5859629	5859677	5859869	5860637	5863729
	5962893	5962907	5962919	5962967	5963159	5963927	5967019
	5995323	5995337	5995349	5995397	5995589	5996357	5999449
,	6068385	6068399	6068411	6068459	6068651	6069419	6072511
	6402705	6402719	6402731	6402779	6402971	6403739	6406831
	6452997	6453011	6453023	6453071	6453263	6454031	6457123
	6614865	6614879	6614891	6614939	6615131	6615899	6618991
;	6615237	6615251	6615263	6615311	6615503	6616271	6619363
	6649557	6649571	6649583	6649631	6649823	6650591	6653683
	6680007	6680021	6680033	6680081	6680273	6681041	6684133
	6904617	6904631	6904643	6904691	6904883	6905651	6908743
-	6911745	6911759	6911771	6911819	6912011	6912779	6915871
	6922635	6922649	6922661	6922709	6922901	6923669	6926761
	7039947	7039961	7039973	7040021	7040213	7040981	7044073
	7143723	7143737	7143749	7143797	7143989	7144757	7147849
"	7151817	7151831	7151843	7151891	7152083	7152851	7155943
	7313367	7313381	7313393	7313441	7313633	7314401	7317493
	7328865	7328879	7328891	7328939	7329131	7329899	7332991
	7439073	7439087	7439099	7439147	7439339	7440107	7443199
!	7477377	7477391	7477403	7477451	7477643	7478411	7481503
	7847967	7847981	7847993	7848041	7848233	7849001	7852093
	7973145	7973159	7973171	7973219	7973411	7974179	7977271
	8015517	8015531	8015543	8015591	8015783	8016551	8019643
(	8210655	8210669	8210681	8210729	8210921	8211689	8214781
	8250243	8250257	8250269	8250317	8250509	8251277	8254369
	8271507	8271521	8271533	8271581	8271773	8272541	8275633
	8313087	8313101	8313113	8313161	8313353	8314121	8317213
)	8345013	8345027	8345039	8345087	8345279	8346047	8349139
	8467527	8467541	8467553	8467601	8467793	8468561	8471653
	8601375	8601389	8601401	8601449	8601641	8602409	8605501
	8849163	8849177	8849189	8849237	8849429	8850197	8853289
Ü	9049185	9049199	9049211	9049259	9049451	9050219	9053311
	9257415	9257429	9257441	9257489	9257681	9258449	9261541
	9257925	9257939	9257951	9257999	9258191	9258959	9262051
	9416265	9416279	9416291	9416339	9416531	9417299	9420391
Daten-pause	9446967	9446981	9446993	9447041	9447233	9448001	9451093
	9460617	9460631	9460643	9460691	9460883	9461651	9464743
	9473337	9473351	9473363	9473411	9473603	9474371	9477463
	9609837	9609851	9609863	9609911	9610103	9610871	9613963
	9699537	9699551	9699563	9699611	9699803	9700571	9703663
ö	9744783	9744797	9744809	9744857	9745049	9745817	9748909
	9817935	9817949	9817961	9818009	9818201	9818969	9822061

Neue Aspekte der Primzahl-Theorie

Das Gesetz, dem die Variationen der Primzahl-Dichten in großen Zahlen-Intervallen zugrunde liegt

Die verborgenen Gesetze, die bei der Zerlegung von Zahlen in ihre Primzahl-Faktoren auftreten

Der Beweis, dass die Primzahl-Zwillinge mod 2, wie auch alle Semi - Zwillinge mod 2n, n>1, in unendlicher Zahl auftreten

Die Klassifizierung und Erweiterung der Mersenne'schen und der Fermat'schen Primzahlen

Verfasser: Georg Viktor Cwienk
Bochum, April 2006

Inhaltsverzeichnis: Seite:

Die Seitenzahl der Monographie (einschließlich der Anlagen) - 305 Seiten

1. Vorworte - Die gelösten und ungelösten Probleme der Primzahl-Theorie:

Seit dem Altertum haben sich zahlreiche berühmte Mathematiker mit der Problematik der Primzahlen beschäftigt. Euklid, der um 300 v. Chr. in Alexandria lebte, stellte einen indirekten Beweis des berühmten Hauptsatzes der Primzahl-Theorie vor, wonach in der unendlichen Menge der natürlichen Zahlen (N) unendlich viele Primzahlen (P) auftreten und die Folge $p := 2, 3, 5, 7, \ldots, p_n$ niemals endet.

Seit dem Mittelalter haben sich so berühmte Mathematiker wie: Fermat, Euler, Gauss, Legendre, Dirichlet, Riemann, Hädamard, Tschebyschew (um nur die wichtigsten zu nennen) mit den Problemen der Primzahlen befasst. Dennoch konnten zahlreiche Vermutungen der Primzahl-Theorie bis heute nicht unter Beweis gestellt werden. Zu den wichtigsten unbewiesenen Vermutungen zählen:

1. - Die Riemann'sche Vermutung, wonach die Zahl der Primzahlen, $\pi(n)$, die Zeta-Funktion erfüllt.

2. - Die Goldbach'sche Vermutung, wonach alle geraden Zahlen $n_g > 2$ als Summe zweier Primzahlen dargestellt werden können.

3. - Die Vermutung, dass es unendlich viele Primzahl-Zwillinge mod 2, $(p_i, p_i + 2)$ gibt.

4. - Die Vermutung, dass es unendlich viele Mersenne- Primzahlen: $M_p = 2^n - 1$ und unendlich viele Fermat- Primzahlen: $F_p = 2^n + 1$ in der unendlichen Menge der natürlichen Zahlen $(n \in N)$ gibt.

Wiewohl es seit langem bekannt ist, dass die Verteilung der Primzahlen in großen Zahlen-Intervallen $(n \to \infty)$ immer größer werdende primzahlfreie Zonen aufweist, zwischen denen Zonen mit größeren Primzahl-Dichten auftreten, konnten für dieses Verhalten der Primzahl-Verteilungen bis heute keine Gesetze gefunden werden, von denen die Verteilungen der Primzahl-Dichten abhängig sind.
Diese Bemühungen wurden aufgegeben, nachdem alle Versuche gescheitert waren, ein Bildungs. Gesetz zu finden, aus dem die Folge der (aller) Primzahlen bestimmt werden kann, was Euler nach intensiven Bemühungen aufgeben musste. Als unlösbar hat es sich auch erwiesen, die Zahl der Primzahlen aus Formeln zu bestimmen, die in beliebigen Zahlen-Intervallen (Δn) auftreten.

Die Mathematiker, allen voran Leonhard Euler (1707, 1783), hatten sich vornehmlich mit dem Gesetz der Summen-Zahl: $\pi(n)$ der Primzahlen befasst, und Euler konstatierte, dass der Quotient: $\pi(n)/n$ bei immer steigenden Zahlen $(n \to \infty)$ asymptotisch dem Grenzwert Null zustrebt, ohne diesen aber je zu erreichen. Bei der Suche nach einer Funktion: $\varphi(n)$, die einen ähnlichen asymptotisch-degressiven Verlauf hat, wie die von Euler eingeführte Funktion: $\pi(n)/n$ stieß Gauss im jugendlichen Alter von nur 15 Jahren (1792) auf die logarithmische Funktion: $\ln(n)$, die dem Verlauf der Euler-Funktion: $\pi(n)/n$ sehr ähnelte, ohne ihr allerdings in kleineren Zahlen-Bereichen (n) stets gleich zu sein. Unter der Mutmaßung, dass beide Funktionen - die Euler'sche Funktion: $\pi(n)/n$ und die Funktion: $\ln(n)$ - einem gleichen Grenzwert zustreben, stellte Gauss die berühmte Grenzwert-Formel auf, die auch als Satz von Gauss bezeichnet wird. Jahre später konnte Hadamard und de la Valleé-Poussin diese Gesetzmäßigkeit unter Rückgriff auf die Riemann'sche Zeta-Funktion unter Beweis stellen.

$$\lim_{n \to \infty} \pi(n) / n / \ln(n) = \lim_{n \to \infty} \pi(n) \cdot \ln(n) / n = 1 \quad \text{..... der Satz von Gauß}$$

Keiner der berühmten Mathematiker, die sich mit der Primzahl-Theorie befasst hatten (von Euler bis Riemann) konnten jedoch Gesetze finden, nach denen die (auch diesen Mathematikern bekannten) Schwankungen der Primzahl-Dichten in steigenden Zahlen-Bereichen (vermeintlich chaotisch) erfolgen. Einen wichtigen Beitrag hatte Gustav Lejeune Dirichlet (1805, 1859) geliefert, der den Nachweis erbracht hatte, dass alle Primzahlen in nur vier arithmetischen Progressionen (die man heute Dirichlet - Progressionen nennt) auftreten können. Diese Progressions-Folgen sind vom Typ:

a, a +d, a +2d, a +3d, ..., a + nd - die Zahlen a und d sind hier teilerfremde Zahlen

In der Folge: $1 + 2n$ (n:= 1, 2, 3, ...) treten alle ungeraden Primzahlen und alle ungeraden Zahlen auf, d.h. die Folge: 1, **3**, **5**, **7**, 9, **11**, **13**, 15, **17**, **19**, 21, **23**, 25, 27, **29**, **31**, 33, 35, **37**, ... - hier wurden die Primzahlen fett gedruckt markiert und die nicht primen Zahlen normal (nicht fett) gedruckt.

Dirichlet bewies, dass diese arithmetische Folge in vier gleichmächtige Teilfolgen aufgeteilt werden kann, in denen (jeweils) alle Primzahlen mit den Endziffern: EZ1, EZ3, EZ7, EZ9 auftreten, die man heute "Dirichlet-Progressionen" nennt - es sind dies die nachstehenden Progressionen, in denen die Primzahlen fett gedruckt markiert sind:

<u>Progression 1:</u> $1 + 10n$:= 1, **11**, 21, **31**, **41**, 51, **61**, **71**, 81, 91, **101**, ... von $n = 0$ bis ∞
<u>Progression 2:</u> $3 + 10n$:= **3**, **13**, **23**, 33, **43**, **53**, 63, **73**, **83**, 93, **103**, ...
<u>Progression 3:</u> $7 + 10n$:= **7**, **17**, 27, **37**, **47**, 57, **67**, 77, 87, **97**, **107**, ...
<u>Progression 4:</u> $9 + 10n$:= 9, **19**, **29**, 39, 49, **59**, 69, **79**, **89**, 99, **109**, ...

In dieser Monographie wird der wichtige Beweis erbracht, dass es acht gleichmächtige Progressionen gibt, in denen die Primzahlen (gleichmächtig) in den Endziffern: EZ1, EZ3, EZ7, EZ9 auftreten.

Mit der vom französischen Mönch Marin Mersenne (1588, 1648) aufgestellten Vermutung, dass in der unendlichen Folge: $2^p -1$, in der im Exponenten ungerade Primzahlen $p \geq 3$ auftreten, unendlich viele Mersenne-Primzahlen: $M_p = 2^p -1$ auftreten, befasst sich seit Jahren das GIMPs-Projekt (Great Internet Mersenne Prime Research), an dem heute Millionen PC-Rechner (weltweit) beteiligt sind. Inzwischen konnte (im Jahr 2005) die 42. Primzahl $M_p = 2^{25964851} -1$ gefunden werden, die bisher größte bekannte Primzahl mit 7.816.230 Dezimalstellen. Die Suche nach noch größeren Primzahlen, M_p, ist in vollem Gange. Von den Fermat-Primzahlen: $F_p = 2^{2^n}+1$ sind hingegen bisher nur fünf Primzahlen bekannt und hier wird bislang keine so intensive Suche nach weiteren Primzahlen vom Typ F_p betrieben. Durch das Auffinden immer weiterer Primzahlen M_p und F_p kann jedoch kein Beweis erbracht werden, dass es unendlich viele Primzahlen M_p und F_p gibt. Um diesen Beweis erbringen zu können, ist der Nachweis erforderlich, dass sich für die Primzahlen M_p und F_p Progressionen der Dirichlet'schen Art aufstellen lassen.

In dieser Monographie wird zunächst der Nachweis erbracht, dass die von Mersenne und von Fermat stammende Definition der Primzahlen: $M_p = 2^p -1$ und $F_p = 2^{2^n}+1$ eine Restriktion der tatsächlich im Umfeld der Folge: $n_g = 2^n$ möglichen Primzahlen vom Typ: $p = 2^n -1$ und $p = 2^n +1$ impliziert.
Es wird der Nachweis erbracht, dass in der Folge: $n_g = 2^n$ sechs Teilfolgen auftreten, wobei jeweils in zwei Teilfolgen Primzahlen vom Typ M_p und in einer Teilfolge Primzahlen vom Typ F_p (fallweise) auftreten können. Aus dieser Klassenteilung der Folge $n_g = 2^n$ resultieren sechs Dirichlet'sche Progressionen: $M_p = 2^{1+4n}-1$, $M_p = 2^{2+4n}-1$, $M_p = 2^{3+4n}-1$, $F_p = 2^{1+4n}+1$, $F_p = 2^{3+4n}+1$, $F_p = 2^{4+4n}+1$ für die eine Prüfung erforderlich ist, ob in ihnen (für $n \to \infty$) unendlich viele Primzahlen auftreten oder ob einige dieser Progressionen "primzahl-leer" sind. In der Monographie wird dabei der Nachweis erbracht, dass Primzahlen vom Typ: M_p und F_p (bis auf den Ausnahmefall: $2^2 -1 =3$ und $2^2 +1 = 5$) niemals Primzahl-Zwillinge mod 2 ergeben können. Um an die beiden Primzahl-Forscher: Mersenne und Fermat anzuknüpfen, werden die Bezeichnungen $M_p = 2^n -1$ und $F_p = 2^n +1$ auch für die Primzahlen beibehalten, in denen die Exponenten nicht prim sind ($n \neq p$) bzw. nicht vom Typ: $n = 2^{2^n}$.
Wenngleich es unendlich viele gerade Zahlen der einfachsten Struktur: $n_g = 2^n$ gibt, die bei immer größer werdendem Exponenten (n) in immer größeren Abständen verstreut in der Menge (N) der natürlichen Zahlen positioniert sind. gibt es unendlich viele zusammengesetzte gerade Zahlen, (composed equal numbers), die (links und rechts) an Primzahlen ($p \in P$) direkt anliegen. Für diese Primzahlen wurde hier die Bezeichnung: $CM_p = n_g -1$ und $CF_p = n_g +1$ gewählt. Da die Primzahlen CF_p direkt im Abstand $\Delta=1$ auf eine gerade Zahl folgen, wurde für diese die Bezeichnung: $n_{g,1} = CF_p -1$ gewählt und für die gerade Zahl, die auf die Primzahl CM_p folgt, die Bezeichnung: $n_{g,2} = CM_p +1$. Sofern man dieser Systematik folgt, ergeben sich aus den Primzahlen F_p und M_p die geraden Zahlen

$n_{g,1} = F_p - 1$ und $n_{g,2} = M_p + 1$. Alle geraden Zahlen $n_g \neq 2^n$ sind Kompositionen vom Typ: $2^n \cdot \Pi p_i$ in denen ungerade Primzahlen fakultativ mit der Primzahl p=2 gekoppelt sind oder aber gerade Zahlen vom Typ: $\Sigma 2^n = 2^n + 2^{n-1} + 2^{n-2} + 2^{n-3} + \ldots + 2^1$ in denen einzelne Summanden: 2^{n-m}, m <n , auch fehlen können; des weiteren z.B. auch die Fakultäten $n_g = n!$. Es gibt offensichtlich unendlich viele Bildungsgesetze für Folgen gerader Zahlen (n_g) aus denen sich fallweise Primzahlen der Typen: $CM_p = n_g - 1$ und $CF_p = n_g + 1$ ergeben.

Dem Beweis des Hauptsatzes der Primzahl-Theorie liegt bekanntlich die nachfolgende Gleichung (1) zugrunde, in welcher in der Fakultät $n_g = \Pi p_i$ (lückenlos) die Primzahlen: $p_1=2$ bis $p_i = p_n$ auftreten, die in dieser Gleichung auch als Potenzen: p_i^n auftreten können. Dabei ist es leicht einsichtig, dass der Hauptsatz der Primzahl-Theorie auch aus den Gleichungen: (2), (3) und (4) beweisbar ist, wobei in den Gleichungen (3) und (4) in den Fakultäten $n_u = \Pi^* p_i$ die Primzahl p=2 nicht vorkommt.

(1) ... $n_u = \Pi p_i + 1$, (2) ... $n_u = \Pi p_i - 1$, (3) .... $n_u = \Pi^* p_i + 2$, (4) ... $n_u = \Pi^* p_i - 2$

Aus allen vier Gleichungen resultieren zwei Schlussfolgerungen:

Schlussfolgerung 1: Die ungeraden Zahlen (n_u) sind Primzahlen vom Typ: $p > p_n$

Schlussfolgerung 2: Sofern die ungeraden Zahlen (n_u) nicht prim sind, sind sie aus der Fakultät: Πp_j zusammengesetzte Zahlen, in der das Kriterium $p_j > p_n$ erfüllt wird.

Die folgenden Beispiele, in denen die Primzahlen fett gedruckt wurden, belegen dies:

Beispiel 1: $n_u = 2^5 + 1 = 32 + 1 = 33 = 3 \cdot 11$, $n_u = 2^5 - 1 = 32 - 1 = \mathbf{31}$

Beispiel 2: $n_u = 2^8 + 1 = 256 + 1 = \mathbf{257}$, $n_u = 2^8 - 1 = 256 - 1 = 255 = \mathbf{3 \cdot 5 \cdot 17}$

Beispiel 3: $n_u = 2 \cdot 3 \cdot 5 + 1 = 30 + 1 = \mathbf{31}$, $n_u = 2 \cdot 3 \cdot 5 - 1 = 30 - 1 = \mathbf{29}$

Beispiel 4: $n_u = \mathbf{3 \cdot 5 \cdot 7 + 2} = 105 + 2 = \mathbf{107}$, $n_u = \mathbf{3 \cdot 5 \cdot 7} - 2 = 105 - 2 = \mathbf{103}$

Beispiel 5: $n_u = \mathbf{3 \cdot 5 \cdot 7 \cdot 11 + 2} = 1155 + 2 = 1157 = \mathbf{13 \cdot 89}$, $n_u = 1155 \mathbf{-2} = \mathbf{1153}$

In der Monographie wird der Nachweis erbracht, dass sich aus diesen geraden Zahlen fallweise Primzahlen der Typen: $CM_p = 2 \cdot \Pi p_i - 1$, $CF_p = 2 \cdot \Pi p_i + 1$ sowie $CM_p = \Sigma 2^n - 1$, $CF_p = \Sigma 2^n + 1$ bzw. $CM_p = n! - 1$, $CF_p = n! + 1$ ergeben. Je nach der Struktur der Bildungsgesetze dieser geraden Zahlen ergeben sich auch hier Klassenteilungen, die zu Dirichlet'schen Progressionen führen.
Diese Dirichlet-Progressionen werden für zahlreiche Bildungs-Folgen der geraden Zahlen abgeleitet.

Aus den Beweisgleichungen (3) und (4) des Hauptsatzes der Primzahl-Theorie folgt, dass es unendlich viele Primzahlen vom Typ: $SM_p = n_u - 2$ und $SF_p = n_u + 2$ gibt, in deren direkter Umgebung im Abstand $\Delta = \pm 2$ ungerade Zahlen (n_u) auftreten, die prim bzw. nicht prim sind. Sofern für eine gleiche ungerade Zahl (n_u) die Primzahlen: $SM_p = n_u - 2$ und $SFp = n_u + 2$ gleichzeitig auftreten, bilden diese einen Semi-Zwilling mod 4, was in Beispiel 4 dokumentiert wurde.

Aus den Beweisgleichungen (1) und (2) des Hauptsatzes der Primzahl-Theorie folgt wiederum, dass es unendlich viele Primzahlen vom Typ: $CM_p = n_g - 1$ und $CF_p = n_g + 1$ gibt. Sofern für eine gleiche gerade Zahl (n_g) die Primzahlen $CM_p = n_g - 1$ und $CF_p = n_g + 1$ gleichzeitig auftreten, bilden diese einen Primzahl-Zwilling mod 2, was in Beispiel 3 dokumentiert wurde.

In dieser Monographie wird die eingangs genannte (bisher noch unbewiesene) 3. Vermutungen (die Vermutung, dass es unendlich viele Zwillinge mod 2 gibt) bewiesen, wobei auch der Nachweis erbracht wird, dass auch die Semi-Zwillinge mod 4 (und alle Semi-Zwillinge mod 2n, n >1) in unendlicher Zahl auftreten. Des weiteren wird für die 4. Vermutung, dass es unendlich viele Mersenne- und Fermat-Primzahlen gibt, auf der Basis des Nachweises der Existenz von sechs Klassen von Dirichlet-Progressionen ein Beweis erbracht.

Die wichtigste Aufgabe, die in dieser Monographie einer Lösung zugeführt werden soll, ist jedoch die Suche nach den Gesetzen, aus denen sich die Schwankungen der Primzahl-Dichten in großen Zahlen-Intervallen ergeben.

Für die Lösung dieser Aufgaben war die Erstellung zahlreicher Tabellen erforderlich, die den Einsatz schneller PC-Rechner und besonderer Test-Programme erfordern, mit deren Hilfe auch die Anlagen erstellt werden konnten, die der Monographie beigefügt sind. Die zahlreichen im Rahmen der Monographie abgeleiteten Tabellen wurden hingegen im Text angeordnet (und kommentiert), da sich aus ihnen wichtige Schlussfolgerungen ergeben.

Der Monographie sind des weiteren 13 (umfangreiche) Anlagen beigefügt, die vom Leser studiert werden können, da diese Anlagen wichtige Ergebnisse von PC- Rechnungen und zahlreiche Teste enthalten, die die Text-Tabellen wesentlich ergänzen. Diese Anlagen enthalten ebenfalls wichtige Schlussfolgerungen. Leser, die die in diesen Anlagen bestimmten Primzahlen und Zwillinge mod 2n, n ≥1 nachvollziehen (überprüfen möchten), sollten zuvor das MuPAD-Pro - System anschaffen.

Derartige PC-Programme und schnelle elektronische Rechner standen den Primzahl-Forschern früher nicht zur Verfügung und dies erklärt, weshalb diesen Forschern (auch den großen Mathematikern) der Weg zur Lösung dieser bis heute unbewiesenen Vermutungen "verbaut" war.

Der Verfasser dankt dem Verleger, Herrn Walter Sander, für zahlreiche Ratschläge und seine kritische Begleitung des Werkes und insbesondere für die Entwicklung zahlreicher PC-Programme, die sich bei der komplizierten Problematik als unverzichtbar erwiesen hatten.

Der Verfasser: Dr. Georg Viktor Cwienk
Bochum, März 2006

2. Einführung:

Alle natürlichen Zahlen ($n \in N$) lassen sich bekanntlich durch die folgende "genetisch-fakultative" Gleichung darstellen:

$$n = 2^{a2} \cdot 3^{a3} \cdot 5^{a5} \cdot 7^{a7} \cdot 11^{a11} \cdot \ldots \cdot p_n^{an} \quad \text{- die "genetisch-fakultative" Zahlengleichung}$$
$$^{a2}, ^{a3}, ^{a5}, ^{a7}, ^{a11}, \ldots, ^{an} := 0, 1, 2, 3, \ldots n$$

darstellen. Sofern man in dieser Gleichung alle Exponenten der ungeraden Primzahlen gleich Null setzt und für Exponenten ^{a2} die Zahlen n:= 1, 2, 3, ... einsetzt, gelangt man zu der Folge der geraden Zahlen $n_g = 2^1=2$, $2^2=4$, $2^3=8$, $2^4=16$, $2^5=32$, $2^6=64$, $2^7=128$, $2^8=256$, $2^9=512$, $2^{10}=1024$, ..., $2^{21}=2097152$, $2^{22}=4194304$, $2^{23}=8388608$, $2^{24}=1677216$, $2^{25}=33554432$, ..., die in der Folge der Zahlen ($n \to \infty$) in immer größeren Abständen verteilt auftreten, wobei die Endziffern dieser geraden Zahlen $n_g = 2^n$ zyklisch die Werte: EZ2, EZ4, EZ8, EZ6 →EZ2, EZ4, EZ8, EZ6 annehmen.

In Anknüpfung an die Ausführungen, die bereits im Rahmen der Vorworte dargestellt wurden. erweisen sich die Mersenne- und Fermat-Primzahlen als Teilmengen der (aller) Primzahlen vom Typ: CM_p und CF_p und es gelten die echten Teilmengen - Relationen (1) und (2):

$$(1) \ldots\ M_p \subset CM_p \quad \text{sowie} \quad (2) \ldots\ F_p \subset CF_p$$

Nachdem die Menge der (aller) geraden Zahlen vom Typ $n_{g,1}$ und $n_{g,2}$ echte Teilmengen der Menge aller geraden Zahlen sind, wird (offensichtlich) auch die nachfolgende Teilmengen - Relation erfüllt:

$$(N_{g,1} + N_{g,2}) \subset N_g$$

Des weiteren gelten die Ungleichungen (3), für $n \to \infty$ und (4) für Teilmengen von n_g. In der Ungleichung (4) wird durch das Symbol: P_n die Zahl der Primzahlen bezeichnet, die im Zahlen - Intervall n =2 bis n auftreten; diese Zahl wird in der Mathematik auch mit $\pi(n)$ bezeichnet.

$$(3) \ldots\ (N_{g,1} + N_{g,2}) < 2 \cdot P \qquad (4) \ldots\ (N_{g,1} + N_{g,2})_n < 2 \cdot (\pi(n) = P_n)$$

Die Ungleichungen: (3) und (4) ergeben sich offensichtlich deshalb, weil in der unendlichen Primzahl - Folge: p = 2, 3, 5, 7, ... , p_n zahlreiche Zwillinge mod 2 auftreten, in denen in den benachbarten Primzahlen (p_i, p_i+2) eine "gemeinsame" gerade Zahl: $n_g = p_i + 1$ situiert ist. Für diese besonderen geraden Zahlen gilt die Identität: $n_{g,1} = n_{g,2}$. Es wurde der Nachweis erbracht, dass diese geraden Zahlen stets nur in drei Folgen situiert sein können, die zu sechs Dirichlet - Progressionen mit definierten Endziffern (EZ) der ungeraden Zahlen $n_u = n_g - 1$ und $n_u = n_g + 1$ führen, in denen die Primzahlen fett gedruckt und die nicht primen ungeraden Zahlen nicht fett notiert wurden:

<u>*Folge 1:*</u> $n_g = 12 + n \times 30$, <u>*Folge 2:*</u> $n_g = 18 + n \times 30$ <u>*Folge 3:*</u> $n_g = 30 + n \times 30$. (n ≥0)

<u>D-Progression 1, (EZ1):</u> $n_u = n_g - 1 = (12 + n \times 30) - 1$:= **11, 41, 71, 101, 131**, 161, **191**, ...
<u>D-Progression 2, (EZ3):</u> $n_u = n_g + 1 = (12 + n \times 30) + 1$:= **13, 43, 73, 103**, 133, **163, 193**, ...

<u>D-Progression 3, (EZ7):</u> $n_u = n_g - 1 = (18 + n \times 30) - 1$:= **17, 47**, 77, **107, 137, 167, 197**, ...
<u>D-Progression 4, (EZ9):</u> $n_u = n_g + 1 = (18 + n \times 30) + 1$:= **19**, 49, **79, 109, 139**, 169, **199**, ...

<u>D-Progression 5, (EZ9):</u> $n_u = n_g - 1 = (30 + n \times 30) - 1$:= **29, 59, 89**, 119, **149, 179**, 209, ...
<u>D-Progression 6, (EZ1):</u> $n_u = n_g + 1 = (30 + n \times 30) + 1$:= **31, 61**, 91, 121, **151, 181**, 211, ...

Durch die Existenz dieser sechs Dirichlet - Progressionen ist der Beweis erbracht, dass Primzahl-Zwillinge mod 2 in der unendlichen Menge der natürlichen Zahlen (N) in einer unendlichen Zahl vorkommen. Damit wurde diese bisher unbewiesene Vermutung bereits unter Beweis gestellt.

In **Anlage** 1 wurden die Zwillinge mod 2 für den Zahlenbereich n:= 2 bis 100.000 vorgestellt und aus **Tab. 2** ist zu entnehmen, dass die Zwillinge mod 2 in den (gekoppelten) Dirichlet - Progressionen: 1&2, 3&4 und 5&6 im Bereich ansteigender Zahlen: $n \to \infty$ spärlicher und in variierender Häufigkeit auftreten. In der vorgestellten Monographie wird auf eine analoge Weise auch der Beweis erbracht dass alle Zwillinge mod 2n, $n \geq 1$ in einer unendlichen Zahl auftreten. Aus diesem Sachverhalt wurde der folgende Grenzwert-Satz deduziert:

$$\lim \Sigma(n_{g,1} + n_{g,2}) = 1.9375 \times \pi(n), \text{ für } n \to \infty$$

Außer den Primzahlen $CM_p = n_g - 1$ und $CF_p = n_g + 1$, kommen in den Umgebungen gerader Zahlen (n_g) Primzahlen vom Typ: $p_1 = n_g$ -3, -5, -7, ... - n_u und vom Typ: $p_2 = n_g$ +3, +5, +7, ... +n_u vor, wobei die geraden Zahlen (n_g) symmetrisch oder asymmetrisch zwischen den Primzahlen (p_1, p_2) situiert sind. Sofern die Primzahlen die Bedingungen: $p_1 = n_g - 3$ und $p_2 = n_g + 3$ erfüllen, bilden diese Primzahlen offensichtlich einen Zwilling mod 6, in dem die gerade Zahl (n_g) symmetrisch zwischen den Primzahlen (p_1, p_2) situiert ist. In analoger Weise sind die geraden Zahlen (n_g) symmetrisch in den Zwillingen mod 10, mod 14 , ... mod $2n_u$ situiert. Es wird an zahlreichen Beispielen demonstriert, dass gerade Zahlen jedoch überwiegend asymmetrisch zwischen den angrenzenden Primzahlen: $p_1 < n_g$ und $p_2 > n_g$ situiert sein können; dann bilden die Primzahlen (p_1, p_2) Semi-Zwillinge mit diversen Abständen $-\Delta_1 = p_1 - n_g$ und $+\Delta_2 = p_2 - n_g$ zu den geraden Zahlen (n_g).

3. Der Beweis der Vermutung, dass es unendlich viele Primzahl-Zwillinge mod 2 gibt:

Für diesen Beweis wird zunächst der Nachweis erbracht, dass es acht (und nicht nur vier) Dirichlet-Progressionen von Primzahlen mit den Endziffern: EZ1, EZ3, EZ7, EZ9 gibt, die jeweils im Abstand von Δ=20 auftreten. Dirichlet hatte (bekanntlich) den Beweis erbracht, dass alle Primzahlen mit den Endziffern: EZ1, EZ3, EZ7, EZ9 im Zahlenbereich $n \to \infty$ in vier Folgen in unendlicher Zahl auftreten.

Für die Aufteilung der ungeraden Primzahlen in acht Klassen ist es erforderlich die Restklassen: Rest 1 und Rest 3 zu analysieren, die sich bei der Division der Primzahlen durch die Zahl 4 ergeben. Man gelangt so zu den Klassenteilungen gemäß **Tab. 1**.

Tab. 1 - Die Folgen der Primzahlen geordnet nach den Endziffern und den Restklassen R=1, R=3:

Klasse 1, EZ1, Rest 1:	π_1:= **41, 61, 101, 181**, ...	alle Primzahlen der Folge: 41 + n x20 mit dem Rest 1
Klasse 2, EZ3, Rest 1	π_2:= **13, 53, 73, 131**, ...	alle Primzahlen der Folge: 13 + n x20 mit dem Rest 1
Klasse 3, EZ7, Rest 1	π_3:= **17, 37, 97, 137**, ...	alle Primzahlen der Folge: 17 + n x20 mit dem Rest 1
Klasse 4, EZ9, Rest 1	π_4:= **29, 89, 109, 139**, ...	alle Primzahlen der Folge: 29 + n x20 mit dem Rest 1
Klasse 5, EZ1, Rest 3	π_5:= 11, 31, 71, 131, ...	alle Primzahlen der Folge: 11 + n x20 mit dem Rest 3
Klasse 6, EZ3, Rest 3	π_6:= 3, 23, 43, 83, 103, ...	alle Primzahlen der Folge: 3 + n x20 mit dem Rest 3
Klasse 7, EZ7, Rest 3	π_7:= 7, 47, 67, 107, 127, ..	alle Primzahlen der Folge: 7 + n x20 mit dem Rest 3
Klasse 8, EZ9, Rest 3	π_8:= 19, 59, 79, 139, ...	alle Primzahlen der Folge: 19 + n x20 mit dem Rest 3

Anmerkungen:

1, Die in **Tab. 1** angeführten Klassen: 1, 2, 3, 4 können zu der Oberklasse I zusammengefasst werden, deren Primzahlen stets den Rest 1 aufweisen - diese Primzahlen wurden fett gedruckt.

2. Die in **Tab. 1** angeführten Klassen: 5, 6, 7, 8 können zu der Oberklasse II zusammengefasst werden, deren Primzahlen stets den Rest 3 aufweisen - diese Primzahlen wurden nicht fett gedruckt.

3. In der **Anlage 2** wurde nachgewiesen, dass die Primzahlen aller acht Klassen in der Zahlenmenge $n \to \infty$ gleichmächtig in einer unendlichen Zahl auftreten und den folgenden gleichen Grenzwerten zustreben:

$$\lim_{n\to\infty} \pi_2(n)/\pi(n) = \lim_{n\to\infty} \pi_2(n)/\pi(n) = \lim_{n\to\infty} \pi_3(n)/\pi(n) = \lim_{n\to\infty} \pi_4(n)/\pi(n) = \pi(n)/8 = 0,125 \cdot \pi(n)$$

$$\lim_{n\to\infty} \pi_5(n)/\pi(n) = \lim_{n\to\infty} \pi_6(n)/\pi(n) = \lim_{n\to\infty} \pi_7(n)/\pi(n) = \lim_{n\to\infty} \pi_8(n)/\pi(n) = \pi(n)/8 = 0,125 \cdot \pi(n)$$

4. Unter Beachtung der in **Tab. 1** eingeführten Notationen der Primzahlen in den Oberklassen I und II, ergibt sich für die Zwillinge mod 2 die Notation: (11, **13**), (**17**, 19), (**29**, 31), (59, **61**), ... und dies besagt, dass in jedem Zwilling mod 2 stets Primzahlen aus den zwei Oberklassen I und II auftreten müssen.

5. In **Anlage 2** wurde des weiteren der Nachweis erbracht, dass die Zahl der Zwillinge mod 2: $\pi^*(n)$ dem folgenden Grenzwert zustrebt:

$$\lim_{n\to\infty} \pi^*(n) = 0,125 \cdot \pi(n)$$ .... Der Grenzwert-Satz der Zwillinge mod 2

Durch die Deduktion dieses Grenzwert-Satzes ist die Vermutung, dass es unendlich viele Zwillingw mod 2 gibt, unter Beweis gestellt worden.

In **Tab. 2** wird der Nachweis erbracht, dass Primzahl-Zwillinge mod 2 nur in den drei Folgen: $n_g = 12 + n \times 30$, $n_g = 18 + n \times 30$, $n_g = 30 + n \times 30$ im Umfeld dieser geraden Zahlen (n_g) auftreten können, nicht aber im Umfeld aller dieser geraden Zahlen vorkommen. Aus **Tab. 2** folgt, das die Dichte der Zwillinge mod 2 in ansteigenden Zahlen-Intervallen zunehmend geringer wird.
Die tieferen Ursachen dieses Sachverhaltes werden in **Tab. 3** analysiert. In dieser Tabelle wurde die Verteilung der Primzahlen im Umfeld der geraden Zahlen $n_g = \Pi p_i$ untersucht.

Tab. 2 - Die Folge der Primzahl-Zwillinge mod 2 in wachsenden Zahlenbereichen (n):

Zwillinge mod 2 in der Folge: $n_g = 12 + n \times 30$:		Zwillinge mod 2:=	Zwilling mod 2: ja / nein:
n = 0	(12 -1) = **11**, (12 +1) = **13**	**(11, 13)**	ja
n = 1	(42 -1) = **41**, (42 +1) = **43**	**(41, 43)**	ja
n = 2	(72 -1) = **71**, (72 +1) = **73**	**(71, 73)**	ja
n = 3	(102 -1) = **101**, (102 +1) = **103**	**(101, 103)**	ja
n = 4	(132 -1) = **131**, (132 +1) = 133	(**131**, 133)	nein
n = 5	(162 -1) = 161, (162 +1) = **163**	(161, **163**)	nein
n = 6	(192 -1) = **191**, (192 +1) = **193**	**(191, 193)**	ja
n = 7	(222 -1) = 221, (222 +1) = **223**	(221, **223**)	nein
n = 8	(252 -1) = **251**, (252 +1) = 253	(**251**, 253)	nein
n = 9	(282 -1) = **281**, (282 +1) = 283	**(281, 283)**	ja
n = 10	(312 -1) = **311**, (312 +1) = **313**	**(311, 313)**	ja
n= 100	(3012 -1) = **3011**, (3012 +1) = 3013	(**3011**, 3013)	nein
n = 101	(3042 -1) = **3041**, (3042 +1) = 3043	(**3041**, 3043)	nein
n = 102	(3072 -1) = 3071, (3072 +1) = 3073	(3071, 3073)	nein
n = 103	(3102 -1) = 3101, (3102 +1) = 3103	(3101, 3103)	nein
n = 104	(3132 -1) = 3131, (3132 +1) = 3133	(3131, 3133)	nein
n = 105	(3162 -1) = 3161, (3162 +1) = **3163**	(3161, **3163**)	nein
n = 106	(3192 -1) = **3191**, (3192 +1) = 3193	(**3191**, 3193)	nein
n = 107	(3222 -1) = **3221**, (3222 +1) = 3223	(**3221**, 3223)	nein
n = 108	(3252 -1) = **3251**, (3252 +1) = **3253**	**(3251, 3253)**	ja
n = 109	(3282 -1) = 3281, (3282 +1) = 3283	(3281, 3283)	nein
n = 110	(3312 -1) = 3311, (3312 +1) = **3313**	(3311, **3313**)	nein
n= 1000	(30012 -1) = **30011**, (30012 +1) = **30013**,	**(30011, 30013)**	ja
n = 1001	(30042 -1) = 30041, (30042 +1) = 30043	(30041, 30043)	nein
n = 1002	(30072 -1) = **30071**, (30072 +1) = 30073	(**30071**, 30073)	nein
n = 1003	(30102 -1) = 30101, (30102 +1) = **30103**	(30101. **30103**)	nein

Tab. 2 - Fortsetzung

Zwillinge mod 2 in der Folge $n_g = 12 + n$ x30:

n = 1004	(30132 -1) = 30131, (30132 +1) = **30133**	(30131, **30133**)	nein
n = 1005	(30162 -1) = **30161**, (30162 +1) = 30163	(**30161**, 30163)	nein
n = 1006	(30192 -1) = 30191, (30192 +1) = 30193	(30191, 30193)	nein
n = 1007	(30222 -1) = 30221, (30222 +1) = **30223**	(30221, **30223**)	nein
n = 1008	(30252 -1) = **30251**. (30262 +1) = 30253	(**30251**, 30353)	nein
n = 1009	(30282 -1) = 30281, (30282 +1) = 30283	(30281, 30283)	nein
n = 1010	(30312 -1) = 30311, (30312 +1) = **30313**	(30311, **30313**)	nein
n = 10000	(300012 -1) = 300011, (300012 +1) = 300013	(300011, 300013)	nein
n = 10001	(300042 -1) = 300041, (300042 +1) = **300043**	(300041, **300043**)	nein
n = 10002	(300072 -1) = 300071, (300072 +1) = **300073**	(300071, **300073**)	nein
n = 10003	(300102 -1) = 300101, (300102 +1) = 300103	(300101, 300103)	nein
n = 10004	(300132 -1) = 300131, (300132 +1) = 300133	(300131, 300133)	nein
n = 10005	(300162 -1) = 300161, (300162 +1) = **300163**	(300161, **300163**)	nein
n = 10006	(300192 -1) = **300191**, (300192 +1) = **300193**	(**300191, 300193**)	ja
n = 10007	(300222 -1) = **300221**, (300222 +1) = 300223	(**300221**, 300223)	nein
n = 10008	(300252 -1) = 300251, (300252 +1) = 300253	(300251, 300253)	nein
n = 10009	(300282 -1) = 300281, (300282 +1) = 300283	(300281, 300283)	nein
n = 30010	(300312 -1) = 300311, (300312 +1) = 300313	(300311, 300313)	nein

Zwillinge mod 2 in der Folge $n_g = 18 + n$ x30:

		Zwilling mod 2:	ja / nein:
n = 0	(18 -1) = **17**, (18 +1) = **19**	(**17, 19**)	ja
n = 1	(38 -1) = **37**, (38 +1) = 39	(**37**, 39)	nein
n = 2	(68 -1) = **67**, (68 +1) = 69	(**67**, 69)	nein
n = 3	(98 -1) = **97**, (98 +1) = 99	(**97**, 99)	nein
n = 4	(128 -1) = **127**, (128 +1) = 129	(**127**, 129)	nein
n = 5	(158 -1) = **157**, (158 +1) = 159	(**157**, 159)	nein
n = 6	(198 -1) = **197**, (198 +1) = **199**	(**197, 199**)	ja
n = 7	(228 -1) = **227**, (228 +1) = **229**	(**227, 229**)	ja
n = 8	(258 -1) = **257**, (258 +1) = 259	(**257**, 259)	nein
n = 9	(288 -1) = 287, (288 +1) = 289	(287, 289)	nein
n = 10	(318 -1) = **317**, (318 +1) = 319	(**317**, 319)	nein
n = 100	(3018 -1) = 3017, (3018 +1) = **3019**	(3017, **3019**)	nein
n = 101	(3048 -1) = 3047, (3048 +1) = **3049**	(3047, **3049**)	nein
n = 102	(3078 -1) = 3077, (3078 +1) = **3079**	(3077, **3079**)	nein
n = 103	(3108 -1) = 3107, (3108 +1) = **3109**	(3107, **3109**)	nein
n = 104	(3138 -1) = **3137**, (3138 +1) = 3139	(**3137**, 3139)	nein
n = 105	(3168 -1) = **3167**, (3168 +1) = **3169**	(**3167, 3169**)	ja
n = 106	(3198 -1) = 3197, (3198 +1) = 3199	(3197, 3199)	nein
n = 107	(3228 -1) = 3227, (3228 +1) = **3229**	(3227, **3229**)	nein
n = 108	(3258 -1) = **3257**, (3258 +1) = **3259**	(**3257, 3259**)	ja
n = 109	(3288 -1) = 3287, (3288 +1) = 3289	(3287, 3289)	nein
n = 110	(3318 -1) = 3317, (3318 +1) = **3319**	(3317, **3319**)	nein
n = 1000	(30018 -1) = 30017, (30018 +1) = 30019	(30017, 30019)	nein
n = 1001	(30048 -1) = **30047**, (30048 +1) = 30049	(**30047**, 30049)	nein
n = 1002	(30078 -1) = 30077, (30078 +1) = 30079	(30077, 30079)	nein
n = 1003	(30108 -1) = 30107, (30108 +1) = **30109**	(30107, **30109**)	nein
n = 1004	(30138 -1) = **30137**, (30138 +1) = **30139**	(**30137, 30139**)	ja
n = 1005	(30168 -1) = 30167, (30168 +1) = **30169**	(30167, **30169**)	nein
n = 1006	(30198 -1) = **30197**, (30198 +1) = 30199	(**30197**, 30199)	nein
n = 1007	(30228 -1) = 30227, (30228 +1) = 30229	(30227, 30229)	nein
n = 1008	(30258 -1) = 30257, (30258 +1) = **30259**	(30257, **302599**	nein
n = 1009	(30288 -1) = 30287, (30288 +1) = 30289	(30287, 30289)	nein
n = 1010	(30318 -1) = 30317, (30318 +1) = **30319**	(30317, **30319**)	nein

Tab. 2 Fortsetzung

Zwillinge mod 2 in der Folge $n_g = 18 + n$ x30:

n = 10000	(300018 -1) = **300017**, (300018 +1) = 300019	(**300017**, 300019)	nein
n = 10001	(300048 -1) = 300047, (300048 +1) = 300049	8300047, 300049)	nein
n = 10002	(300078 -1) = 300077, (300078 +1) = 300079	(300077, 300079)	nein
n = 10003	(300108 -1) = 300107, (300108 +1) = **300109**	(300107, **300109**)	nein
n = 10004	(300138 -1) = **300137**, (300138 +1) = 300139	(**300137**, 300139)	nein
n = 10005	(300168 -1) = 300167, (300168 +1) = 300169	(300167, 300169)	nein
n = 10006	(300198 -1) = 300197, (300198 +1) = 300199	(300197, 300199)	nein
n = 10007	(300228 -1) = 300227, (300228 +1) = **300229**	(300227, **300229**)	nein
n = 10008	(300258 -1) = 300257, (300258 +1) = 300259	(300257, 300259)	nein
n = 10009	(300288 -1) = 300287, 8300288 +1) = 300289	(300287, 300289)	nein
n = 10010	(300318 -1) = **300317**, (300318 +1) = **300319**	(**300317, 300319**)	ja

Zwillinge mod 2 in der Folge $n_g = 30 + n$ x30:		Zwilling mod 2:=	ja / nein:
n = 0	(30 -1) = **29**, (30 +1) = **31**	(**29, 31**)	ja
n = 1	(60 -1) = **59**, (60 +1) = **61**	(**59, 61**)	ja
n = 2	(90 -1) = **89**, (90 +19 =91	(**89**, 91)	nein
n = 3	(120 -1) = 119, (120 +1) = 121	(119, 121)	nein
n = 4	(150 -1) = **149**, (150 +1) = **151**	(**149, 151**)	ja
n = 5	(180 -1) = **179**, (180 +1) = **181**	(**179, 181**)	ja
n = 6	(210 -1) = 209, (210 +1) = **211**	(209, **211**)	nein
n = 7	(240 -1) = **239**, (240 +1) = **241**	(**239, 241**)	ja
n = 8	(270 -1) = **269**, (270 +1) = **271**	(**269, 271**)	ja
n = 9	(300 -1) = 299, (300 +1) = 301	(299, 301)	nein
n = 10	(330 -1) = 329, (330 +1) = **331**	(329, **331**)	nein
n = 100	(3030 -1) = 3029, (3030 +1) = 3031	(3029, 3031)	nein
n = 101	(3060 -1) = 3059, (3060 +1) = **3061**	(3059, **3061**)	nein
n = 102	(3090 -1) = **3089**, (3090 +1) = 3091	(**3089**, 3091)	nein:
n = 103	(3120 -1) = **3119**, (3120 +1) = **3121**	(**3119, 3121**)	ja
n = 104	(3150 -1) = 3149, (3150 +1) = 3151	(3149, 3151)	nein
n = 105	(3180 -1) = 3179, (3180 +1) = **3181**	(3179, **3181**)	nein
n = 106	(3210 -1) = **3209**, (3210 +1) = 3211	(**3209**, 3211)	nein
n = 107	(3240 -1) = 3239, (3240 +1) = 3241	(3239, 3241)	nein
n = 108	(3270 -1) = 3269, (3270 +1) = **3271**	(3269, **3271**)	nein
n = 109	(3300 -1) = **3299**, (3300 +1) = **3301**	(**3299, 3301**)	ja
n = 110	(3330 -1) = **3329**, (3330 +1) = **3331**	(**3329, 3331**)	ja
n = 1000	(30030 -1) = **30029**, (30030 +1) = 30031	(**30029**, 30031)	nein
n = 1001	(30060 -1) = **30059**, (30060 +1) = 30061	(**30059**, 30061)	nein
n = 1002	(30090 -1) = **30089**, (30090 +1) = **30091**	(**30089, 30091**)	ja
n = 1003	(30120 -1) = **30119**, (30120 +1) = 30121	(**30119**, 30121)	nein
n = 1004	(30150 -1) = 30149, (30150 +1) = 30151	(30149, 30151)	nein
n = 1005	(30180 -1) = 30179, (30180 +1) = **30181**	(30179, **30181**)	nein
n = 1006	(30210 -1) = 30209, (30210 +1) = **30211**	(30209, **30211**)	nein
n = 1007	(30240 -1) = 30239, (30240 +1) = **30241**	(30239, **30241**)	nein
n = 1008	(30270 -1) = **30269**, (30270 +1) = **30271**	(**30269, 30271**)	ja
n = 1009	(30300 -1) = 30299, (30300 +1) = 30301	(30299, 30301)	nein
n = 1010	(30330 -1) = 30329, (30330 +1) = 30331	(30329, 30331)	nein
n = 10000	(300030 -1) = 300029, (300030 +1) = 300031	(300029, 300031)	nein
n = 10001	(300060 -1) = 300059, (300060 +1) = 300061	(300059, 300061)	nein
n = 10002	(300090 -1) = 300089, (300090 +1) = 300091	(300089, 300091)	nein
n = 10003	(300120 -1) = **300119**, (300120 +1) = 300121	(**300119**, 300121)	nein
n = 10004	(300150 -1) = **300149**, (300150 +1) = **300151**	(**300149, 300151**)	ja
n = 10005	(300180 -1) = 300179, (300180 +1) = 300181	(300179, 300181)	nein
n = 10006	(300210 -1) = 300209, (300210 +1) = 300211	(300209, 300211)	nein

Tab. 2 Fortsetzung

Zwillinge mod 2 in der Folge $n_g = 30 + n \times 30$:

n = 10007	(300240 -1) = **300239**, (300240 +1) = 300241	(**300239**, 300241)	nein
n = 100008	(300270 -1) = 300269, (300270 +1) = 300271	(300269, 300271)	nein
n = 100009	(300300 -1) = **300299**, (300300 +1) = **300301**	(**300299, 300301**)	ja
n = 100010	(300330 -1) = 300329, (300330 +1) = **300331**	(300329, **300331**)	nein

Anmerkungen:

1. Die **Tab. 2** auftretenden Primzahlen wurden fett und alle nicht primen Zahlen nicht fett gedruckt.

2. In den Dekaden: 10, 10^2, 10^3, 10^4 wurde für die Folgen: $n_g = 12 + n \times 30$, $n_g = 18 + n \times 30$, $n_g = 30 + n \times 30$ das Auftreten von Zwillingen mod 2 für 11 aufeinander folgende Zahlen (n) getestet.

3. Aus **Tab. 2** geht hervor, dass die Zahl der Zwillinge mod 2 von Dekade zu Dekade geringer wird, dennoch aber ein variierendes Verhalten aufweist. Der tiefere Grund für die "Dichte-Abnahme" der Zwillinge mod 2 in immer größer werdenden Zahlen-Intervallen ($n \to \infty$) wird aus den (noch anzustellenden) Untersuchungen plausibel. Vorweg gesagt, können in primzahlfreien Zahlen-Intervallen (Δn) auch keine Zwillinge mod 2 auftreten und darüber hinaus treten in den Umgebungen dieser primzahlfreien Intervalle auch noch gewisse "Sperr-Zonen" für Zwillinge mod 2 auf.

4. Die Zahl der Zwillinge mod 2, die in den (jeweils) 44 Testungen der D-Progressionen mit den Endziffern: EZ1 & EZ3, EZ7 & EZ9, EZ9 & EZ1 auftreten, ergibt die Werte: 10 (für die Endziffern EZ1 & EZ3), 7 (für die Endziffern EZ7 & EZ9) und 13 (für die Endziffern EZ9 & EZ1).. Aus **Anlage 1** kann jedoch entnommen werden, dass die Zwillinge mod 2 in großen Zahlen-Intervallen in den D-Progressionen zahlenmäßig gleich verteilt auftreten.

5. Es gilt noch der wichtige Hinweis, dass in **Tab. 1** der Primzahl-Drilling: (3, 5, 7) mit den verketteten Zwillingen mod 2: (3, 5) und (5, 7) nicht angeführt wurde. Inmitten des Zwillings mod 2 : (3, 5) ist die gerade Zahl $n_g = 4$ situiert, die in mehrstelligen Zwillingen mod 2 nicht auftreten kann, da die ungeraden Nachfolger-Zahlen $n_u = EZ4 + 1$ die Endziffer EZ5 annehmen und somit nicht prim sein können. Inmitten des Zwillings mod 2: (5, 7) ist wiederum die gerade Zahl $n_g = 6$ situiert, die in mehrstelligen Zwillingen mod 2 nicht auftreten kann, da die ungeraden Vorgänger-Zahlen $n_u = EZ6 - 1$ die Endziffer EZ5 haben und somit nicht prim sein können.

Die Analyse der Ursachen für die Abnahme der Dichten der Zwillinge mod 2 in großen Zahlenbereichen:

In **Tab. 3** wurden die Verteilungen der Primzahlen in den Umgebungen der Fakultäten: $n_g = \Pi p_i$ ($p_i = 2$ bis $p_i = p_n$) bestimmt und die Ergebnisse näher kommentiert. Die Erstellung von **Tab. 3** ist mit einem hohen Zeitaufwand verbunden und erfordert zahlreiche PC-Primzahl - Prüfungen, die mit einem hohen Zweitaufwand verbunden waren. Dieser Aufwand hat sich jedoch sehr gelohnt, da aus den erzielten Ergebnissen wichtige Schlussfolgerungen abgeleitet werden können, die im Anschluss an **Tab. 3** präzisiert werden. Der Leser dieser Monographie kann die in **Tab. 3** ausgewiesenen Ergebnisse mit Hilfe von PC-Testen (prüfend) nachvollziehen. Für die Lösung dieser Aufgabe eignet sich (besonders) das MUPAD - Testverfahren.

Tab. 3 - Die Verteilung der Primzahlen in den Umgebungen der Zahlen $n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot \ldots \cdot 181$

$n_g = \Pi p_i :=$	$\Delta_1 =$	$\Delta_2 =$	$\Delta_3 =$	$\Delta_4 =$	$\Delta_5 =$	$\Delta_6 =$	$\Sigma\Delta :=$
$n_g = 2 \cdot 3 = 6$	---	-3	-1	+1	+5	+7	10
$n_g = 2 \cdot 3 \cdot 5 = 30$	-11	-7	-1	+1	+7	+11	22
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 = 210$	-17	-13	-11	+1	+13	+17	34
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 = 2310$	-17	-13	-1	+1	+23	+29	46
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 13 = 30030$	-19	-17	-1	+17	+29	+41	60
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 17 = 510510$	-53	-47	-29	+19	+41	+43	96
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 19 = 9699690$	-41	-37	-23	+23	+37	+41	82
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 23 = 223092870$	-79	-61	-43	+37	+47	+53	132
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 29 = 6469693230$	-73	-67	-41	+61	+89	+97	170
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 31 = 200560490130$	-83	-79	-73	+1	+67	+83	166
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 37 = 7420738134810$	-139	-107	-59	+61	+101	+103	242
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 41 = 304250263527210$	-53	-47	-1	+71	+107	+151	204
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 43 = 13082761331670030$	-149	-109	-89	+47	+67	+71	220

Tab. 3 - Fortsetzung

$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 47 = 614889782588491410$	-151	-97	-67	+107	+109	+181	332
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 53 = 32589158477190044730$	-127	-89	-73	+59	+73	+149	276
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 59 = 1922760350154212639070$	-163	-109	-107	+61	+89	+101	264
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 61 = 117288381359406970983270$	-241	-223	-89	+109	+167	+179	420
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 67 = 7858321551080267055879090$	-113	-107	-101	+89	+139	+151	264
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 71 = 557940830126698960967415390$	-199	-179	-127	+103	+229	+283	482
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 73 = 40729680599249024150621323470$	-313	-257	-97	+79	+163	+193	506
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 79 = 3217644767340672907899084554130$	-127	-101	-83	+151	+193	+241	368
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 83 = 267064515689275851355624017992790$	-263	-197	-89	+197	+269	+487	750
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 89 = 23768741896345550770650537601358310$	-101	-97	-1	+101	+157	+163	264
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 97 = 2305567963945518424753102147331756070$	-311	-257	-251	+233	+293	+397	708
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 101 = 232862364358497360900063316880507363070$	-149	-139	-131	+233	+523	+647	796
$n_g = 2^2 \cdot 3 \cdot 5 \cdot \ldots \cdot 101 = 465724728716994721800126633761014726140$	-199	-197	-151	+1	+131	+313	512
$n_g = 2^3 \cdot 3 \cdot 5 \cdot \ldots \cdot 101 = 931449457433989443600253267522029452280$	-311	-251	-107	+103	+157	+173	442
$n_g = 2 \cdot 3 \cdot \ldots \cdot 11^2 \cdot \ldots \cdot 101 = 2561486007943470969900696485685580993770$	-397	-239	-229	+431	+479	+503	900
$n_g = 2^3 \cdot 3^2 \cdot 5 \cdot \ldots \cdot 101 = 2794348372301968330800759802566088356840$	-277	-257	-173	+223	+257	+293	570
$n_g = 2^3 \cdot 3^2 \cdot \ldots \cdot 11^2 \cdot \ldots \cdot 101 = 30737832095321651838808357828226971925240$	-307	-227	-1	+251	+349	+353	660
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 103 = 23984823528925228172706521638692258396210$	-631	-389	-293	+223	+243	+251	882
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 107 = 2566376117594999414479597815340071648394470$	-211	-197	-151	+127	+157	+229	440
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 109 = 279734996817854936178276161872067809674997230$	-313	-273	-263	+223	+251	+277	590
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 113 = 31610054640417607788145206291543662493274686990$	-277	-257	-251	+191	+193	+239	516
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 127 = 4014476939333036189094441199026045136645885247730$	-373	-257	-223	+163	+179	+271	644
$n_g = 2 \cdot 3 \cdot \ldots \cdot 13^2 \cdot \ldots \cdot 127 = 52188200211329470458227735587338586776396508220490$	-281	-199	-1	+281	+283	+439	720

Tab. 3 - Fortsetzung

$n_g = 2 \cdot 3 \cdot 5 \cdot .. \cdot 131$ = 525896479052627740771371797072411912900610967452630
-397 -223 -179 +229 +383 +509 906

$n_g = 2 \cdot 3 \cdot 5 \cdot .. \cdot 137$ = 72047817630210000485677936198920432067383702541010310
-523 -449 -389 +643 +647 +827 1350

$n_g = 2 \cdot 3 \cdot 5 \cdot ... \cdot 139$ = 10014646650599190067509233131649940057366334653200433090
-419 -373 -281 +239 +311 +367 786

$n_g = 2 \cdot 3 \cdot 5 \cdot .. \cdot 149$ = 1492182350939279320058875736615841968547583863326864530410
-373 -233 -151 +157 +223 +239 612

$n_g = 2 \cdot 3 \cdot 5 \cdot .. \cdot 151$ = 225319534991831177328890236228992001350685163362356544091910
-541 -463 -197 +167 +317 +331 845

$n_g = 2 \cdot 3 \cdot 5 \cdot .. \cdot 157$ = 35375166993717494840635767087951744212057570647889977422429870
-491 -479 -173 +509 +541 +601 1092

$n_g = 2 \cdot 3 \cdot 5 \cdot ... \cdot 163$ = 5766152219975951659023630035336134306565384015606066319856068810
-431 -409 -239 +239 +457 +773 1204

$n_g = 2 \cdot 3 \cdot 5 \cdot .. \cdot 167$ = 962947420735983927056946215901134429196419130606213075415963491270
-439 -257 -233 +199 +211 +313 752

$n_g = 2 \cdot 3 \cdot 5 \cdot .. \cdot 173$ = 166589903787325219380851695350896256250980509594874862046916839897 10
-433 -409 -191 +191 +503 +823 1256

$n_g = 2 \cdot 3 \cdot 5 \cdot ... \cdot 179$ = 29819592777931214269172453467810429868925511217482600306406141441580 90
-967 -383 -223 +199 +251 +331 1298

$n_g = 2 \cdot 3 \cdot 5 \cdot .. \cdot 181$ = 5397346292805549782720214077673687806275517530364350655459511599582614290
-367 -317 -223 +383 +479 +523 890

<u>Anmerkung:</u>
In der Fortsetzung von **Tab. 3** wird die Lokalisierung der (hier untersuchten) geraden Zahlen $n_g = \Pi p_i$ innerhalb von Zwillingen mod 2n, n ≥1 dargestellt. Die Zahlen $n_g = \Pi p_i$ sind (wie ersichtlich) in diesen Zwillingen in der Regel asymmetrisch situiert und fallweise relativ symmetrisch. Aus der <u>Fortsetzung 1</u> von **Tab. 3** kann entnommen werden, dass die ansteigenden Zahlen $n_g = \Pi p_i$ in zunehmend größeren Zwillingen situiert sind. In der Fortsetzung 1 von Tab. 3 wurden auch die mittleren Abstände: $\Sigma \Delta/5$ angegeben, die variierend eine steigende Tendenz haben.

Tab. 3 - <u>Fortsetzung 1</u>

$n_g = \Pi p_i$:=	Zwilling mod:=	$\Sigma\Delta$:=	$\Sigma\Delta/5$:=
$2 \cdot 3 \cdot 5$	2	22	4,4000
$2 \cdot 3 \cdot 5 \cdot 7$	12	34	6,8000
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11$	2	46	9,2000
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13$	18	60	12,000
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17$	48	96	19,200
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19$	46	82	16,400
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23$	80	132	26,400
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29$	102	170	34,000
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31$	72	166	33,200
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37$	120	242	40,333
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41$	72	204	40,800
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot .. \cdot 43$	136	220	44,000
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot .. \cdot 47$	174	332	66,40
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot .. \cdot 53$	132	276	55,20
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot .. \cdot 59$	168	264	52,800
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot .. \cdot 61$	198	420	84,000

$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 67$	190	264	52,800
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 71$	230	482	96,400
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 73$	176	506	101,200
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 79$	280	368	73,600
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 83$	286	750	150,000
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 89$	102	264	52,800
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 97$	484	708	141,600
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 101$	364	796	159,200
$2^2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 101$	152	512	102,400
$2^3 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 101.$	210	442	88,400
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11^2 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 101$	660	900	180,000
$2^3 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 101.$	396	570	114,000
$2^3 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11^2 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 28 \cdot 31 \cdot 37 \cdot \ldots \cdot 101$	352	660	132,000
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 103$	516	882	176,400
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \ldots \cdot 107$	278	440	88,000
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots 109$	486	590	118,000
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 113$	442	516	123,200
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 127$	386	644	128,800
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 131.$	282	720	144,000
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 137$	1032	1350	270,000
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 139$	520	786	157,200
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 149$	308	612	122,400
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 151$	364	872	174,400
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 157$	612	1032	206,400
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 163$	478	1212	242,400
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 167$	432	752	150,400
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 173..$	382	1256	251,200
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \ldots \cdot 179$	422	1298	259,600
$2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot \ldots \cdot 181$	606	890	148,333

Anmerkungen und Schlussfolgerungen:

1. In **Tab. 3** wurden jeweils sechs Primzahlen ermittelt, die in der direkten Umgebung der geraden Zahlen $n_g = \Pi p_i$ auftreten - die Primzahlen: $p_1 < p_2 < p_3 < n_g$ und die Primzahlen: $p_6 > p_5 > p_4 > n_g$.
Die Bestimmung dieser (jeweils sechs) Primzahlen, die in den "primen Abständen Δ" auftreten, erfordert den Einsatz geeigneter PC- Testverfahren (z.B. der MuPAD- Faktor - Zerlegung).

2. Die Abstände in denen diese Primzahlen gegenüber der geraden Zahl (n_g) auftreten, wurden mit den Symbolen: $\Delta_1 = p_1 - n_g$, $\Delta_2 = p_2 - n_g$, $\Delta_3 = p_3 - n_g$ sowie $\Delta_4 = p_4 - n_g$, $\Delta_5 = p_5 - n_g$, $\Delta_6 = p_6 - n_g$ bezeichnet. Am rechten Rand der Tabelle wurde das Intervall: $\Sigma\Delta = \Delta_1 + \Delta_2 + \Delta_3 + \Delta_4 + \Delta_5 + \Delta_6$ ausgewiesen, in dem diese sechs Primzahlen situiert sind. Daraus ist ersichtlich, dass dieses Intervall mit steigender Fakultät ($p = 2$ bis p_n) eine steigende Tendenz zeigt - anders formuliert treten zunehmend geringere "Primzahl-Dichten" auf. Damit verbunden ist (wie ersichtlich) auch eine Abnahme der Dichten der Zwillinge mod 2, die in **Tab. 3** durch gestrichelte Linien markiert wurden.

3. Aus **Tab. 3** ergibt sich die wichtige Erkenntnis, dass alle (sechs) Primzahlen, die im Intervall $\Sigma\Delta$ auftreten, "Primzahl-Abstände" aufweisen, die stets größer sind als die größte Primzahl (p_n), die in der Fakultät $n_g = \Pi p_i$ vorkommt. Diese Gesetzmäßigkeit steht im Einklang mit dem Hauptsatz der Primzahl-Theorie - bildet jedoch eine wichtige Erweiterung dieses Hauptsatzes (!).

4. Aus **Tab. 3** kann ein wichtiger (bisher in der Primzahl-Forschung unbekannter) "Prozess der Reduktion und der Genese von Zwillingen mod 2" wahrgenommen werden. Zu einer gewissen Reduktion der Zahl der Zwillinge mod 2 führt der Umstand, dass in den Zahlen-Intervallen $\Sigma\Delta$ nicht alle Primzahlen (= Δ) auftreten, die in der Folge: $p_j > p_n$ auftreten und dort Zwillinge mod 2 bilden.
Dies ist der tiefere Grund für die Reduktion der Dichte der Zwillinge mod 2 in den analysierten Zahlen-Intervallen in den Umgebungen der Fakultäten $n_g = \Pi p_i$. Auf der anderen Seite können in den direkten Umgebungen der Zahlen $n_g = \Pi p_i$ neue Zwillinge mod 2 kreiert werden, die vom Typ: ($n_g - 1$, $n_g + 1$) sind. Auch diese (neu kreierten) Zwillinge mod 2 wurden in **Tab. 3** gestrichelt markiert.

5. In **Tab. 3** wurden drei "generierte" Zwillinge mod 2 ausgewiesen, die vor den Zahlen $n_g = \Pi p_i$ noch

nicht vorkommen. Es sind dies die Zwillinge:

$(n_g = 6 - 1 = 5,\ n_g = 6 + 1 = 7)$,
$(n_g = 30 - 1 = 29,\ n_g = 30 + 1 = 31)$ und
$(n_g = 2310 - 1 = 2309,\ n_g = 2310 + 1 = 2311)$,

die allesamt in der Folge: $n_g = 30 + n \times 30$ situiert sind. In **Tab. 3** wurden die Abstände $\Delta = \pm 1$ fett gedruckt hervorgehoben. Das Auftreten von Zwillingen mod 2 des Typs: $n_g \pm 1 = \Pi p_i \pm 1$ erweitert offensichtlich die Gleichung, die dem Beweis des Hauptsatzes der Primzahl - Theorie zugrunde liegt. Alle Zwillinge mod 2 vom Typ: $(\Pi p_i - 1,\ \Pi p_i + 1)$ sind Elemente der Folge $n_g = 30 + n \times 30$.

6. Die in **Tab. 3** des weiteren auftretenden acht Zwillinge mod 2 sind Elemente der nachstehend ausgewiesenen Folgen:

- der Zwilling mod 2: (30011, 30013) ist in der Folge: $n_g = 12 + 30n$, n = 1000 situiert,
- der Zwilling mod 2: (510551, 510553) ist in der Folge $n_g = 12 + 30n$, n = 17018 situiert,
- der Zwilling mod 2: (7420738134911, 7420738134913) ist in der Folge: $n_g = 12 + 30n$ situiert,
- der Zwilling mod 2: (614889782588491517, 614889782588491519), liegt in Folge $n_g = 18 + 30n$
- der Zwilling mod 2: (1922760350154212638961, 1822760350154212638963) liegt in der Folge 12 + 30n
- der Zwilling mod 2: (465724728716994721800126633761014725941, 465724728716994721800126633761014725943) ist in der Folge 12 + 30n situiert.
- der Zwilling mod 2 (31610054640417607788145206291543662493274687181, 31610054640417607788. 145206291543662493274687183) ist Element der Folge 12 + 30n
- der Zwilling mod 2: (52188200211329470458227735587338586776396508220771, 52188200211329470 458227735587338586776396508220773) ist Element der Folge 12 + 30n.

Anmerkung:
Die Werte (n), die für die großen Zwillinge mod 2 gelten, kann der Leser leicht selbst bestimmen.

7. Aus **Tab. 3** können sehr wichtige Erkenntnisse abgeleitet werden:

Erkenntnis 1:
Alle Primzahlen, die der geraden Zahl $n_g = \Pi p_i$ vorausgehen bzw. auf diese Zahl folgen, können nur in Primzahl-Abständen: $\pm \Delta = p_j > p_n$ auftreten. Sofern man die Folgen der Primzahlen $p_j < n_g$ und $p_j > p_n > n_g$ beliebig erweitert (was der Leser durch Teste nachvollziehen kann), treten stets (ausschließlich) nur Primzahl-Abstände auf, die das Kriterium: $\pm\Delta = p_j > p_n$ erfüllen.

Erkenntnis 2:
In der Folge der Primzahl-Abstände ($\pm\Delta$) treten nicht alle Primzahlen: $p_j > p_n$ auf. Dieser Sachverhalt kann nur so gedeutet werden, dass in den Umgebungen der geraden Zahlen $n_g = \Pi p_i$ Semi - Zwillinge mod 2n, $n \geq 1$ auftreten, deren Abstand größer ist, als in der (primären) Folge der Primzahlen: $p_j > p_n$. .

Erkenntnis 3:
Die dritte Erkenntnis, die aus **Tab. 3** gewonnen werden kann, besagt, dass die Primzahl-Dichten in den Zahlen-Intervallen, die von der Zahl $n_g =: \Pi p_i = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 101$ um die Faktoren n:= 2, 4, 11, 12, ... weiter entfernt situiert sind, ähnliche (aber variierende) Werte aufweisen. Dabei erweist es sich, dass auch in den Umgebungen dieser geraden Zahlen, die im doppelten bzw. im zwölf mal so großen Abstand (Zahlen-Intervall) situiert sind, ausschließlich Primzahl-Abstände ($\Delta_1, \Delta_2, \Delta_3, \Delta_4, \Delta_5, \Delta_6$) auftreten können. Die durch den Summen-Abstand ($\Sigma\Delta$) definierte Primzahl - Dichte weist dabei eine fallende (jedoch variable) Tendenz auf.

Erkenntnis 4:
Aus dem Faktum, dass zahlreiche Primzahlen der Folge p:= 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 191, 103, 107, 109, 113,, .., 181. in den Abständen (Δ) nicht auftreten können, resultiert die Abnahme der Primzahl - Dichten in steigenden Intervallen (n).

8. Aus der Fortsetzung 1 von **Tab. 3** ergeben sich weitere (wichtige) Erkenntnisse, die zu bisher nicht bekannten Gesetzen über die Genese der Primzahl-Dichten in allen Zahlen-Bereichen ($n \to \infty$) führen.

Erkenntnis 5:

Erkenntnis 5:
Im direkten Umfeld der Primzahl-Fakultäten: $n_g = \Pi p_i(p = 2 \text{ bis } p_n)$ treten stets die (örtlich) geringsten Primzahl-Dichten auf. Links und rechts von diesen in der Fortsetzung 1 ausgewiesenen Zwillingen mod 2n, $n \geq 1$ treten stets Zwillinge geringerer Modularität auf und damit steigende Primzahl-Dichten.

Erkenntnis 6:
Wiewohl die Primzahl-Dichten mit zunehmend steigenden Werten der Fakultäten $n_g = \Pi p_i$ eine fallende Tendenz aufweisen, variieren diese Primzahl-Dichten fallweise. Die tieferen Ursachen für diese Variation der Primzahl-Dichten können aus der Fortsetzung 3 von **Tab. 3** erkannt werden.

9. Wichtige Erkenntnisse gewinnt man aus der Fortsetzung 2 von **Tab. 3**, in der demonstriert wird, dass die Addition bzw. die Subtraktion von Primzahlen $p < p_n$ zu den Fakultäten Πp_i ($p = 2$ bis p_n) niemals zu Primzahlen führen kann. In den Fakultäten der nicht primen Zahlen: $\Pi p_i \pm (p < p_n)$ treten nämlich die Primzahlen $p < p_n$ stets fakultativ auf. Sofern man hingegen zu den Fakultäten $\Pi p_i(p = 2 \text{ bis } p_n)$ Primzahlen $p > p_n$ addiert bzw. subtrahiert und sich dabei (im Ergebnis) nicht prime Zahlen vom Typ: $\Pi p_i \pm (p > p_n)$ ergeben, sind diese nicht primen Zahlen aus Fakultäten bedeutend größerer Primzahlen konstituiert. Um die Tragweite dieser Gesetzmäßigkeiten "abzusichern" werden in der Fortsetzung 2 von **Tab. 3** die großen Fakultäten: $n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot \ldots \cdot 157$ und $n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot \ldots \cdot 163$ analysiert.

Tab. 3 - Fortsetzung 2

$n_g = \Pi p_i \pm p :=$

$2 \cdot 3 \cdot 5 \cdot \ldots \cdot 97 - 97 = 97 \cdot 2376874189634550770650537601358309$

Anmerkung: Der zweite Primzahl-Faktor ist gleich der Primzahl: $2 \cdot 3 \cdot 5 \cdot \ldots \cdot 89 - 1$ (siehe **Tab. 3**, Seite 13)

$2 \cdot 3 \cdot 5 \cdot \ldots \cdot 97 + 97 = 97 \cdot 131 \cdot 1039 \cdot 2719 \cdot 6422589188429437337180614 1$

$2 \cdot 3 \cdot 5 \cdot \ldots 157 + 3 = 3^8 \cdot 1907718880931972625735437 \cdot 28262728263476674857642056599 04789$

$2 \cdot 3 \cdot 5 \ldots \cdot 157 + 17 = 17 \cdot 2528537287 \cdot 7047324420078603216935369 \cdot 1167766333592092305457 09937$

$2 \cdot 3 \cdot 5 \cdot \ldots \cdot 157 + 151 = 151 \cdot 15808987 \cdot 63744878632519379 \cdot 2324728342672159128796177692 30477827$

$2 \cdot 3 \cdot 5 \cdot \ldots \cdot 163 + 149 = 149 \cdot 28933 \cdot 115758309 \cdot 30640504837 \cdot 373871753276922144673937189525 047747519$

$2 \cdot 3 \cdot 5 \cdot \ldots \cdot 163 + 151 = 151 \cdot 20143859 \cdot 18956863500130286396279227816147833831244161 63079665629$

$2 \cdot 3 \cdot 5 \cdot \ldots \cdot 163 + 157 = 157 \cdot 773473 \cdot 101700283 \cdot 62753998511 \cdot 74400821012198473683067799179 38159719$

$2 \cdot 3 \cdot 5 \cdot \ldots \cdot 163 - 157 = 157 \cdot 949203835243414001 \cdot 3869251560098277718549142943411359337 53775329$

$2 \cdot 3 \cdot 5 \cdot \ldots \cdot 163 + 151 \; 157 = 151 \cdot 157 \cdot 2053 \cdot 236287 \cdot 50139583920036705626327904216374085517728 5630121421$

$2 \cdot 3 \cdot 5 \cdot \ldots \cdot 163 - 151 \cdot 157 = 151 \cdot 157 \cdot 18082283919903547251512813436200705932388606 64533879$

$2 \cdot 3 \cdot 5 \cdot \ldots \cdot 163 + 241 = 6303912182400251940684 53 \cdot 914694249084614470711948476657207 4375967$

$2 \cdot 3 \cdot 5 \cdot \ldots \cdot 163 - 241 = 2897 \cdot 19903873731363312595870314240028078379583652107718558 2321577$

Anmerkung: Aus der Fortsetzung 2 von **Tab. 3** gehen zwei wichtige Gesetze hervor:

Gesetz 1: Bei der Addition bzw. Subtraktion von Primzahlen $p < p_n$ zu der Fakultät Πp_i ($p = 2$ bis p_n) ergeben sich stets nicht prime Zahlen, in deren Primzahl-Fakultäten die Primzahlen ($p < p_n$) stets (solitär) auftreten. Dieses Gesetz gilt auch für die Fakultäten von Primzahlen ($p < p_n$)

Gesetz 2: Bei der Addition bzw. Subtraktion von Primzahlen $p > p_n$ zu der Fakultät Πp_i ($p = 2$ bis p_n) können die Primzahlen: $p > p_n$ nicht in den Fakultäten der nicht primen Zahlen $\Pi p_i \pm (p > p_n)$ (solitär) auftreten. Sofern die Zahlen $\Pi p_i + p$ bzw. $\Pi p_i - p$ keine Primzahlen bilden, erscheinen in ihren Fakultäten Primzahlen, die wesentlich größer sind als die (addierten bzw. subtrahierten) Primzahlen ($p > p_n$).

10. In der Fortsetzung 3 von **Tab. 3** wird demonstriert, dass (stets) einige der Primzahlen $p > p_n$ zu Nachfolger-Primzahlen $p >> p_n$ "fusionieren". In der Fortsetzung 3 von **Tab. 3** wurden die Primzahlen aufgelistet, die größer als die Primzahlen: $p_n = 157$ bzw. $p_n = 163$ sind. die als größte Primzahlen in den Fakultäten: $\Pi(2 \cdot 3 \cdot 5 \cdot 7 \cdot .. \cdot 157)$ bzw. $\Pi(2 \cdot 3 \cdot 5 \cdot 7 \cdot .. \cdot 163)$ auftreten. Diejenigen Primzahlen dieser Folge, die addiert bzw. subtrahiert von Πp_i zu neuen Primzahlen "fusionieren", wurden zusammen mit dem (jeweiligen Vorzeichen versehen) fett gedruckt hervorgehoben.

Tab. 3 - Fortsetzung 3 (Folge der Primzahlen $p > 157$)

163, 167, **-173**, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 437, 439, 443, 449, 457, 461, 463, 467, 479, **-487**, **-491**, 499, 503, **+509**, 521 523, **+541**, 547, 557, 563, 571, 577, 587, 593, 599, **+601**, 607, 613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, **+683**, 691,

Tab. 3 - Fortsetzung 3 (Folge der Primzahlen $p > 163$)

+167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, **±239**, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, **-409**, 419, 421, 433, **-439**, 443, 449, **+457**, 461, 463, 467, 479, 487

Anmerkung:

Die in der Fortsetzung 3 von **Tab. 3** ermittelten primen Abstände ($\pm\Delta_i$) bestätigen dass hier gefundene **Gesetz 2** in vollem Umfang. Die Ergebnisse der mit dem MuPAD - Faktorisierungsprogramm durchgeführten Untersuchungen der Zahlen: $2 \cdot 3 \cdot 5 \cdot .. \cdot 157 \pm p > 157$ und $2 \cdot 3 \cdot 5 \cdot .. \cdot 163 \pm p > 163$ führten zu den o.g. Primzahlen, die in der direkten Umgebung dieser (lückenlosen) Fakultäten auftreten.

4. Die verborgenen Gesetze, die bei der Zerlegung nicht primer Zahlen in ihre Primzahl-Faktoren gelten:

Die in **Tab. 3** erkannten (bisher verborgenen) Gesetze, die bei der Zerlegung nicht primer Zahlen in ihre Primzahl-Faktoren auftreten, werden in diesem Kapitel vertiefend analysiert und vollauf bestätigt. Zunächst wird das **Gesetz 1** für die Zahl $n_g = \Pi p_i(p_i = 2 \text{ bis } p_n = 47) = 614889782588491410$ in **Tab. 4** an mehren Beispielen bestätigt. Der Leser kann die in **Tab. 4** vorgestellten Faktorisierungen unter Einsatz des MUPAD - Programms (selbst) nachvollziehen und ergänzend bestätigen.

Tab. 4 - Die Primzahl-Faktoren der Zahlen $n_g = 614889782588491410 \pm (p \leq p_n = 47)$

$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 = 614889782588491410$ Beispiel :=

$n_g = 614889782588491410 + 11 = 11 \cdot 57253657 \cdot 976340623$ Beispiel 1

$n_g = 614889782588491410 - 11 = 11 \cdot 73 \cdot 187897 \cdot 4075321589$ Beispiel 2

$n_g = 614889782588491410 + 31 = 31 \cdot 19835154277048111$ Beispiel 3

$n_g = 614889782588491410 - 31 = 31 \cdot 19835154277048109$ Beispiel 4

$n_g = 614889782588491410 - 41 \cdot 43 \cdot 47 = 41 \cdot 43 \cdot 47 \cdot 229 \cdot 541 \cdot 1549 \cdot 38669$ Beispiel 5

$n_g = 614889782588491410 + 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 = 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 \cdot 331 \cdot 571 \cdot 34231$ Beispiel 6

$n_g = 2 \cdot 3 \cdot 5 \cdot 7^2 \cdot 11 \cdot 13^2 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 = 5595497021555 2718310$ Beispiel 7

$n_g = 55954970215552718310 + 1 = 251 \cdot 401 \cdot 555930593988661$ Beispiel 8

$n_g = 55954970215552718310 - 1 = 437837 \cdot 127798633316857$ Beispiel 9

Tab. 4 - Fortsetzung

$n_g = 5595497021555271831 0 + 7 = 7 \cdot 193 \cdot 141157 \cdot 258421 \cdot 1135411$	Beispiel 10
$n_g = 55954970215552718310 - 7 = 7 \cdot 11087 \cdot 207763 \cdot 3470230909$	Beispiel 11
$n_g = 55954970215552718310 + 7^2 = 7^2 \cdot 25981 \cdot 56518 \cdot 777664469$	Beispiel 12
$n_g = 55954970215552718310 - 7^2 = 7^2 \cdot 2711 \cdot 94559 \cdot 4454615261$	Beispiel 13
$n_g = 55954970215552718310 + 13 = 13 \cdot 274093 \cdot 698359 \cdot 22486333$	Beispiel 14
$:n_g = 55954970215552718310 - 13 = 13 \cdot 157 \cdot 2798233 \cdot 9797421449$	Beispiel 15
$n_g = 55954970215552718310 + 13^2 = 13^2 \cdot 331094498316879991$	Beispiel 16
$n_g = 55954970215552718310 - 13^2 = 13^2 \cdot 331094498316879989$	Beispiel 17
$n_g = 55954970215552718310 + 7 \cdot 13 \cdot 17 = 7 \cdot 13 \cdot 17 \cdot 89 \cdot 881 \cdot 461298922459$	Beispiel 16
$n_g = 55954970215552718310 - 7 \cdot 13 \cdot 17 = 7 \cdot 13 \cdot 17 \cdot 1279 \cdot 4969 \cdot 5691265079$	Beispiel 19

Anmerkungen:

1. Die in **Tab. 4** vorgestellten Beispiele erfüllen (ausnahmslos) das Gesetz 1 - dies sowohl für die Addition als auch die Subtraktion einzelner Primzahlen ($p \leq p_n = 47$) als auch für die Addition bzw. Subtraktion beliebiger Produkte von Primzahlen ($p \leq p_n = 47$).

2. Das Faktorisierungs - Gesetz 1 hat zur Folge, dass die Primzahl-Dichten im Umfeld der Zahl $n_g = \Pi p_i =$ 614889782588491410 eine fallende Tendenz zeigen - siehe **Tab. 3**.

In **Tab. 5** wird demonstriert, dass das Gesetz 2 für alle Primzahlen gilt, die in den Fakultäten Πp_i nicht auftreten; diese Primzahlen können bei der Addition bzw. Subtraktion zu den Fakultäten Πp_i nicht auftreten - sie "verschmelzen" vielmehr zu (wesentlich) größeren Primzahl-Faktoren oder zu einer einzigen Primzahl. In **Tab. 5** wurden die Primzahlen: p =2, 3, 5, 7, ..., 47 sukzessive in der Fakultät $n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 = 514889782588491410$ entfernt und danach addiert bzw. subtrahiert.
Die so gewonnenen Primzahl-Fakultäten bestätigen stets die Gültigkeit von Gesetz 2.

Tab. 5 - Die Bestätigung von Gesetz 2 bei Ausgrenzung einzelner Primzahlen: p = 2, 3, 5, 7, 11, 13, 17, ...

$n_u = 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 + 2 = 307444891294245707$	Beispiel 1
$n_u = 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 - 2 = 83 \cdot 21647 \cdot 17116335603$	Beispiel 2
$n_u = 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 - 4 = 307444891294245701$	Beispiel 3
$n_g = 2 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 + 3 = 103 \cdot 1303 \cdot 495973 \cdot 3079189$	Beispiel 4
$n_g = 2 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 - 3 = 204963260862830457$	Beispiel 5
$n_g = 2 \cdot 3 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 + 5 = 1433 \cdot 3248661 \cdot 26417399$	Beispiel 6
$n_g = 2 \cdot 3 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 - 1 = 122977956517698281$	Beispiel 7
$n_g = 2 \cdot 3 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 - 5 = 5827 \cdot 35279 \cdot 598226969$	Beispiel 8
$:n_g = 2 \cdot 3 \cdot 5 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 + 1 = 87841397512641631$	Beispiel 9
$n_g = 2 \cdot 3 \cdot 5 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 + 7 = 22204241 \cdot 3956063957$	Beispiel 10
$n_g = 2 \cdot 3 \cdot 5 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 - 1 = 87841397512641629$	Beispiel 11
$n_g = 2 \cdot 3 \cdot 5 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 - 7 = 321947 \cdot 272844280309$	Beispiel 12
$n_g = " \cdot 3 \cdot 5 \cdot 7 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 + 11 = 10459 \cdot 15073 \cdot 354580403$	Beispiel 13

Tab. 5 - Fortsetzung

$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 - 11 = 356137 \cdot 156959459827$ Beispiel 14

$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 + 1 = 47299214045268571$ Beispiel 15
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 + 13 = 47299214045268583$ Beispiel 16
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 - 13 = 47299214045268557$ Beispiel 17

$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 + 17 = 264643 \cdot 136674641729$ Beispiel 18
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 - 17 = 131 \cdot 18773 \cdot 118843 \cdot 123757$ Beispiel 19

Tab. 5 Fortsetzung (Ausgrenzung der Primzahlen: 19, 23, 31. 37 41, 43 bzw. 47 in der Fakultät $n_g = \Pi p_i$)

$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 + 19 = 1723231 \cdot 18780198439$ Beispiel 20
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 - 19 = 6917 \cdot 488701 \cdot 9573763$ Beispiel 21

$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 + 23 = 10081963 \cdot 2651699711$ Beispiel 22
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47 - 23 = 19157 \cdot 1395538882571$ Beispiel 23

$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 28 \cdot 37 \cdot 41 \cdot 43 \cdot 47 + 1 = 19835154277048111$ Beispiel 24
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 37 \cdot 41 \cdot 43 \cdot 47 + 31 = 19835154277048141$ Beispiel 25
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 37 \cdot 41 \cdot 43 \cdot 47 - 31 = 19835154277048079$ Beispiel 26
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 37 \cdot 41 \cdot 43 \cdot 47 - 1 = 19835154277048109$ Beispiel 27
:
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 41 \cdot 43 \cdot 47 + 37 = 2663 \cdot 3209 \cdot 1944709201$ Beispiel 28
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 41 \cdot 43 \cdot 47 - 37 = 157 \cdot 5369017 \cdot 18715197$ Beispiel 29

$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 43 \cdot 47 + 41 = 347 \cdot 353 \cdot 122436030151$ Beispiel 30
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 43 \cdot 47 - 41 = 14997311770450969$ Beispiel 31

$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 47 + 43 = 14299762385778913$ Beispiel 32
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 47 - 43 = 353 \cdot 40509241886059$ Beispiel 33

$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 + 47 = 13082761331670077$ Beispiel 34
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 - 47 = 181 \cdot 72280449346243$ Beispiel 35

Tab, 5 - Fortsetzung (Ausgrenzung der Primzahlen p:= 11, 19 und 31 in der Fakultät $n_g = \Pi p_i(p = 2 \text{ bis } 47)$

$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 13 \cdot 17 \cdot 23 \cdot 29 \cdot 37 \cdot 41 \cdot 43 \cdot 47 + 11 = 11 \cdot 181 \cdot 47667023801$ Beispiel 36
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 13 \cdot 17 \cdot 23 \cdot 29 \cdot 37 \cdot 41 \cdot 43 \cdot 47 - 11 = 127 \cdot 747283814077$ Beispiel 37

$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 13 \cdot 17 \cdot 23 \cdot 29 \cdot 37 \cdot 41 \cdot 43 \cdot 47 + 19 = 1531 \cdot 61988925139$ Beispiel 38
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 13 \cdot 17 \cdot 23 \cdot 29 \cdot 37 \cdot 41 \cdot 43 \cdot 47 - 19 = 59 \cdot 118873 \cdot 13531753$ Beispiel 39

$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 13 \cdot 17 \cdot 23 \cdot 29 \cdot 37 \cdot 41 \cdot 43 \cdot 47 + 31 = 16649 \cdot 5700345029$ Beispiel 40
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 13 \cdot 17 \cdot 23 \cdot 29 \cdot 37 \cdot 41 \cdot 43 \cdot 47 - 31 = 1439 \cdot 65952080881$ Beispiel 41

$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 13 \cdot 17 \cdot 23 \cdot 29 \cdot 37 \cdot 41 \cdot 43 \cdot 47 + 11 \cdot 19 = 2011 \cdot 47192960909$ Beispiel 42
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 13 \cdot 17 \cdot 23 \cdot 29 \cdot 37 \cdot 41 \cdot 43 \cdot 47 - 11 \cdot 19 = 9483571 \cdot 10007311$ Beispiel 43

$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 13 \cdot 17 \cdot 23 \cdot 29 \cdot 37 \cdot 41 \cdot 43 \cdot 47 + 11 \cdot 31 = 4639 \cdot 125711 \cdot 162739$ Beispiel 44
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 13 \cdot 17 \cdot 23 \cdot 29 \cdot 37 \cdot 41 \cdot 43 \cdot 47 - 11 \cdot 31 = 94905044387449$ Beispiel 45

$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 13 \cdot 17 \cdot 23 \cdot 29 \cdot 37 \cdot 41 \cdot 43 \cdot 47 + 19 \cdot 31 = 10223 \cdot 53597 \cdot 173209$ Beispiel 46

Tab. 5 - Fortsetzung bei (gleichzeitiger) Ausgrenzung der Primzahlen: p = 11, 19 und 31:

$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 13 \cdot 17 \cdot 23 \cdot 29 \cdot 37 \cdot 41 \cdot 43 \cdot 47 - 19 \cdot 31 = 223 \cdot 425583158687$ Beispiel 47

$n_g = 2\cdot3\cdot5\cdot7\cdot13\cdot17\cdot23\cdot29\cdot37\cdot41\cdot43\cdot47-11\cdot19\cdot31 = 2111\cdot44957387201$ Beispiel 49

Tab. 5 - Fortsetzung mit addierten und subtrahierten Primzahlen $p > 47$ bzw. den Zahlen: +1 und -1:

$n_g = 2\cdot3\cdot5\cdot7\cdot13\cdot17\cdot23\cdot29\cdot37\cdot41\cdot43\cdot47+53 = 349\cdot244381\cdot1112747$ Beispiel 50
$n_g = 2\cdot3\cdot5\cdot7\cdot13\cdot17\cdot23\cdot29\cdot37\cdot41\cdot43\cdot47+61 = 94905044387851$ Beispiel 51
$n_g = 2\cdot3\cdot5\cdot7\cdot13\cdot17\cdot23\cdot29\cdot37\cdot41\cdot43\cdot47+73 = 94905044387863$ Beispiel 52
$n_g = 2\cdot3\cdot5\cdot7\cdot13\cdot17\cdot23\cdot29\cdot37\cdot41\cdot43\cdot47+103 = 94905044387893$ Beispiel 53
$n_g = 2\cdot3\cdot5\cdot7\cdot13\cdot17\cdot23\cdot29\cdot37\cdot41\cdot43\cdot47-71 = 94905044387719$ Beispiel 54
$n_g = 2\cdot3\cdot5\cdot7\cdot13\cdot17\cdot23\cdot29\cdot37\cdot41\cdot43\cdot47-79 = 94905044387711$ Beispiel 55
$n_g = 2\cdot3\cdot5\cdot7\cdot13\cdot17\cdot23\cdot29\cdot37\cdot41\cdot43\cdot47-89 = 94905044387701$ Beispiel 56
$n_g = 2\cdot3\cdot5\cdot7\cdot13\cdot17\cdot23\cdot29\cdot37\cdot41\cdot43\cdot47+1 = 11\cdot181\cdot47667023801$ Beispiel 57
$n_g = 2\cdot3\cdot5\cdot7\cdot13\cdot17\cdot23\cdot29\cdot37\cdot41\cdot43\cdot47-1 = 337\cdot281617342397$ Beispiel 58

Tab. 5 - Fortsetzung - Die Generierung von Primzahl-Zwillingen mod 2 nach den Faktorosierungs - Gesetzen

Beispiel:=

$n_g = 2\cdot3^3\cdot5\cdot7\cdot11\cdot13^2\cdot17\cdot19\cdot23\cdot29\cdot31\cdot37^2\cdot41\cdot43\cdot47-1 = 2661857868825579313889$ 1
$n_g = 2\cdot3^3\cdot5\cdot7\cdot11\cdot13^2\cdot17\cdot19\cdot23\cdot29\cdot31\cdot37^2\cdot41\cdot43\cdot47+1 = 2661857868825579313891$ 1

$n_g = 2^2\cdot3\cdot5\cdot7\cdot11\cdot13^2\cdot17\cdot19^2\cdot23\cdot29^2\cdot31\cdot37\cdot41\cdot43\cdot47^2-1 = 414018818192048213164019$ 2
$n_g = 2^2\cdot3\cdot5\cdot7\cdot11\cdot13^2\cdot17\cdot19^2\cdot23\cdot29^2\cdot31\cdot37\cdot41\cdot43\cdot47^2+1 = 414018818192048213164021$ 2

$n_g = 2\cdot3\cdot5\cdot7\cdot11\cdot13\cdot17\cdot19^2\cdot23\cdot29^2\cdot31\cdot37\cdot41\cdot43^2\cdot47^2-1 = 6847234300868489679251 09$ 3
$n_g = 2\cdot3\cdot5\cdot7\cdot11\cdot13\cdot17\cdot19^2\cdot23\cdot29^2\cdot31\cdot37\cdot41\cdot43^2\cdot47^2+1 = 684723430086848967925111$ 3

$n_g = 2\cdot7\cdot23!-1 = 221928234369323159999$ 4
$n_g = 2\cdot7\cdot23!+1 = 221928234369323160001$ 4

Anmerkungen und Schlussfolgerungen::

1. Die Gesetze: Gesetz 1 und Gesetz 2 werden in allen in **Tab. 5** angeführten Rechenbeispielen stets erfüllt. Danach gelten diese Gesetze (eineindeutig) in allen möglichen Primzahl - Faktorisierungen für alle nicht primen Zahlen.

2. Aus Gesetz 2 folgt, dass in den Umgebungen nicht primer Zahlen $n_g = \Pi p_i (p_1 = 2,$ lückenlos bis $p_n)$ Zahlen-Intervalle mit geringen Primzahl-Dichten auftreten müssen. Diese Primzahl-Dichten müssen mit steigenden "p_n - Zahlen" immer geringer werden.

3. Aus den vorgestellten Untersuchungen konnten die (tieferen) Ursachen für die ("wellenartigen") Schwankungen der Primzahl-Dichten in immer größeren Zahlenbereichen ($n \to \infty$) erkannt werden, wonach (jeweils) links und rechts auf die Intervalle mit geringen Primzahl-Dichten Intervalle mit wesentlich höheren Primzahl-Dichten auftreten (müssen).

4. In **Anlage 2** wurden diese (wellenartigen) Primzahl-Dichten für die Umgebungen der Fakultäten Πp_i (p = 2 bis p_n := 17, 19, ..., 251) in den Intervallen := -1000 bis +1000 mit der MuPAD - Methode bestimmt und kommentiert. Aus diesen Untersuchungen resultieren die Ursachen für die wellenartigen Schwankungen der Primzahl-Dichten, die sich in wachsenden Abständen (modulartig) wiederholen, sofern die höchsten Primzahlen (p_n) in den Fakultäten Πp_i unverändert bleiben.
Sofern diese (oberen) Primzahlen (p_n) zunehmend in den Fakultäten Πp_i erhöht werden, gelangt man in Zahlenbereiche mit (zwangsweise) geringeren wellenartigen Primzahl-Dichten.
Diese Gesetzmäßigkeiten konnten von den früheren großen Mathematikern nicht erschlossen werden, da diesen keine PC-Rechner und geeignete Testverfahren zur Verfügung gestanden hatten.

Die Ableitung der Gesetzmäßigkeit für Zahlen-Intervalle mit geringen Primzahl-Dichten:

Die Ableitung der Gesetzmäßigkeit für Zahlen-Intervalle mit geringen Primzahl-Dichten:

Den Primzahl-Forschern ist seit langem bekannt, dass in großen Zahlenbereichen ($n \to \infty$) immer wieder Zonen mit geringen Primzahl-Dichten auftreten, auf die Zonen mit höheren Primzahl-Dichten folgen. Die tieferen Gründe für diese Schwankungen der Primzahl-Dichten konnten bisher jedoch nicht entdeckt werden.

Ausgehend von der in **Tab. 3** abgeleiteten Primzahl-Dichte in der Umgebung der Zahl $n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 19 \cdot 23 = 223.092.870$, wo im Abstand von $\Sigma\,\Delta$=132 sechs Primzahlen situiert sind (was einem mittleren Abstand von $\Sigma\,\Delta/5$ = 26,4 entspricht) wurden in **Tab. 6** die Primzahl-Dichten analysiert, die in größeren Abständen von n_g auftreten. Diese Zeile wurde in **Tab. 3** unterstrichen hervorgehoben.
In **Tab. 6** wurden die Abstände: $\Delta = \pm 29 \cdot 31 = 899$ und $\Delta_2 = \pm 29 \cdot 31 \cdot 37 = 33263$ gewählt. Die in diesen Fakultäten auftretenden Primzahlen: p_j:= 29, 31, 37 sind größer als die oberste Primzahl (p_n = 23), die in der Fakultät $n_g = \Pi p_i$= 223.092.870 vorkommt. In **Tab. 6** wurden die sich ergebenden Primzahl-Dichten: $\Sigma\,\Delta$ ermittelt, die in den Umgebungen der Zahlen: n_g = 223.092.870 ±899 und n_g = 223.092.870 ±33263 auftreten.

Tab. 6 - Die Bestimmung der Primzahl-Dichten in den Umgebungen der Zahlen: n_g ±899 = $29 \cdot 31$ und n_g = ±33263: = $29 \cdot 31 \cdot 37$ sowie n_g = 1363783 = $29 \cdot 31 \cdot 37 \cdot 41$. Die Fortsetzung der Primzahlen, die in der Umgebung von n_g = 223092870 weiter links und rechts von den Primzahlen gem. **Tab. 3** auftreten - hier werden jeweils zwei Abschnitte mit jeweils sechs Primzahlen vorgestellt:

	Δ_1:=	Δ_2:=	Δ_3:=	Δ_4:=	Δ_5:=	Δ_6:=	$\Sigma\,\Delta$:=
n_g = 223092870 - 899 = = 223091971	-1009	-997	-983	-953	-911	-907	102
n_g = 223092870 + 899 = = 223093769	<u>+899</u> (29·31)	+929	+937	+941	+1031	+1039	140
n_g = 22309870 - 33263 = = 223059607	-33317	-33311	-33277 (107·311)	-33263 (29·31·37)	-33257 (7·4751)	-33239 (43·773)	78
n_g = 22309870 + 33263 = = 223126133	+33263 (29·31·37)	+33317	+33323	+33343	+33349	+33377	114
n_g = 223092870 - 1363783 = = 221729087	-1363883	-1363867	-1363853 (127·10739)	(29·131·353) -1363841	-1363799	-363783 (29·31·37·41)	100
n_g = 223092870 + 1363783 = = 224456653	+1363783 (29·31·37·41)	+1363787	+1363837	+1363841 (29·131·359)	+1363847	+1363861 (541·2521)	78
n_g = 223092870 - 79 = = 223092791	<u>Intervall 2</u> -181	-137	-127	-113	-101	-97	84
n_g = 223092870 + 53 = = 223092923	<u>Intervall 4</u> +67	+79	+101	+109	+131	+193	126
n_g = 223092870 - 181 = = 223092689	<u>Intervall 3</u> -293	-283	-269	-239	-199 --------	-197	96
n_g = 223092870 + 193 = = 223093063	<u>Intervall 5</u> +199	+223	+229	+241	+251	+263	64

<u>Anmerkungen und Schlussfolgerungen:</u>

1. In **Tab. 3** und in **Tab. 6** wurden insgesamt 30 Primzahlen bestimmt, die in der direkten Umgebung der geraden Zahl $n_g = \Pi p_i$ = 223092870 auftreten - jeweils 15, die links von n_g situiert sind und 15, die rechts von n_g situiert sind. Alle diese (30) Primzahlen treten (ausschließlich) in Primzahl-Abständen

sich die folgenden (variierenden) Primzahl-Dichten:

Die Primzahl-Dichte im Intervall 1:	$\Sigma\Delta = 132$	- Abstands-Dichte:: $\Sigma\Delta/5 = 132 / 5 = 26{,}4$
Die Primzahl-Dichte im Intervall 2:	$\Sigma\Delta = 84$	- Abstands-Dichte: $\Sigma\Delta/5 = 84 / 5 = 16{,}8$
Die Primzahl-Dichte im Intervall 3:	$\Sigma\Delta = 96$	- Abstands-Dichte: $\Sigma\Delta/5 = 96 / 5 = 19{,}2$
Die Primzahl-Dichte im Intervall 4:	$\Sigma\Delta = 126$	- Abstands-Dichte: $\Sigma\Delta/5 = 126 / 5 = 25{,}2$
Die Primzahl-Dichte im Intervall 5:	$\Sigma\Delta = 64$	- Abstands-Dichte: $\Sigma\Delta/5 = 64 / 5 = 12{,}8$

Aus diesen Analysen folgt, dass die Primzahl-Dichte in der direkten (nahen) Umgebung von n_g am geringsten ist und in den weiter links situierten Intervallen schnell ansteigt und in dem weiter rechts situierten Intervall (Intervalle 4) zunächst gering bleibt, im darauf folgenden Intervall (Intervall 5) aber signifikant größer wird. Am geringsten ist die Primzahl-Dichte im direkten Umfeld von n_g wo die aufeinander folgenden Primzahlen im Abstand .$\Delta_3 + \Delta_4 = 80$ auftreten.

3. Aus **Tab. 6** geht hervor, dass ab den Abständen: $\Delta := 29 \cdot 31,\ 29 \cdot 31 \cdot 37,\ 29 \cdot 31 \cdot 37 \cdot 41$ auch nicht prime Abstände der Primzahlen (bezogen auf $n_g = 22309870$) auftreten können, wobei diese Zahlen (Δ) jeweils aus Primzahl-Fakultäten stammen, in denen nur größere Primzahlen auftreten als $p_n = 23$.

4. Aus **Tab. 6** ist zu entnehmen, dass in den hier analysierten Intervallen nur der Zwillinge mod 2: (223092671, 223092673) vorkommt, der in den Abständen: $\Delta = -199$ und $\Delta = -197$ von n_g im Intervall 3 auftritt. Dieser Zwilling mod 2 wurde in **Tab. 6** strichliniert markiert.
Sofern das Intervall 5 um drei weitere Intervalle a′ sechs Primzahlen erweitert wird, erscheint im Intervall 8 der Zwilling mod 2: (223093511, 22393513), der von der geraden Zahl ($n_g = 223092870$) in den Primzahl-Abständen: $\Delta = +641$ und $\Delta = +643$ entfernt ist.

5. Durch eine ähnliche Analyse der Primzahl-Dichten für die Zahlen $n_g = \Pi p_i$, $p_n := 29, 31, 37, 41, 43, 47, 53, 59, 61, \ldots$ kann der Nachweis erbracht werden, dass die Primzahl-Dichten in den Umgebungen dieser geraden Zahlen ein gleiches Verhalten aufweisen, wie die Primzahl-Dichten, die im Umfeld der geraden Zahl n_g, die in **Tab. 6** analysiert wurden. Dabei ist beachtlich, dass die Primzahl-Dichten im Intervall 1 (wie aus **Tab. 3** entnommen werden kann) mit steigendem Wert der höchsten Primzahl (p_n) in der Fakultät $n_g = \Pi p_i$ fallende Werte aufweisen müssen.

6. Durch die in **Tab. 3** und **Tab. 6** vorgenommenen Analysen, konnten (wie ersichtlich) die tieferen Gründe gefunden werden, die den Schwankungen der Primzahl-Dichten in steigenden Zahlenbereichen ($n \to \infty$) zugrunde liegen. Die aus **Tab. 6** gewonnenen Erkenntnisse finden ihre volle Bestätigung in **Anlage 2** - siehe auch die dortigen Kommentare,.
Diese Thematik soll nun durch die Analyse der Primzahl-Dichten im Umfeld der Fakultäten $n_g = n!$ erweitert werden.

Die Darstellung der Fakultäten $n_g = n!$ durch die Fakultäten $n_g = 2^{n1} \cdot 3^{n2} \cdot 5^{n3} \cdot 7^{n4} \cdot \ldots\ p^m$:

Im mathematischen Schrifttum wurde das Auftreten primzahlfreier Intervalle in der Umgebung der geraden Zahlen $n_g = n!$ bisher nur vereinfachend untersucht, verbunden mit der Schlussfolgerung, dass in der Zahlenfolge

$n! + 2,\ n! + 3,\ n! + 4,\ \ldots,\ n! + n+1$ Anmerkung: $n! + 1$ kann fallweise eine Primzahl sein..

keine Primzahlen auftreten können, und dass daher auf beliebig große Fakultäten n! bei steigenden Werten von n ($n \to \infty$) beliebig große primzahlfreie Zahlen-Intervalle auftreten müssen.

In die bisher angestellten Betrachtungen wurden jedoch die primzahlfreien Intervalle nicht einbezogen, die "im Vorfeld" der Fakultäten n! auftreten. Die Ergänzung der hier erforderlichen Analysen soll anhand von **Tab. 5** erfolgen. In **Tab. 7** wurden die Fakultäten n! durch Primzahl-Fakultäten dargestellt. In **Tab. 5** wurden (ausgenommen die Fakultät 4!) nur die Fakultäten der Primzahlen bis p=23 angeführt und die in der Umgebung dieser Fakultäten auftretenden sechs direkt benachbarten Primzahlen und deren Summen-Abstand $\Sigma\Delta$ bestimmt. Die primzahlfreien Zonen in den Umgebungen der Fakultäten n! sind größer als bisher vermutet wurde und zeigen ein stetiges Wachstun bei ansteigenden Fakultäten: $n_g = n!$.

Tab. 7 Die Bestimmung der Primzahlen im Umfeld der Fakultäten n!:

Die primzahlfreien Zonen in den Umgebungen der Fakultäten n! sind größer als bisher vermutet wurde un(zeigen ein stetiges Wachstun bei ansteigenden Fakultäten: $n_g = n!$.

Tab. 7 Die Bestimmung der Primzahlen im Umfeld der Fakultäten n!:

n! :=	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$:=
$2! = 1 \cdot 2 = 2$				**+1**	+3	+5	4
$3! = 2 \cdot 3 = 6$		-3	**-1**	**+1**	+5	+7	10
$4! = 2^3 \cdot 3 = 24$	-7	-5	**-1**	+5	+7	+13	20
$5! = 2^3 \cdot 3 \cdot 5 = 120$	-13	-11	-7	+7	+11	+17	30
$7! = 2^4 \cdot 3^2 \cdot 5 \cdot 7 = 5040$	-19	-17	**-1**	+11	+19	+37	56
$11! = 2^8 \cdot 3^4 \cdot 5^2 \cdot 7 \cdot 11 = 39916800$	-43	-17	-13	**+1**	+17	+19	62
$13! = 2^{10} \cdot 3^5 \cdot 5^2 \cdot 7 \cdot 11 \cdot 13 = 6227020800$	-67	-61	-23	+67	+73	+109	176
$17! = 2^{15} \cdot 3^6 \cdot 5^3 \cdot 7^2 \cdot 11 \cdot 13 \cdot 17 = 355687428096000$	-239	-61	-59	+31	+53	+59	298
$19! = 2^{16} \cdot 3^8 \cdot 5^3 \cdot 7^2 \cdot 11 \cdot 13 \cdot 17 \cdot 19 = 121645100408832000$	-199 ---- -197	-113	-101	+89	+109	+113	310
$23! = 2^{19} \cdot 3^{10} \cdot 5^4 \cdot 7^3 \cdot 11^2 \cdot 13 \cdot 17 \cdot 19 \cdot 23 = 25852016740665940000$	-197	-191	-89	+97	+109	+149	346

Anmerkungen und Schlussfolgerungen:

1. Die in **Tab. 7** auftretenden Zwillinge mod 2 wurden durch strichlinierte Linien markiert und die Abstände $\Delta = \pm 1$ wurden fett gedruckt hervorgehoben.

2. In **Tab. 7** findet das (bereits in **Tab. 3** gefundene) Gesetz die Bestätigung, dass im Umfeld der Zahlen $n_g = \Pi p_i$ alle Primzahlen, die links und rechts von der geraden Zahl (n_g) auftreten, nur in Primzahl-Abständen: $\Delta = \pm p_j > p_n$ auftreten können. Des weiteren treten jedoch in der Folge: Δ_i nicht alle Primzahlen $p_j > p_n$ auf. Dies ist die tiefere Ursache für die Abnahme der Primzahl-Dichten.

3. Ein Vergleich der Primzahl-Dichten, die in **Tab. 7** und in **Tab. 3** auftreten, zeigt, dass im Umfeld der Fakultäten n! wesentlich kleinere Primzahl-Dichten auftreten als im Umfeld der Fakultäten $n_g = \Pi p_i$. wiewohl in beiden Fakultäten die gleichen Primzahlen auftreten. Hierzu die folgende Hilfstabelle 1, aus der ersichtlich ist, in welchem "Maße" die Zahlen $n_g = \Pi p_i$ in den Fakultäten n! in höhere Zahlen.-Intervalle verschoben werden:

Hilfstabelle: - Die Verschiebung der Zahlen $n_g = \Pi p_i$ in den Fakultäten n!

$n_g = \Pi p_i$:=	$p_n!$:=	Faktor $p_n! / \Pi p_i = k$
$n_g = 2 \cdot 3 \cdot 5 = 30$	5! = 120	k = 4
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 = 210$	7! = 5040	k = 24
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 = 2310$	11! = 39916800	k = 17280
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 13 = 30030$	13! = 6227020800	k = 207360
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 17 = 510510$	17! = 355687428096000	k = 6967295998
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 19 = 9699690$	19! = 121645100408832000	k > 12541249160
$n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 23 = 223092870$	23! = 15852016740665940000	k > 115880067100

4. In der folgenden Hilfstabelle 2 werden die Primzahl-Dichten im direkten Umfeld der geraden Zahlen: $n_g = \Pi p_i$ und $n_g = p_n!$ miteinander verglichen - als Maßstab dient dabei der Abstand $\Sigma\Delta$ innerhalb dessen sechs Primzahlen situiert sind.

Hilfstabelle 2: - Der Vergleich der Primzahl-Dichten im Umfeld von $n_g = \Pi p_i$ und $n_g = p_n!$

$n_g = \Pi p_i$:=	$\Sigma\Delta$:=	$n_g = p_n!$:=	$\Sigma\Delta$:=

$n_g = \Pi p_i$	$\Sigma\Delta$:=	$n_g = p_n!$	$\Sigma\Delta$:=
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11$	46	$n_g = 11!$	62
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13$	60	$n_g = 13!$	176
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17$	96	$n_g = 17!$	298
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19$	82	$n_g = 19!$	310
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23$	132	$n_g = 23!$	402

Anmerkung:

Aus den Quotienten: $\Sigma\Delta/5$ resultieren die mittleren Abstände in denen jeweils die sechs ermittelten Primzahlen in den Umgebungen der Zahlen: $n_g = \Pi p_i$ und $n_g = p_n!$ auftreten. Aus **Tab. 3** und **Tab. 5** ist zu entnehmen, dass diese sechs Primzahlen (paarweise) unterschiedliche Abstände aufweisen.

5 Der Leser, der die hier entdeckten Gesetze der variierenden Primzahl-Dichten in steigenden Zahlen-Intervallen nachprüfen möchte, kann (ausgehend von **Tab. 3**) weitere Tabellen generieren, in denen die geraden Primzahl-Fakultäten: $n_g = \Pi p_i$ mit beliebigen Faktoren $k \geq 2$ multipliziert wurden. Dieser mathematisch interessierte Leser wird sich danach davon überzeugen, dass die bislang nicht verstandenen Schwankungen der Primzahl-Dichten in immer größer werdenden Zahlen-Intervallen ($n \to \infty$) konkret nachvollziehbare Ursachen haben und sich in "länger werdenden" Wellen fortsetzten..

6. Von großer Wichtigkeit für das Verständnis der Schwankungen der Primzahl-Dichten ist noch das Verständnis des Grenzbereiches bis zu dem die Abstände der Primzahlen $p_j = \Pi p_i \pm\Delta_{prim}$ gültig sind. Aus den Analysen, die in **Tab. 4** vorgestellt wurden, geht hervor, dass die Grenze dieses Gesetzes bis zur Grenz-Zahl $n_u = p_{n-1} \cdot p_n$ reicht. "Große Fakultäten Πp_i" weisen deshalb lange Zonen mit geringen Primzahl-Dichten auf, wobei der in der Hilfstabelle 2 angeführte Faktor "k" eine Rolle spielt.

7. In **Tab. 8** wurden Primzahl-Fakultäten "mit ausgeblendeten Primzahlen" bzw. mit Primzahlen die in gewissen Potenzen auftreten, für die Fakultät $n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23$ näher analysiert. Dabei zeigt sich ein abweichendes Verhalten der Primzahl-Dichten in den Umgebungen der Zahlen (n_g). Insbesondere können hier "nicht prime Abstände" (Δ) zwischen den aufeinander folgenden Primzahlen auftreten, die relativ klein sein können.

Tab. 8 - Die Primzahl-Dichten in den Umgebungen der Zahlen $n_g = \Pi p_i$, $p_1 = 2$ bis $p_n = 23$, in denen einzelne Primzahlen nicht auftreten bzw. in Potenzen auftreten:

n_g :=	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$:
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 =$ = 223092870	-79	-61	-43	+37	+47	+53	132
$n_g = 2 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 =$ = 74364290	-9	-3 -------	-1	+9	+41	+71	80
$n_g = 2 \cdot 3 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 =$ = 44618574	-43	-31	-5	+25	+47	+61	126
$n_g = 2 \cdot 3 \cdot 5 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 =$ = 31870410	-43------	-41	-7	+31	+49	+61	104
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 13 \cdot 17 \cdot 19 \cdot 23 =$ = 20281170	-29	-11	-1	+29	+41	+59	88
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 17 \cdot 19 \cdot 23 =$ = 17160990	-31-------	-29	-1-------	+1	+13	+31	62
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 19 \cdot 23 =$ = 13123110	-47	-31------	-29	+1	+37	+59	106
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 23 =$ = 11741730	-43-------	-41	-1	+29	+41-----	+43	86
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 =$ = 9699690	-41	-37	-23	+23	+37	+41	82
$n_g = 2^2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 =$ = 466185740	-97	-37	-31	+37	+61	+67	164
$n_g = 2 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 =$ = 699278610	-89	-67	-1	+49	+67	+83	172

Tab. 8 - Fortsetzung

= 466185740	-97	-37	-31	+37	+61	+67	164
$n_g = 2 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 =$ = 699278610	-89	-67	-1	*+49*	+67	+83	172
Tab. 8 - Fortsetzung							
$n_g = 2 \cdot 3 \cdot 5^2 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 =$ = 1165464350	-67	*-51*	-33	*+21*	+53	+89	156
$n_g = 2 \cdot 3 \cdot 5 \cdot 7^2 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 =$ = 1631650090	-47	*-33*	-23	+1	*+27*	+37	84
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11^2 \cdot 13 \cdot 17 \cdot 19 \cdot 23 =$ = 2564021570	-81	*-63*	*-57*	*+21*	+29	*+69*	150
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13^2 \cdot 17 \cdot 19 \cdot 23 =$ = 3030207310	*-57*	-47	-17	+7	*+21*	*+121*	178
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17^2 \cdot 19 \cdot 23 =$ = 3962578790	*-111*	*-21*	-19	*+21*	+53	*+57*	168
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19^2 \cdot 23 =$ = 4428764530	*-39*	-17	-11	+3	*+51*	*+99*	138
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23^2 =$ = 5361136010	*-91*	-67	-7	*+9*	*+33*	*+63*	154
$n_g = 2 \cdot 5 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 = 10623470$	*-27*	-9	<u>-7</u>	*+21*	+29	+41	68
$n_g = 2 \cdot 3 \cdot 5 \cdot 11 \cdot 17 \cdot 19 \cdot 23 = 2451570$	-31 -----	-29	<u>-13</u>	<u>+7</u>	*+31*	*+49*	80
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 13 \cdot 19 \cdot 23 = 1193010$	-59	-43	-41	+1	<u>+11</u>	+31	74
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 17 \cdot 23 = 903210$	-47	-31	<u>-13</u>	+1	<u>+13</u>	+41	88
$n_g = 2 \cdot 10623491 = 21246982$	*-51*	*-35* -----	*-33*	+1	<u>+19</u>	<u>+37</u>	88

<u>Anmerkungen und Schlussfolgerungen:</u>

1. In **Tab. 8** wurden alle Zwillinge mod 2 mit gestrichelten Linien verbunden und die nicht primen Abstände (Δ) wurden kursiv gedruckt.

2. Die in den Fakultäten $n_g = \Pi p_i$ nicht auftretenden Primzahlen treten (fallweise) in den Abständen (Δ) auf. Diese (primen) Abstände wurden in **Tab. 6** unterstrichen markiert.

3. Die in den Fakultäten $n_g = \Pi p_i$ nicht auftretenden Primzahlen, kommen (fallweise) quadratisch in den nicht primen Abständen vor - siehe z.B. die Abstände $\Delta := 9 = 3^2$, $25 = 5^2$ und $49 = 7^2$.

4. In den Fakultäten $n_g = \Pi p_i$, in denen einzelne Primzahlen doppelt multipliziert sind, treten zahlreiche nicht prime Abstände auf, die kursiv notiert wurden.

5. Die in **Tab. 8** ermittelten Primzahl-Zwillinge mod 2 wurden in der (folgenden) Hilfstabelle 3 unter der Angabe ihrer Klassen - Zugehörigkeit ausgewiesen in der Reihenfolge in der sie in **Tab. 6** auftreten:

<u>Hilfstabelle 3:</u>

Zwilling mod 2:	Element der Klassenfolge:=	n:=
(74.364.287, 74.364.289)	18 + n x30	n = 2.478.809
(31.870.367, 31.870.369)	18 + n x30	n = 1.062.345
(17.160.959, 17.160.961)	30 + n x30	n = 572.032
(17.160.989, 17.160.991)	30 + n x30	n = 572.033
(2.451.539, 2.451.541)	30 + n x30	n = 81.718
(21.246.947, 21.246.949)	18 + n x30	n = 708.231

In **Tab. 9** wurden (in Ergänzung zu **Tab. 8**) die Primzahl-Dichten in den Umgebungen der Fakultäten: 6!, 8!, 9!, 10!, 12!, 14!, 15!, 16!, 18! und 20! (durch PC-Teste) ermittelt. Dabei konnte festgestellt werden, dass im

Umfeld all dieser Fakultäten ausschließlich prime Abstände (Δ_{prim}) auftreten, und dass die Primzahl-Dichten stetig (bei steigenden Fakultäten) geringer werden.

Tab. 9 - Die Bestimmung der Primzahl-Dichten im Umfeld der Fakultäten: 6!, 8!, 9!, 10!, 12!, 14!, 15!, 16!, 18! und 20!:

n!:=	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$:
6! = 720	-19	-11	-1	+7	+13	+19	38
8! = 40.320	-43	-37	-1	+23	+31	+37	80
9! = 362.880	-29	-17	-13	+17	+23	+31	60
10! = 3.626.800	-23	-17	-11	+11	+19	+41	64
12! = 479.001.600	-17	-13	-1	+29	+43	+59	76
14! = 87.178.291.200	-43	-17	-1	+19	+41	+71	72
15! = 1.307.674.368.000	-71	-59	-47	+43	+149	+179	250
16! = 20.922.789.888.000	-89	-71	-53	+23	+41	+73	162
18! = 6.402.373.905.728.000	-47	-43 -----	-41	+37	+61	+73	120
20! = 243.290.200.817.664.000	-139	-61	-53	+41	+53	+181	320

<u>Anmerkungen und Schlussfolgerungen:</u>

1. Die in **Tab. 9** ermittelten Primzahl-Dichten im Umfeld der Fakultäten: 6!, 8!, 9!, 10!, 12!, 14!, 15!, 16! 18! und 20! ergänzen die in **Tab. 8** ermittelten Primzahl-Dichten in den Umgebungen der Fakultäten: 2!, 3!, 4!, 5!, 7!, 11!, 13!, 17!, 19! und 23!.

2. Die in **Tab. 9** ermittelten Primzahlen, die in den Umgebungen dieser Fakultäten auftreten, bestätigen die Gesetzmäßigkeit, dass alle Primzahlen, die in den Umfeldern der Fakultäten $n_g = n!$ auftreten, stets nur in Primzahl-Abständen vom Typ: $\pm\Delta_{prim}$ auftreten können. Des weiteren wird hier bestätigt, dass die mittleren Primzahl-Dichten in zunehmend größeren Zahlen-Intervallen: n! eine signifikante Tendenz zu fallenden Werten aufweisen, da die Summen-Abstände:: $\Sigma\Delta$, zwischen den 6 (hier lokalisierten) Primzahlen ein stetiges Wachstum aufweisen, m.a.W. der mittlere Abstand: $\Sigma\Delta/6$ zunehmend steigt.

3. In **Tab. 9** wurde der Zwilling mod 2: (6402373905727957, 6402373905727959) gefunden, der in der Klasse: 18 + n x30, n = 21341246357598 situiert ist Dieser Zwilling mod 2 wurde durch eine unterbrochene Linie markiert. Sofern man die Intervalle um die Zahlen n! vergrößert, kann man weitere Zwillinge mod 2 finden. Dies besagt, dass auch in den großen Zahlen-Intervallen mit den sehr geringen Primzahl-Dichten Zwillinge mod 2 situiert sein können. In derartigen Zwillingen mod 2 können die Fakultäten $n_g = n!$ auch direkt im Abstand $\Delta = \pm 1$ situiert sein. Diese Zwillinge mod 2 sind Elemente der Klasse: 30 + n x30.

5. Das Gesetz dem die Abnahme der Primzahl-Dichten in großen Zahlen-Intervallen unterliegt:

Aus den Analysen, die in **Tab. 3**, **Tab. 5**, **Tab. 8** und **Tab. 9** demonstriert wurden, resultiert das folgende Gesetz dem die Dichte-Schwankungen der Primzahlen in immer größeren Zahlen-Intervallen ($n \to \infty$) folgen müssen:

Das Gesetz der Dichte-Schwankungen der Primzahlen in wörtlicher Formulierung:

Die Primzahl-Dichten in den Umgebungen der Fakultäten $n_g = \Pi p_i$ ($p_i = 2$ lückenlos bis $p_i = p_n$) und der Fakultäten n! - definiert durch die mittleren Abstände: $\Sigma\Delta/6$ - unterliegen mit steigenden Primzahlen p_{n+1} (Nachfolger von p_n) zwangsweise einer definierten Reduktion, wobei in den Intervallen zwischen $n_g = \Pi p_i$ und $n_g = p_{n+1} \cdot \Pi p_i$ sowie $n_g = n!$ und $n_g = (n+1)!$ wieder höher Primzahl-Dichten folgen, die von den Bildungsgesetzen der hier auftretenden geraden Zahlen abhängig sind.
In den direkten Umgebungen von $n_g = \Pi p_i$ und $n_g = n!$ treten alle Primzahlen ($p_j < n_g$ und $p_j > n_g$) ausschließlich in Primzahl-Abständen (Δ_{prim}) auf.

Durch das hier nachgewiesene Gesetz der Dichte-Schwankungen der Primzahlen in anwachsenden Zahlen-Intervallen ($n \to \infty$) konnte nach Wissen des Verfassers erstmalig der (bisher verborgene) Grund für die Primzahl-Dichte-Schwankungen in steigenden Zahlen-Intervallen erkannt werden.

Anhand des folgenden Beispiels kann die primzahlfreie Zone in der Umgebung der Fakultät $n_g = \Pi p_i$, $p_1 = 2$ bis p_n = 25852016740665940103 (dies ist die zweit größte in **Tab. 7** im Umfeld der Fakultät 23! getestete Primzahl) annähernd bestimmt werden, wobei es auf die (detaillierte) Bestimmung dieser enorm großen Zahl $n_g = \Pi p_i$ nicht ankommt. Die auf p_n folgende Primzahl ist die Primzahl p_{n+1} = 25652016740665590163. Möglich sind hier die folgenden vier Varianten:

<u>Variante 1:</u> In der direkten Umgebung von n_g ist ein Zwilling mod 2 (n_g -1, n_g +1) situiert, der Element der Folge 30 + 30n ist.

<u>Variante 2:</u> In der direkten Umgebung von n_g ist die Primzahl n_g -1 situiert, der eine Primzahl $p \leq n_g$ - (p_{n+1} = 25852016740665940163) vorangeht, wobei auf die Primzahl n_g -1 eine Primzahl $p \geq n_g$ + (p_{n+1} = 25852016740665940163) folgt.

<u>Variante 3:</u> In der direkten Umgebung von n_g ist die Primzahl n_g +1 situiert, der eine Primzahl $p \leq n_g$ - (p_{n+1} = 25852016740665940163) vorangeht und die Primzahl $p \geq n_g$ + (p_{n+1} = 25852016740665940163) auf die Primzahl n_g +1 folgt.

<u>Variante 4:</u> Die ungeraden Zahlen: n_g -1 und n_g +1 sind nicht prim, dann ist vor der geraden Zahl $n_g = \Pi p_i$ eine Primzahl $p \leq n_g$ - (p_{n+1} = 25852016740665940163) situiert, auf die eine Primzahl $p \geq n_g$ + (p_{n+1} = 25852016740665940163) folgt.

<u>Das "Vorschau-Ergebnis" lautet:</u>

1. Im Abstand von $\Delta \geq 2$ x 25852016740665940163 = 51704033481331880326 können (fallweise) vier, drei bzw. zwei Primzahlen im Umfeld der geraden Zahl $n_g = \Pi p_i$ situiert sein. Überdies können alle Primzahlen, die weiter entfernt im Umfeld von $n_g = \Pi p_i$ vorkommen nur in Primzahl-Abständen $\pm\Delta_{prim}$ auftreten.

2. Bis auf die (fallweise) möglichen Primzahlen n_g -1 und n_g +1 können im direkten Umfeld von $n_g = \Pi p_i$ nur Primzahlen in den Abständen: $\pm\Delta_{prim} \geq p_{n+1}$ = 25852016740665940163 auftreten.

3. Weitere Informationen über die Schwankungen der Primzahl-Dichten in mittleren und großen Zahlen - Intervallen können aus **Anlage 3** entnommen werden.

<u>Die Schlussfolgerung aus dem vorgestellten Rechen-Beispiel lautet:</u>

Die Primzahl-Dichten, die in beliebig großen Zahlen-Intervallen um die Zahlen $n_g = \Pi p_i$, von $p_1 = 2$ bis $p_n \to \infty$, sowie um die Zahlen n!, $n \to \infty$ auftreten, können nach dem hier unter Beweis gestellten Gesetz der Primzahl-Dichten rechnerisch annähernd bestimmt werden, was bisher als unmöglich galt.

6. Der Beweis, dass es unendlich viele Zwillinge mod 4 gibt, die in sechs Klassen auftreten:

In der Primzahl-Forschung wurde bisher nur, die in dieser Monographie unter Beweis gestellte Vermutung, dass die Zwillinge mod 2 wahrscheinlich in einer unendlichen Zahl auftreten, aufgestellt, nicht aber die Vermutung, dass auch die "Semi-Zwillinge mod 4" in einer unendlichen Zahl auftreten.

Für diese (bislang nicht geäußerte) Vermutung, zu der man bei der Analyse der Verteilungen der Zwillinge mod 4 in der Folge der Primzahlen: p = 2, 3, 5, <u>7, 11</u> , <u>13, 17,</u> <u>19, 23</u> ... gelangt, soll nun der Beweis erbracht werden. Der Leser wird dabei auch auf die **Anlage 2** hingewiesen.

Der Ausgangspunkt zu diesem Beweis ist die in **Tab. 10** vorgestellte Aufteilung der natürlichen Zahlen in vier gleichmächtige (unendliche) Teilfolgen:

Tab. 10 - Die Aufteilung der Folge der natürlichen Zahlen (n) in vier gleichmächtige Teilfolgen:

			Anmerkungen:
Folge 1 :	1, **5**, 9, **<u>13, 17</u>**, 21, 25, **29**, 33, **<u>37, 41</u>**, ...	(1 +4n), n = 0, 1, 2, 3, ...	Folge 1 ist Primzahl-aktiv
Folge 2:	**2**, 6, 10, 14, 18, 22, 26, 30, 34, 38, ...	(2 +4n), n = 0, 1, 2, 3. ...	Folge 2 ist Primzahl-passiv
Folge 3:	**<u>3, 7, 11</u>**, 15, **<u>19, 23</u>**, 27, **31**, 35, 39, **43**, ...	(3 +4n), n = 0, 1, 2. 3, ...	Folge 3 ist Primzahl-aktiv
Folge 4:	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, ...	(4 +4n), n = 0, 1, 2, 3, ...	Folge 4 ist Primzahl-passiv

Anmerkungen:

1. Die Folge 1 ist eine Dirichlet-Progression, in der unendlich viele Primzahlen mit den Endziffern: EZ1, EZ3, EZ7, EZ9 auftreten (die einzige Primzahl p = 5 mit der Endziffer EZ5 kommt hier nur einmal vor. Alle in der Folge 1 auftretenden Primzahlen ergeben bei der Teilung durch die Zahl n = 4 den Rest 1.

2. Die Folge 3 ist eine Dirichlet-Progression, in der unendlich viele Primzahlen mit den Endziffern: EZ1, EZ3, EZ7, EZ9 auftreten. Alle Primzahlen der Folge 3 ergeben bei der Teilung durch die Zahl n = 4 den Rest 3.

3. Aus **Tab. 10** ist ersichtlich, dass Zwillinge mod 4 stets nur aus Primzahlen der Folge 1 oder aber aus der Folge 3 gebildet werden können. Diese Zwillinge mod 4 wurden in **Tab. 9** jeweils unterstrichen markiert, wobei alle Primzahlen fett gedruckt wurden, die nicht primen Zahlen aber normal gedruckt.

4. Aus **Tab. 10** ist ersichtlich, dass die in der Folge 3 auftretenden Zwillinge mod 4: **(3, 7)** und **(7, 11)** ein (über die Primzahl p = 7) verkettetes Zwillings-Paar mod 4 bilden. Weitere "verkettete Zwillinge mod 4" können (offensichtlich) in den Folgen 1 und Folge 3 nicht auftreten.

Aus **Tab. 10** resultiert die Schlussfolgerung, dass es acht Klassen von Primzahlen gibt, die gemäß **Tab. 11** in zwei Oberklassen zusammengefasst werden können:

Tab. 11 - Die Klassen-Teilung der Primzahlen, die in den Folgen 1 und 3 auftreten:

<u>Oberklasse I: - Alle Primzahlen der Folge 1, die bei Division durch 4 den Rest 1 ergeben</u>

Klasse 1:	Alle Primzahlen der Folge 1 mit EZ1:	**41, 61, 101, 141** ...	Primzahlen im Abstand mod 20n, n ≥1
Klasse 2:	Alle Primzahlen der Folge 1 mit EZ3:	**13, 53, 73, 113** ...	Primzahlen im Abstand mod 20n, n ≥1
Klasse 3:	Alle Primzahlen der Folge 1 mit EZ7:	**17, 37, 77, 97** ...	Primzahlen im Abstand mod 20n, n ≥1
Klasse 4:	Alle Primzahlen der Folge 1 mit EZ9:	**29, 89, 109 149** ...	Primzahlen im Abstand mod 20n, n ≥1

<u>Oberklasse II - Alle Primzahlen der Folge 3, die bei der Division durch 4 den Rest 3 ergeben:</u>

Klasse 5:	Alle Primzahlen der Folge 3 mit EZ1:	11, 31, 71, 131, ...	Primzahlen im Abstand mod 20n, n ≥1
Klasse 6:	Alle Primzahlen der Folge 3 mit EZ3:	3, 23, 43, 83, ...	Primzahlen im Abstand mod 20n, n ≥1
Klasse 7:	Alle Primzahlen der Folge 3 mit EZ7:	7, 47, 67, 107, ...	Primzahlen im Abstand mod 20n, n ≥1
Klasse 8:	Alle Primzahlen der Folge 3 mit EZ9:	19, 59, 79, 139, ...	Primzahlen im Abstand mod 20n, n ≥1

Anmerkung und Schlussfolgerungen:

1. Dirichlet hatte bekanntlich nur vier Primzahl-Klassen unterschieden, die Primzahlen mit den Endziffern: EZ1, EZ3, EZ7 und EZ9. Die Einführung von acht Primzahl-Klassen erweist sich jedoch als notwendig, um den Nachweis zu führen, dass Zwillinge mod 4 in einer unendlichen Zahl auftreten.

2. Die Primzahlen der Oberklasse I wurden fett gedruckt und die der Oberklasse II normal gedruckt. Diese Art der Unterscheidung der Primzahlen der Oberklasse I und II erweist sich als vorteilhaft und wird in den weiteren Untersuchungen beibehalten.

3. Aus den in **Tab. 9** eingeführten Folgen (der Folge 1 und der Folge 3) und den in **Tab. 10** erfolgten Klassen-Teilungen (in acht Dirichlet-Progressionen) gewinnt man die wichtige Einsicht (Erkenntnis), dass Zwillinge mod 4 stets nur aus Primzahlen jeweils einer Oberklasse bestehen können.

Der Nachweis, dass Zwillinge mod 4 in sechs Dirichlet-Progressionen auftreten können:

Aus den Erkenntnissen, die sich aus **Tab. 10** und **Tab. 11** ergeben, und aus dem (einsichtigen) Sachverhalt, dass Zwillinge mod 4 nur dann auftreten können, wenn die Differenzen der Endziffern der Primzahlen den Wert 4 aufweisen, gelangt man (leicht) zu der Feststellung, dass dies nur für die folgenden zyklischen Kombinationen der Endziffern erfüllbar ist:

EZ3 & EZ7, EZ7 & EZ1, EZ9 & EZ3 .... die Kombinationen der Endziffern mit der Differenz Δ=4.

Hieraus ergeben sich sechs Dirichlet-Progressionen, die nachfolgend in **Tab. 12** unter Angabe der Klassen-Kombinationen angeführt werden:

Tab. 12 - Die sechs Dirichlet-Progressionen in denen Zwillinge mod 4 auftreten können:

Dirichlet-Progression 1:	Die Kombination der Klassen: 3 & 1	mit dem ersten Zwilling mod 4: **(37, 41)**
Dirichlet-Progression 2:	Die Kombination der Klassen: 2 & 3	mit dem ersten Zwilling mod 4: **(13, 17)**
Dirichlet-Progression 3:	Die Kombination der Klassen: 4 & 2	mit dem ersten Zwilling mod 4: **(109, 113)**
Dirichlet-Progression 4:	Die Kombination der Klassen: 6 & 7	mit dem ersten Zwilling mod 4: (3, 7)
Dirichlet-Progression 5:	Die Kombination der Klassen: 7 & 5	mit dem ersten Zwilling mod 4: (7, 11)
Dirichlet-Progression 6:	Die Kombination der Klassen: 8 & 6	mit dem ersten Zwilling mod 4: (19, 23)

Da die (alle) Zwillinge mod 4, die in den sechs (in **Tab. 12** angeführten) Dirichlet-Progressionen auf den ersten Zwilling mod 4 stets nur in Abständen mod 20n, n ≥1 folgen können, ergeben sich die nachfolgend genannten sechs Bildungs-Gesetze, in denen die (alle möglichen) Zwillinge mod 4 auftreten können, gemäß **Tab. 13**::

Tab. 13 - Die sechs Bildungs-Gesetze der Zwillinge mod 4:

Bildungs-Gesetz 1:	**37** + 20n, **41** + 20n .	- der erste Nachfolger ist der Zwilling mod 4: **(97, 101)**, n = 3
Bildungs-Gesetz 2:	**13** + 20n. **17** + 20n	- der erste Nachfolger ist der Zwilling mod 4: **(193, 197)**, n = 9
Bildungs-Gesetz 3:	**109** +20n, **113** + 20n	- der erste Nachfolger ist der Zwilling mod 4: **(229, 233)**, n = 6
Bildungs-Gesetz 4:	3 + 20n, 7 + 20n	- der erste Nachfolger ist der Zwilling mod 4: (43, 47), n = 2
Bildungs-Gesetz 5:	7 + 20n, 11 + 20n	- der erste Nachfolger ist der Zwilling mod 4: (127, 131), n = 6
Bildungs-Gesetz 6:	19 + 20n, 23 + 20n	- der erste Nachfolger ist der Zwilling mod 4: (79, 83), n = 3

Anmerkungen und Schlussfolgerungen:
Die in **Tab. 13** zitierten sechs Bildungs-Gesetze können dadurch vereinfacht werden, dass nur die erste Primzahl (p) angeführt wird, mit der Maßgabe, dass Zwillinge mod 4 nur dann auftreten, wenn auch p +4 prim ist. Dann ergeben sich die folgenden (vereinfachten) sechs Folge - Gesetze für die Zwillinge mod 4:

Gesetz 1: **37** + 20n, Gesetz 2: **13** + 20n, Gesetz 3: **109** + 20n
Gesetz 4: 3 + 20n, Gesetz 5: 7 + 20n, Gesetz 6: 19 + 20n

Unter Beachtung des Sachverhaltes, dass inmitten der Zwillinge mod 4 stets eine nicht prime ungerade Zahl (n_u) situiert ist, die einen Abstand von Δ= ±2 zu den Primzahlen (p, p±4) hat, können auch sechs Folge-Gesetze für diese ungeraden Zahlen (n_u) eingeführt werden, die nachfolgend angegeben wurden:

Folge 1: n_u = 39 + 20n, Folge 2: n_u = 15 + 20n, Folge 3: n_u = 111 + 20n
Folge 4: n_u = 5 + 20n, Folge 5: n_u = 9 + 20n, Folge 6: n_u = 21 + 20n

Für die Folge 1 ergibt sich (bei Anführung aller nicht primen Zahlen, die zwischen den Primzahlen: **37** und **41** situiert sind) die Notation: (**37**, 38, 39, 40, **41**), in der die ungerade Zahl: n_u = 39 unterstrichen markiert wurde.

Ähnlich wie bei den Folgen der geraden Zahlen: $n_g = 12 + 30n$, $n_g = 18 + 30n$, $n_g = 30 + 30n$, die inmitten von Zwillingen mod 2 situiert sein müssen (die fallweise aber nicht in Zwillingen mod 2 situiert sind), müssen die ungeraden Zahlen (n_u) der Folgen: 1, 2, 3, 4, 5, 6 inmitten von Zwillingen mod 4 situiert sein, fallweise aber sind diese ungeraden Zahlen nicht inmitten von Zwillingen mod 4 situiert. Auf jeden Fall können aber nur ungerade Zahlen (n_u) aus den sechs Folgen inmitten von Zwillingen mod 4 situiert sein.

Nachdem die Existenz der sechs Dirichlet-Progressionen für die Zwillinge mod 4 deduziert werden konnte, liegt der (gesuchte) Beweis vor, dass Zwillinge mod 4 in einer unendlichen Zahl in der Menge der natürlichen Zahlen (für $n \to \infty$) auftreten.

In **Tab. 14** wird demonstriert, dass in den Umgebungen der Fakultät: $n_u = \Pi p_i$, $p_1 = 3$ bis $p_n = 13, 17, 19$ und in den Umgebungen von $n_u = 3 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19$ sowie $n_u = 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23$ Zwillinge mod 4 auftreten können.

Tab. 14 - Die in der Umgebung der ungeraden Zahlen $n_u = 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot \ldots \cdot 19$ und $n_u = 3 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19$ und $n_u = 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23$ auftretenden Primzahlen:

	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$:
$n_u = 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 =$ = 15.015	-46	-32	**-2**	**+2**	+16	+38	84
$n_u = 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 =$ = 255.255	-8	-4	**-2**	+4	+58	+74	82
$n_u = 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 =$ = 4.849.845	-58	-32	**-2**	+16	+62	+64	122
$n_u = 3 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 =$ = 969.969	-50	-46	-40	+8	+20	+58	108
$n_u = 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 23 =$ = 7.436.429	-30	-18	-6	+18	+24	+30	60

Anmerkungen:

1. In den Umgebungen aller nicht primen ungeraden Zahlen (n_u) können nur geradzahlige Abstände (Δ) zu den benachbarten Primzahlen: $p_1 < p_2 < p_3 < n_u$ und $p_6 > p_5 > p_4 > n_u$ auftreten, daher können die Primzahlen: p_3 und p_4 offensichtlich niemals Zwillinge mod 2 bilden.

2. Die Primzahlen: p_3 und p_4 können (fallweise) im kleinsten Abstand $\Delta = 4$ auftreten und bilden dann einen Zwilling mod 4. Einen derartigen Zwilling mod 4 bilden die Primzahlen **(15013, 15017)**, der ein Element der Klasse 2 ist - hier wird die Gleichung: $13 + 20n$, $n = 750$ erfüllt.

3. Der Zwilling mod 4: (969919, 969923) ist ein Element der Klasse 6 und erfüllt die Gleichung: $19 + 20n$ $n = 48495$.

4. Der Zwilling mod 4: (255247, 255251) ist ein Element der Klasse 5 und erfüllt die Gleichung $7 + 20n$, $n = 12762$.

5. Im weiteren Umfeld der ungeraden Zahl $n_u = 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 = 4849845$ ist der Zwilling mod 2: (4849781, 4849783) situiert, der ein Element der Folge: $n_g = 12 + 30n$, $n = 161659$ ist.

6. Die Zwillinge mod 4 wurden durch gestrichelte Linien markiert und der Zwilling mod 2 durch eine punktierte Linie.

7. Die kleinsten Abstände ($\Delta = -2$ und $\Delta = +2$, die in der direkten Umgebung der ungeraden Zahlen (n_u) auftreten, wurden fett gedruckt markiert

---

7. Der Beweis, dass alle Zwillinge mod 2n, n > 1 in unendlichen Zahlen auftreten.

Nach dem gleichen Prinzip können Dirichlet-Progressionen für alle Zwillinge mod 2n, n >2 entwickelt werden und dies besagt, dass alle Zwillinge mod 2n, n ≥1 in der unendlichen Zahlen-Menge (n → ∞) in einer unendlichen Zahl auftreten. Dies ist der Beweis, dass nicht nur Zwillinge mod 2 und Zwillinge mod 4 in unendlichen Zahlen auftreten, sondern eine wichtige Verallgemeinerung, für die in der Primzahl-Forschung (bis heute) nicht einmal eine Vermutung aufgestellt wurde.

Für den Beweis der Goldbach'schen Vermutung, der in der Monographie des Verfassers unter dem Titel:

"Der Beweis der Goldbach'schen Vermutung - Die verborgenen Eigenschaften von Primzahlzwillingen mod 2 und von Primzahlzwillingen mod 2n, n > 1"

erbracht wurde, sind die Strukturen (und Eigenschaften) der Zwillinge mod ≥6 besonders wichtig, deshalb wird der Leser auf die Anlagen hingewiesen, die dieser Monographie beigefügt sind. Diese Anlagen sollen deshalb kurz kommentiert werden.

Kommentar zu **Anlage 1**:
Die Anlage 1 trägt den Titel: "Die Ordnungsfolge der ungeraden Primzahlen von p = 3 bis p = 99.991 und die Folge der 1224 Primzahlzwillinge mod 2". Die im .Intervall: n ≤100.000 auftretenden Zwillinge mod 2 wurden hier gelbfarbig markiert, woraus ersichtlich ist, dass die Zwillinge mod 2 oft in Abständen Δ ≥ 4 direkt aufeinander folgen können oder durch Folgen von Zwillingen mod >2 unterbrochen sind. Für die Folge der ersten 36 Zwillinge mod 2 wurden die (stark variierenden) Abstände (Δ) angeführt.

Kommentar zu **Anlage 2**:
In **Anlage 2** wurden für das Intervall n:= 0: bis 10.000 die Folgen der Zwillinge: mod 2, mod 4, mod 6, mod 8, mod 10, mod 100, dargestellt.

Kommentar zu **Anlage 3**:
In **Anlage 3** wurden die Zwillinge mod 2, mod 4, mod 6, mod 8, mod 10 und mod 100 angeführt, die im Intervall n:= 1.000.000 bis 1.010.000 situiert sind.

Eigenschaft 1: Alle Zwillinge mod 2 und alle Zwillinge mod 4, treten mit Ausnahme der direkt verketteten Zwillinge mod 2: (3, 5, 7) und der verketteten Zwillinge mod 4: (3, 7, 11) stets isoliert auf.

Eigenschaft 2: In allen Zwillingen mod 4n, n ≥1 treten stets Primzahlen der gleichen Oberklassen I bzw. II auf.

Eigenschaft 3: In allen Zwillingen mod 2n, n_u = 1, 3, ... treten stets Primzahlen unterschiedlicher Obergruppen: I und II auf. Die Primzahlen der Obergruppe I wurden fett gedruckt und die der Obergruppe II nicht fett gedruckt.

Eigenschaft 4: Die Zwillinge mod n ≥6 können in zwei Varianten: Variante 1 und Variante 2 auftreten. Innerhalb von Zwillingen der Variante 1 treten keine Primzahlen auf, wogegen innerhalb von Zwillingen der Variante 2 auch Primzahlen situiert sein können. Die Zwillinge der zweiten Variante sind deshalb "Kompositionen aus Zwillingen kleinerer Modalität". Ein Zwilling mod 6 der Variante 2 kann deshalb einen Zwilling mod 2 und einen Zwilling mod 4 umfassen.

Eigenschaft 5: Aus Anlage 2 ist zu entnehmen, dass Zwillinge mod >6 in mehrfacher Verkettung auftreten können und diese Eigenschaft führt (schließlich) zum Beweis der Goldbach'schen Vermutung.

Der Beweis, dass es unendlich viele Zwillinge mod 6, mod 8, mod 10, ... , mod 100 gibt:

Für das volle Verständnis dieses Beweises ist die Aufteilung aller (in der unendlichen Folge der Primzahlen auftretenden) Primzahlen in zwei Oberklassen: I und II erforderlich, wonach es acht Klassen von Primzahlen gibt, wogegen Dirichlet nur vier Klassen von Primzahlen erkannt hatte.
Das Prinzip der Teilung aller Primzahlen in acht Klassen ergibt sich (einsichtig) aus **Tab. 15** und **16.**

Tab. 15 Die Gliederung aller natürlichen Zahlen in vier (gleichmächtige) Folgen im Abstand mod 4:
Rest bei Division durch 4, R::=

Die Folge 2 und Folge 4 sind Primzahl-inaktive Folgen, wogegen die Folge 1 und Folge 3 Dirichlet'sche Progressionen bilden, in denen unendlich viele Primzahlen mit den Endziffern: EZ1, EZ3, EZ7, EZ9 auftreten., die sich jedoch durch ihren Rest (Rest 1 bzw. Rest 3) unterscheiden.

In **Tab. 16** werden die in Folge 1 und Folge 3 auftretenden Primzahlen "ausgesiebt" und die zwischen ihnen auftretenden Abstände (Δ) aufgeführt.

Tab. 16 Die Primzahlen der Folge 1 und Folge 3 mit den zwischen ihnen auftretenden Abständen Δ:

Folge 1:	5,	**13,**	**17,**	**29,**	**37,**	**41,**	**53,**	**61,**	**73,**	**89,**	**97,**	**101,**
Δ_1:=		8	*4*	12	8	*4*	12	8	12	16	8	*4*
Folge 3:	3,	7,	11,	19,	23,	31,	43,	47,	59,	67,	71,	79,
Δ_3:=		*4*	*4*	8	*4*	8	12	*4*	12	8	*4*	8

Folge 1:	**109,**	**113,**	**137,**	**149,**	**157,**	**173,**	**181,**	**193,**	**197,**	**229,**	**233,**	**241**
Δ_1:=	8	*4*	24	12	8	16	8	12	*4*	32	*4*	8
Folge 3:	83,	103,	107,	127,	131,	139,	151,	163,	167,	179	191,	199
Δ_3:=	*4*	20	*4*	20	*4*	8	12	12	*4*	12	12	8

Anmerkungen und Schlussfolgerungen:

1. Alle Primzahlen der Folge 1 (Rest 1) wurden fett gedruckt und alle Primzahlen der Folge 3 (Rest 3) wurden nicht fett gedruckt.

2. Zwillinge mod 4, mod 8, mod 12, ... , mod 4n, n ≥1 können, wie aus **Tab. 17** ersichtlich, jeweils nur aus Primzahlen der Folge 1 bzw. der Folge 3 gebildet werden.

3. In der Primzahl-Folge von **Tab. 16** treten insgesamt 14 Zwillinge mod 4 auf - dieser Abstand wurde kursiv notiert. In **Tab. 16** treten auch 14 Zwillinge mod 8 auf, die durch die unterstrichene Zahl 8 markiert wurden.

In **Tab. 17** wurden die ungeraden Primzahlen von p = 3 bis p = 241 in der Reihenfolge ihres Auftretens dargestellt mit Angabe der Abstände, die zwischen ihnen auftreten. Die Primzahlen der Folge 1 wurden hier fett gedruckt und die Primzahlen der Folge 3 nicht fett gedruckt.

Tab. 17 Die Folge der Primzahlen von p = 3 bis p = 241 und ihre Abstände, Δ:

Δ:=	2	2	4	2	4	2	4	6	2	6	4	
p:=	43,	47,	**53,**	59,	**61,**	67,	71,	**73,**	79,	83,	**89,**	**97,**
Δ:=	2	4	6	6	2	6	4	2	6	4	6	8
p:=	**101,**	103,	107,	**109,**	**113,**	127,	131,	**137,**	139,	**149,**	151,	**157,**
Δ:=	4	2	4	2	4	6	4	6	2	10	2	6
p:=	163,	167,	**173,**	179,	**181,**	191,	**193,**	**197.**	199,	**229,**	**233,**	**241**
Δ:=	6	4	6	6	2	10	2	4	2	10	4	8

Anmerkungen und Schlussfolgerungen:

1. Aus **Tab. 17** resultieren wichtige Informationen (Gesetzmäßigkeiten), die aus der Klassenteilung der Primzahlen in die Oberklassen I mit den Klassen: 1 (EZ1), 2 (EZ3), 3 (EZ7), 4 (EZ9) und in die Oberklasse II mit den Klassen: 5 (EZ1), 6 (EZ3), 7 (EZ7), 8 (EZ9) resultieren, die in der Primzahl - Forschung .bisher nicht erkannt wurden. Um diese Gesetzmäßigkeiten wahrzunehmen, wurden alle

1. Aus **Tab. 17** resultieren wichtige Informationen (Gesetzmäßigkeiten), die aus der Klassenteilung der Primzahlen in die Oberklassen I mit den Klassen: 1 (EZ1), 2 (EZ3), 3 (EZ7), 4 (EZ9) und in die Oberklasse II mit den Klassen: 5 (EZ1), 6 (EZ3), 7 (EZ7), 8 (EZ9) resultieren, die in der Primzahl - Forschung .bisher nicht erkannt wurden. Um diese Gesetzmäßigkeiten wahrzunehmen, wurden alle Primzahlen der Oberklasse I, deren Primzahlen bei Division durch die Zahl 4 den Rest 1 (R1) ergeben, fett gedruckt hervorgehoben und alle Primzahlen der Oberklasse II, mit Rest 3 (R3) nicht fett gedruckt.

2. Aus **Tab. 17** resultieren in Verbindung mit **Tab. 16** die folgenden Erkenntnisse (Wahrnehmungen).

Erkenntnis 1:
Alle Zwillingen mod 2, mod 6, ..., mod (2 + 2n, n ≥0) werden stets von Primzahlen gebildet, die Elemente der Oberklassen I und II sind - siehe z.B. die Zwillinge mod 2: (11, **13**), (**17**, 19), (**29**, 31) und die Zwillinge mod 6: (**17**, 23), (23, **29**), (31, **37**), (**61**, 67), ..., (**101**, 107).

Erkenntnis 2:
Die einzigen "verketteten" Zwillinge mod 2 bilden die Zwillinge mod 2: (3, **5**) ---(**5**, 7) und die einzigen verketteten Zwillinge mod 4 bilden die Zwillinge mod 4: (3, 7) --- (7, 11). Alle auf diese verketten Zwillinge mod 2 und Zwillinge mod 4 folgenden Zwillinge mod 2 und Zwillinge mod 4 können nicht als verkettete (direkt benachbarte) Zwillinge auftreten.

Erkenntnis 3:
Zwillinge mod 6 können als verkettete Zwillinge auftreten . siehe z.B. die folgenden Zwillinge mod 6: (47, **53**) --- (**53**, 59) und (167, **173**) --- (**173**, 179).

Erkenntnis 4:
Zwillinge mod 6 können aus direkt aufeinander folgenden Zwillingen mod 2 und Zwillingen mod 4 "komponiert" sein - siehe z.B. die Zwillinge: (**17**, 19) --- (19, 23), die den Zwilling mod 6: (**17**, 23) bilden sowie die Zwillinge: (67, 71) --- (71, **73**), die den Zwilling mod 6: (67, **73**) bilden. Aus diesen Beispielen ist ersichtlich, dass die Zwillinge mod 2 und mod 4, die einen Zwilling mod 6 bilden, links bzw. rechts im Zwilling mod 6 situiert sein können.

Erkenntnis 5:
Der Zwilling mod 8: (3, 11), der aus den beiden verketteten Zwillingen mod 4: (3, 7) --- (7, 11) gebildet wird, ist der einzige Zwilling mod 8 dieser Art, da alle folgenden Zwillinge mod 4 niemals in einer direkten Verkettung auftreten können.

Erkenntnis 6:
Zwillinge mod 8 können aus zwei direkt aufeinander folgenden Primzahlen gebildet werden, die im Abstand (Δ=8) auftreten - siehe z.B. die Zwillinge mod 8: (**89**, **97**) und (**233**, **241**).

Erkenntnis 7:
Zwillinge mod 8 können aus den folgenden Kompositionen von Zwillingen gebildet werden:

Variante 1:
Zwilling mod 2 gefolgt von Zwilling mod 4 gefolgt von Zwilling mod 2 - siehe z.B. die folgenden Zwillinge: (11, **13**) --- (**13**, **17**) --- (**17**, 19), die den Zwilling mod 8: (11, 19) ergeben.

Variante 2:
Zwilling mod 6 gefolgt von Zwilling mod 2 bzw. Zwilling mod 2 gefolgt von Zwilling mod 6 - siehe z.B.: (23, **29**) --- (**29**, 31) mit dem Zwilling mod 8: (23, 31) sowie (71, **73**) --- (**73**, 79) mit dem Zwilling mod 8: (71, 79).

Erkenntnis 8:
Aus **Tab. 17** kann entnommen werden, dass Zwillinge mod > 8 direkt aus zwei aufeinander folgenden Primzahlen gebildet werden können oder aber aus (zulässigen) Kompositionen von Zwillingen kleinerer Modularität, die in Verkettung aufeinander folgen. Diese Eigenschaft der Semi-Zwillinge ist für den Beweis der Goldbach'schen Vermutung von besonderer Bedeutung - siehe die Monographie u.d.T.:

Der Beweis, dass es unendlich viele Zwillinge mod 6 gibt, kann nun angetreten werden:

Aus den auf der Basis von **Tab. 16** und **Tab. 17** vorgenommenen Analysen lassen sich drei Folgen herleiten, in denen Zwillinge mod 6 (fallweise) auftreten können. Dabei ist beachtlich, dass alle Zwillinge mod 6 nur von Primzahlen: p_1, p_2 mit den folgenden Endziffern gebildet werden können:

Variante 1: p_1 (EZ1) und p_2 (EZ7) - Beispiele: (11, **17**), (31, **37**), (**61**, 67) - Folge 1
Variante 2: p_1 (EZ3) und p_2 (EZ9) - Beispiele: (**13**, 19). (23, **29**), (**53**, 59) - Folge 2
Variante 3: p_1 (EZ7) und p_2 (EZ3) - Beispiele: (7, **13**), (**37**, 43). (47, **53**) - Folge 3

In den Folgen der Zwillinge mod 6 kommt der Zwilling mod 6: (5, 11), in dem die Primzahl p_1 (EZ5) auftritt, solitär vor, da alle mehrstelligen ungeraden Zahlen n_u (EZ5) nicht prim sein können. Den in den o.g. Beispielen zitierten Folgen entsprechen jeweils zwei Dirichlet-Progressionen, in denen (bekanntlich) unendlich viele Primzahlen mit gleichen Endziffern (EZ) auftreten , in denen die hier auftretenden Primzahlen (p_1, p_2) nach ihren Klassen - Zugehörigkeiten fett bzw. nicht fett gedruckt wurden.

D - Progression 1: $11 + 10n$: $p_1 = 11$, $n_u = 21$, $p_1 = 31$, $p_1 =$ **41**, $n_u = 51$, $p_1 =$ **61**, ... Folge 1
D - Progression 2: **17** + 10n: $p_2 =$ **17**, $n_u = 27$, $p_2 =$ **37**, $p_2 = 47$. $n_u = 57$, $p_2 = 67$, ... Folge 1

D - Progression 3: **13** + 10n: $p_1 =$ **13**, $p_1 = 23$, $n_u = 33$, $p_1 = 43$, $p_1 =$ **53**, $n_u = 63$, ... Folge 2
D - Progression 4: $19 + 10n$: $p_2 = 19$, $p_2 =$ **29**, $n_u = 39$, $n_u = 49$, $p_2 = 59$, $n_u = 69$, ... Folge 2

D - Progression 5: $7 + 10n$: $p_1 = 7$, $p_1 =$ **17**, $n_u = 27$, $p_1 =$ **37**, $p_1 = 47$, $n_u = 57$, ... Folge 3
D - Progression 6: **13** +10n: $p_2 =$ **13**, $p_2 = 23$, $n_u = 33$, $p_2 = 43$, $p_2 = 43$, $n_u = 63$, ... Folge 3

Die Existenz dieser sechs Dirichlet - Progressionen, in denen fallweise Primzahlen (p_1, p_2) jeweils im Abstand $\Delta = 6$ auftreten, ist bereits der Beweis, dass Zwillinge mod 6 in einer unendlichen Zahl in der Folge der natürlichen Zahlen $n \rightarrow \infty$ auftreten. Für diesen Beweis ist es beachtlich, dass Zwillinge mod 6 fallweise aus einem Zwilling mod 2 und einem Zwilling mod 4 zusammengesetzt sein können, wobei dann innerhalb dieser Zwillinge mod 6 eine Primzahl situiert ist.
Unter Beachtung der innerhalb der Zwillingen mod 6 der Folge 1, Folge 2 und Folge 3 situierten Zahlen, können die nachfolgenden drei (vereinfachten) Bildungsgesetze gefunden werden:

1. Aus der Analyse der Zahlenfolgen, mit den unterstrichenen inmitten situierten geraden Zahlen:
(11, 12, **13**, 14, 15, 16, **17**), (21, 22, 23, 24, 25, 26, 27), (31, 32, 33, 34, 35, 36, **37**), ...
kann das nachstehende Bildungsgesetz der Zwillinge mod 6 für die Folge 1 definiert werden:

Das Gesetz der Folge 1: $n_g = 14 + 10n$ .... in allen Zwillingen mod 6 der Folge 1 treten die Zahlen $n_g = 14 + 10n$ (fallweise) inmitten der Zwillinge auf.

2. Aus der Analyse der Zahlenfolgen, mit den unterstrichenen inmitten situierten geraden Zahlen:
(**13**, 14, 15, 16, **17**, 18, 19), (23, 24, 25, 26, 27, 28, **29**), (33, 34, 35, 36, **37**, 38, 39), ...
kann das nachstehende Bildungsgesetz der Zwillinge mod 6 für die Folge 2 definiert werden:

Das Gesetz der Folge 2: $n_g = 16 + 10n$ .... in allen Zwillingen mod 6 der Folge 2 treten die Zahlen $n_g = 16 + 10n$ (fallweise) inmitten der Zwillinge auf.

3. Aus der Analyse der Zahlenfolgen, mit den unterstrichenen inmitten situierten geraden Zahlen:
(7, 8, 9, 10, 11, 12, **13**), (**17**, 18, 19, 20, 21, 22, 23), (27, 28, **29**, 30, 31, 32, 33), ...
kann das nachstehende Bildungsgesetz der Zwillinge mod 6 für die Folge 3 definiert werden:

Das Gesetz der Folge 3: $n_g = 10 + 10n$ ... in allen Zwillingen mod 6 der Folge 3 treten die Zahlen $n_g = 10 + 10n$ (fallweise) inmitten der Zwillinge auf.

(7, 8, 9, 10, 11, 12, **13**), (**17**, 18, 19, 20, 21, 22, 23), (27, 28, **29**, 30, 31, 32, 33), ...
kann das nachstehende Bildungsgesetz der Zwillinge mod 6 für die Folge 3 definiert werden:

Das Gesetz der Folge 3: $n_g = 10 + 10n$ ... in allen Zwillingen mod 6 der Folge 3 treten die Zahlen $n_g = 10 + 10n$ (fallweise) inmitten der Zwillinge auf.

In einer (gewissen) Analogie zu den drei Folgegesetzen der Zwillinge mod 2 ergeben sich für alle Zwillinge mod 6 die drei Folgegesetze:

1. Folgegesetz: $14 + 10n$, 2. Folgegesetz: $16 + 10n$, 3. Folgegesetz: $10 + 10n$

Wie bereits aus den angeführten Beispielen hervorgeht, sind nicht alle geraden Zahlen der drei Folgen inmitten von Zwillingen mod 6 situiert, es kann jedoch keine Zwillinge mod 6 geben, in denen andere gerade Zahlen inmitten situiert sind, als die (fallweisen) geraden Zahlen der drei Folgen - dies wird in **Tab. 18** an drei Beispielen von Zwillingen mod 6 demonstriert, die in einem relativ großen Zahlen-Intervall auftreten:

Tab. 18 - Beispiele für Zwillinge mod 6 im Zahlen-Intervall n > 2.000.000:

Zwilling mod 6 der Folge 1:

$n_g = 14 + 10n$	n:=	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$:=
$n_g = 2.000.224$ $= 14 + 200021 \cdot 10$ $= 2^5 \cdot 62507$	200.021	-47	-15	-3 -------	+3	+25	+37	94

Zwilling mod 6 der Folge 2:

$n_g = 16 + 10n$	n:=	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$:=
$n_g = 2.000.426$ $= 16 + 200021 \cdot 10$ $= 2 \cdot 1000213$	200.021	-33	-9 -------	-3 -------	+3	+21	+71	104

Anmerkung:
In der Umgebung der Zahl $n_g = 2.000.426$ treten zwei Zwillinge mod 6 auf, die direkt aufeinander folgen, es sind dies die Zwillinge mod 6: (**2000417**, 2000423) --- (2000423, **2000429**). Diese Zwillinge mod 6 wurden durch gestrichelte Linien markiert.

Zwilling mod 6 der Folge 3:

$n_g = 10 + 10n$	n:=	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$:=
$n_g = 2.000.110$ $= 10 + 200010 \cdot 10$ $= 2 \cdot 5 \cdot 7 \cdot 28573$	200.010	-27	-17	-3 ------	+3	+33	+37	64

Anmerkungen:

1. Alle in **Tab. 18** gefundenen Zwillinge mod 6 sind nicht aus Zwillingen mod 2 und Zwillingen mod 4 gebildet. .Derartige "komponierte" Zwillinge mod 6 treten dagegen häufiger in kleineren Zahlen-Intervallen auf.

2. Die relativ großen Primzahl-Dichten in den Umgebungen der hier analysierten geraden Zahlen (n_g) finden ihre Begründung in den "sehr lückenhaften" Primzahl-Fakultäten dieser geraden Zahlen..

Die Schlussfolgerung lautet:
Es wurde einsichtig der Beweis erbracht, dass Zwillinge mod 6 in einer unendlichen Zahl auftreten und darüber hinaus wurden die möglichen Strukturen der Zwillinge mod 6 genau herausgestellt.

Bevor der Beweis vorgestellt wird, dass auch Zwillinge mod 8 in einer unendlichen Zahl auftreten, werden für die Zwillinge mod 4 drei Zahlenfolgen für die (zentral) inmitten der Zwillinge mod 4 situierten ungeraden Zahlen (n_u) deduziert, in denen (fallweise) Zwillinge mod 4 auftreten können.

Es wurde bereits unter Beweis gestellt, dass es nur einen Zwilling mod 4 gibt, der aus zwei aufeinander folgenden Zwillingen mod 2 konstituiert ist, nämlich den Zwilling mod 4 (3, 5) --- (5, 7), und dass alle folgenden Zwillinge mod 4 bis auf die verketteten Zwillinge mod 4: (3, 7) --- (7, 11) stets in definierten Abständen auftreten müssen, was in **Anlage 2** und **Anlage 3** bestätigt wird.
Danach können die bereits deduzierten sechs Folgen, in denen die inmitten von Zwillingen mod 4 situierten ungeraden Zahlen (n_u) auftreten, zu den folgenden drei Folgen zusammengefasst werden.

Folge 1: $n_u = 19 + 20n$,	Folge 2: $n_u = 15 + 20n$	Folge 3: $n_u = 11 + 20n$
Folge 4: $n_u = 5 + 20n$,	Folge 5: $n_u = 9 + 20n$,	Folge 6: $n_u = 21 + 20n$

Die Zusammenfassung der Folgen mit den gleichen Endziffern führt zu den kombinierten Folgen:

Folge 1: $n_u = 9 + 10n$,	*Folge 2:* $n_u = 5 + 10n$,	*Folge 3:* $n_u = 11 + 10n$

Die kursiv notierten Folgen sind die Bildungsgesetze der ungeraden Zahlen, die (fallweise) inmitten der Zwillinge mod 4 auftreten. Die Folgen: *Folge 1*, *Folge 3* bilden zwei Dirichlet - Progressionen in denen unendlich viele prime und nicht prime ungerade Zahlen mit den (gleichen) Endziffern: EZ9 bzw. EZ1 auftreten. Nachdem inmitten von Zwillingen mod 4 keine Primzahlen auftreten können, scheiden alle Primzahlen der *Folge 1* und der *Folge 3* aus. Die *Folge 2* ist dagegen eine "Primzahl - inaktive" Dirichlet - Progression, in der (bis auf die Primzahl p = 5) nur ungerade Zahlen mit der Endziffer EZ5 auftreten.
Die Existenz der unendlichen Folgen: *Folge 1*, *Folge 2*, *Folge 3* bilden somit den Beweis, dass die Zwillinge mod 4 in einer unendlichen Zahl im Zahlen-Intervall $n \to \infty$ auftreten.
In **Tab. 19** finden alle logischen Kriterien, die zu diesem Beweis geführt haben, eine Bestätigung:

Tab. 19

Folge 1:	*Folge 2:*	*Folge 3:*	n:=
(7, 8, 9, 10, 11), +	(**3**, 4, 5, 6, 7), +	(*9*, 10, **11**, 12, **13**), -	0
(17, 18, **19**, 20, *21*), -	(**13**, 14, 15, 16, **17**), +	(**19**, 20, 21, 22, **23**), +	1
(*27*, 28, **29**, 30, 31), -	(**23**, 24, 25, 26, 27), -	(**29**, 30, **31**, 32, *33*), -	2
(37, 38, 39, 40, 41), +	(*33*, 34, 35, 36, **37**), -	(*39*, 40, **41**, 42, **43**), -	3
(47, 48, 49, 50, *51*), -	(**43**, 44, 45, 46, **47**), +	(*49*, 50, 51, 52, **53**), -	4
(*57*, 58, **59**, 60, 61), -	(**53**, 54, 55, 56, *57*), -	(**59**, 60, 61, 62, *63*), -	5
(**67**, 68, 69, 70, **71**), +	(*63*, 64, 65, 66, **67**), -	(*69*, 70, 71,72, **73**), -	6
(*77*, 78, 79, 80, *81*), -	(**73**, 74, 75, 76, *77*), -	(**79**, 80, 81, 82, **83**), +	7
(*87*, 88, **89**, 90, *91*), -	(**83**, 84, 85, 86, *87*), -	(**89**, 90, 91, 92, *93*), -	8
(**97**, 98, 99, 100, **101**), +	(*93*, 94, 95, 96, **97**), -	(*99*, 100, **101**, 102. **103**), -	9
(**107**, 108, **109**, 110, *111*), -	(**103**, 194, 195, 106, **107**), +	(**109**, 110, 111, 112, **113**), +	10

Anmerkungen und Hinweise:

(97, 98, 99, 100, 101), +	(93, 94, 95, 96, 97), -	(99, 100, 101, 102. 103), -	9
(107, 108, 109, 110, 111), -	(103, 194, 195, 106, 107), +	(109, 110, 111, 112, 113), +	10

Anmerkungen und Hinweise:

1. In *Folge 1* und *Folge 2* treten in der Index-Folge: n = 0 bis 10 vier Zwillinge mod 4 auf und in der *Folge 3* drei Zwillinge mod 4. Diese Zwillinge wurden durch das Zeichen + markiert. Jeweils sechs bzw. sieben Folge-Glieder bilden keine Zwillinge mod 4. Diese Folge-Glieder wurden mit dem Zeichen - markiert. Alle inmitten der Folge-Glieder situierten ungeraden Zahlen mit den jeweiligen Endziffern: EZ9, EZ5, EZ1 wurden in **Tab. 30** durch Unterstreichung markiert. Des weiteren wurden alle primen Zahlen fett gedruckt hervorgehoben.

2. In **Tab. 19** wurden die Ursachen verdeutlicht, die dazu führen, dass einzelne Folge-Glieder keine Zwillinge mod 4 bilden können - diese Ursachen sind:

   Ursache 1:
   Die inmitten der Folgeglieder situierten ungeraden Zahlen sind fett gedruckte Primzahlen.
   Ursache 2:
   Eine bzw. beide "Randzahlen" der Folge-Glieder sind keine Primzahlen. Diese "Randzahlen" wurden kursiv geduckt markiert.

3. Aus **Tab. 19** folgt eine (gewisse) Gleichverteilung der Zwillinge mod 4 in *Folge 1, Folge 2, Folge 3* und diese Gesetzmäßigkeit der Zwillinge mod 4 wird in **Anlage 2** vollauf bestätigt,

---

Der Beweis, dass Zwillinge mod 8 in einer unendlichen Zahl vorkommen:

Eine analoge "logische Struktur-Analyse" führt zum Beweis, dass auch Zwillinge mod 8 in einer unendlichen Zahl auftreten. Zwillinge mod 8, die links von der Primzahl (p_1) und rechts von der Primzahl (p_2) abgegrenzt sind, erfordern, dass diese Primzahlen in den folgenden Endziffern auftreten:

Variante 1: (p_1, EZ3, p_2, EZ1) Variante 2: (p_1, EZ9, p_2, EZ7) Variante 3: (p_1, EZ1, p_2, EZ9)

Aus **Tab. 20** können die ungeraden Zahlen (n_u) bestimmt werden, die in *Folge 1, Folge 2* und in *Folge 3* inmitten von Zwillingen mod 8 situiert sein müssen:

Tab. 20 - Die ungeraden zentral situierten Zahlen der *Folge 1, Folge 2 und Folge 3* von n = 0 bis n = 9:

Folge 1: $n_u = 7 + 10n$	*Folge 2:* $n_u = 13 + 10n$	*Folge 3:* n;= $n_u = 15 + 10n$	n:=
(3, 4, 5, 6, 7, 8, 9, 10, 11), +	(9, 10, 11, 12, 13, 14, 15, 16, 17), -	(11, 12, 13, 14, 15, 16, 17, 18, 19), +	0
(13, 14, 15, 16, 17, 18, 19, 20, 21), -	(19, 20, 21, 22, 23, 24, 25, 26, 27), -	(21. 22, 23, 24, 25, 26, 27, 28, 29), -	1
(23, 24, 25, 26, 27, 28, 29, 30, 31), +	(29, 30, 31, 32, 33, 34, 35, 36, 37), +	(31. 32. 33, 34, 35, 36, 37,38, 39), -	2
(33, 34, 35, 36, 37, 38, 39, 40, 41), -	(39, 40, 41, 42, 43, 44, 45, 46, 47), -	(41, 42, 43, 44, 45, 46, 47, 48, 49), -	3
(43, 44, 45, 46, 47, 48, 49, 50, 51), -	(49, 50, 51, 52, 53, 54, 55, 56, 57), -	(51, 52, 53, 54, 55, 56, 57, 58, 59), -	4
(53, 54, 55, 56, 57, 58, 59, 60, 61), +	(59, 60, 61, 62, 63, 64, 65, 66, 67), +	(61, 62, 63, 64, 65, 66, 67, 68, 69), -	5
(63, 64, 65, 66, 67, 68, 69, 70, 71), -	(69, 70, 71, 72, 73, 74, 75, 76, 77), -	(71, 72, 73, 74, 75, 76, 77, 78, 79), +	6
(73, 74, 75, 76, 77, 78, 79, 80, 81), -	(79. 80, 81, 82, 83, 84, 85, 86, 87), -	(81, 82, 83, 84, 85, 86, 87, 88, 89), -	7
(83, 84, 85, 86, 87, 88, 89, 90, 91), -	(89. 90, 91, 92, 93, 94, 95, 96, 97), +	(91, 92, 93, 94, 95, 96, 97, 98, 99), -	8

1 In **Tab. 20** treten in *Folge 1*, *Folge 2* und *Folge 3* jeweils drei Zwillinge mod 8 auf. Diese Zwillinge wurden mit dem Zeichen + markiert.

2. In **Tab. 20** wurden alle Primzahlen fett gedruckt und die inmitten (zentral) situierten ungeraden Zahlen unterstrichen hervorgehoben.

3. Wie aus **Tab. 20** erkannt werden kann, können Zwillinge mod 8 in drei Varianten auftreten:

<u>Variante 1:</u>
Zwillinge mod 8, in denen nur zwei Primzahlen (p_1, p_2) auftreten, die diese Zwillinge beranden.

<u>Variante 2:</u>
Zwillinge mod 8 können aus einem Zwilling mod 6 und einem Zwilling mod 2 "konstituiert" sein; dann treten in diesen Zwillingen mod 8 drei Primzahlen: $p_1 < p_2 < p_3$ auf.

<u>Variante 3:</u>
Zwillinge mod 8 können aus einem Zwilling mod 2 gefolgt von einem Zwilling mod 4 und gefolgt von einem Zwilling mod 2 konstituiert sein; dann treten in diesem Zwilling mod 8 vier Primzahlen.
$p_1 < p_2 < p_3 < p_4$ auf, wie in der *Folge 3* für $n = 0$ und $n = 9$.

In **Tab. 21** wird der Nachweis erbracht, dass Zwillinge mod 8 nur in *Folge 1*, *Folge 2* und *Folge 3* in beliebig großen Zahlen-Intervallen auftreten können, womit der Beweis, dass Zwillinge mod 8 in einer unendlichen Zahl auftreten, einsichtig bestätigt wird.

Tab. 21 - Beispiele für Zwillinge mod 8 in *Folge 1*, *Folge 2*, *Folge 3* in großen Zahlenbereichen:

Folge 1:

$n_u = 7 + 10n$	n:=	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$:=
n_u = 10000000001727	1000000000172	-68	-58	-4 ------	+4	+82	+104	172
= 7 + 1000000000000001720 =								
= 7 + 1000000000172 · 10 =								
= $3^2 \cdot 7 \cdot 43 \cdot 36913990403$								

Folge 2:

$n_u = 13 + 10n$	n:=	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$:=
n_u = 10000000000001193=	10000000000000118	-20	-16	-4 ------	+4	+10	+80	100
= 13 + 10000000000180 =								
= 13 + 1000000000118 · 10 =								
= $3 \cdot 41 \cdot 813008130091$								

Folge 3:

$n_u = 15 + 10n$	n;=	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$:=:
n_u = 1000000000000065	100000000000005	-148	-128	-4 ------	+4	+14	+34	182
= 15 + 1000000000000065=								
= 15 + 100000000000005 · 10=								
= $3 \cdot 7 \cdot 17^2 \cdot 191 \cdot 223 \cdot 38685209$								

<u>Anmerkungen und Schlussfolgerungen:</u>

1. Aus **Tab. 21** kann entnommen werden, dass in den Umgebungen der ungeraden Zahlen (n_u), die zentral inmitten von Zwillingen mod 8 situiert sind, relativ große Primzahl-Dichten auftreten.

2. Ursächlich für diese relativ großen Primzahl-Dichten sind die stark lückenhaften Primzahl-Fakultäten, aus denen diese ungeraden Zahlen der *Folge 1*, *Folge 2* und der *Folge 3* konstituiert sind.

In **Tab. 22** wird der Nachweis erbracht, dass in den Umgebungen der relativ großen Primzahlen, die fakultativ in die in **Tab. 21** analysierten ungeraden Zahlen (n_u) der *Folge 1*, *Folge 2* und *Folge 3*

2. Ursächlich für diese relativ großen Primzahl-Dichten sind die stark lückenhaften Primzahl-Fakultäten, aus denen diese ungeraden Zahlen der *Folge 1, Folge 2* und der *Folge 3* konstituiert sind.

In **Tab. 22** wird der Nachweis erbracht, dass in den Umgebungen der relativ großen Primzahlen, die fakultativ in die in **Tab. 21** analysierten ungeraden Zahlen (n_u) der *Folge 1, Folge 2* und *Folge 3* eingehen, wiederum Zwillinge mod 8 situiert sind. In **Tab. 22** wurden deshalb jeweils zwölf Primzahlen: $p_1 < p_2 < p_3 < p_4 < p_5 < p_6 < n_u = p < p_7 < p_8 < p_9 < p_{10} < p_{11} < p_{12}$ bestimmt.

Tab. 22 - Die Zwillinge mod 8 in den Umgebungen der Primzahlen: p = 36913990403, p = 813008130091 und p = 38685209, die fakultativ in die ungeraden Zahlen (n_u) der *Folge 1, Folge 2, Folge 3* von **Tab. 21** eingehen:

	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	Δ_7	Δ_8	Δ_9	Δ_{10}	Δ_{11}	Δ_{12}	$\Sigma\Delta=$
p = 36913990403													
Δ:=	-222	-132	<u>-120</u>	<u>-112</u>	-102	-36	+78	+96	+146	+224	+240	+258	480
p = 813008130091													
Δ:=	-152	<u>-62</u>	<u>-54</u>	-18	-12	<u>-8</u>	+6	+48	+66	+108	+148	+156	308
p = 38685209													
Δ:=	-102	-76	<u>-48</u>	<u>-40</u>	-18	-16	+44	+50	<u>+72</u>	<u>+80</u>	+132	+150	252

<u>Anmerkungen und Hinweise:</u>

1. In **Tab. 22** wurden ein Zwillinge mod 8 bestimmt, der im Umfeld der Primzahl p = 36813990403 auftritt - dieser Zwilling mod 8 (36913990283, 36913990291) wurde unterstrichen markiert.

   Dieser Zwilling mod 8 ist ein Element der *Folge 2*.

2. Im Umfeld der Primzahl p = 813008130091 treten zwei Zwillinge mod 8 auf - dies sind die folgenden Zwillinge mod 8: (813008130083, 813008130091) und (813008130029, 813008130037).
   Wie ersichtlich, geht die (fakultative) Primzahl p = 813008130091 direkt in einen Zwilling mod 8 ein, der ein Element der *Folge 2* ist, derweil der Zwilling mod 8 (813008130029, 813008130037) ein Element der *Folge 3* ist.

3. Im Umfeld der Primzahl p = 38685209 treten zwei Zwillinge mod 8 auf, nämlich der Zwilling mod 8 (38685161, 38685169) und (38685281, 38685289), die beide Elemente der *Folge 3* sind.

4. In großen Zahlen-Intervallen treten (offensichtlich) überwiegend "reine" Zwillinge mod 8 auf, also die Zwillinge mod 8 der Variante 1 (siehe **Tab. 21**).

Nachdem der Beweis erbracht wurde, dass Zwillinge mod 2, mod 4, mod 6, mod 8 in unendlicher Zahl auftreten, soll anschließend der Beweis erbracht werden, dass auch Zwillinge mod 10 in einer unendlichen Zahl auftreten. In **Tab. 22** wird nachgewiesen, dass Zwillinge mod 10 in vier unendlichen Folgen: *Folge 1, Folge 2, Folge 3* und *Folge 4* (fallweise) auftreten, für die die folgenden Bildungsgesetze gelten:

<u>Bildungsgesetz der *Folge 1*:</u> $n_g = 6 + 10n, \ n \geq 0$

<u>Bildungsgesetz der *Folge 2*:</u> $n_g = 8 + 10n, \ n \geq 0$

<u>Bildungsgesetz der *Folge 3*:</u> $n_g = 12 + 10n, \ n \geq 0$

Tab. 22 - Die Bildungsgesetze der *Folge 1, Folge 2, Folge 3* und *Folge 4*: der Zwillinge mod 10:

n	*Folge 1*: n_g = 6 + 10n	*Folge 2*: n_g = 8 + 10n
n = 0	(1, **2**, **3**, 4, **5**, 6, **7**, 8, 9, 10, **11**). -	(**3**, 4, **5**, 6, **7**, 8, 9, 10, **11**, 12, **13**), +
n = 1	(**11**, 12, **13**, 14, 15, 16, **17**, 18, 19, 20, 21), -	(**13**, 14, 15, 16, **17**, 18, **19**, 20, 21, 22, **23**),+
n = 2	(21, 22, **23**, 24, 25, 26, 27, 28, **29**, 30, **31**), -	(**23**, 24, 25, 26, 27, 28, **29**, 30, **31**, 32, 33), -
n = 3	(**31**, 32, 33, 34, 35, 36, **37**, 38, 39, 40, **41**), +	(33, 34, 35, 36, **37**, 38, 39, 40, **41**, 42, **43**), -
n = 4	(**41**, 42, **43**, 44, 45, 46, **47**, 48, 49, 50, 51). -	(**43**, 44, 45, 46, **47**, 48, 49, 50, 51, 52, **53**), +
n = 5	(51, 52, **53**, 54, 55, 56, 57, 56, **59**, 60, **61**), -	(**53**, 54, 55, 56, 57, 58, **59**, 60, **61**, 62, 63), -
n = 6	(**61**, 62, 63, 64, 65, 66, **67**, 68, 69, 70, **71**), +	(63, 64, 65, 66, **67**, 68, 69, 70, **71**, 72, **73**). -
n = 7	(**71**, 72, **73**, 74, 75, 76, 77, 78, **79**, 80, 81), -	(**73**, 74, 75, 76, 77, 78, **79**, 80, 81, 82, **83**), +
n = 8	(81, 82, **83**, 84, 85, 86, 87, 88, **89**, 90, 91), -	(83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93), -
n = 9	(91, 92, 93, 94, 95, 96, **97**, 98, 99, 100, **101**), -	(93, 94, 95, 96, **97**, 98, 99, 100, **101**, 102, **103**), -
n = 10	(**101**, 102, **103**, 104, 105, 106, **107**, 108, **109**, 110, 111), -	(**103**, 104, 105, 106, **107**, 108, 109, 110, 111, 112, **113**),+
n = 18	(**181**, 182, 183, 184, 185, 186, 187, 188, 189, 190, **191**), +	
n = 28	→	(**283**, 284, 285, 286, 287, 288, 289, 290, 291, **293**), +

n	*Folge 3*: n_g = 12 + 10n	*Folge 4*: n_g = 14 + 10n
n = 0	(7, 8, 9, 10, **11**, 12, **13**, 14, 15, 16, **17**), +	(9, 10, **11**, 12, **13**, 14, 15, 16, **17**, 18, **19**), -
n = 1	(**17**, 18, **19**, 20, 21, 22, **23**, 24, 25, 26, 27), -	(**19**, 20, 21, 22, **23**, 24, 25, 26, 27, 28, **29**), +
n = 2	(27, 28, **29**, 30, **31**, 32, 33, 34, 35, 36, **37**), -	(**29**, 30, **31**, 32, 33, 34, 35, 36, **37**, 38, 39), -
n = 3	(**37**, 38, 39, 40, **41**, 42, **43**, 44, 45, 46, **47**), +	(39, 40, **41**, 42, **43**, 44, 45, 46, **47**, 48, 49), -
n = 4	(**47**, 48, 49, 50, 51, 52, **53**, 54, 55, 56, 57), -	(49, 50, 51, 52, **53**, 54, 55, 56, 57, 58, **59**), -
n = 5	(57, 58, **59**, 69, **61**, 52, 63, 64, 65, 66, **67**), -	(**59**, 60, **61**, 62, 63, 64, 65, 66, **67**, 68, 69), -
n = 6	(**67**, 68, 69, 70, **71**, 72, **73**, 74, 75, 76, 77), -	(69, 70, **71**, 72, **73**, 74, 75, 76, 77, 78, **79**), -
n = 7	(77, 78, **79**, 80, 81, 82, **83**, 84, 85, 86, 87), -	(**79**, 80, 81, 82, **83**, 84, 85, 86, 87, 88, **89**), +
n = 8	(87, 88, **89**, 90, 91, 92, 93, 94, 95, 96, **97**), -	(**89**, 90, 91, 92, 93, 94, 95, 96, **97**, 98, 99), -
n = 9	(**97**, 98, 99, 100, **101**, 102, **103**, 104, 105, 106, **107**), +	(99, 100, **101**, 102, **103**, 104, 105, 106, **107**, 108, **109**), -
n = 10	(**107**, 108, **109**, 110, 111, 112, **113**, 114, 115, 116, 117), -	(**109**, 110, 111, 112, **113**, 114, 115, 116, 117, 118, 119), -
n = 33	(**337**, 339, 339, 340, 341, 342, 343, 344, 345, 346, **347**), +	
n = 70	→	(**709**, 710, 711, 712, 713, 714, 715, 716, 717, 718, **719**), +

Anmerkungen und Schlussfolgerungen:

1. Die *in Folge 1, Folge 2, Folge 3 und Folge 4* auftretenden Zwillinge mod 10 wurden mit dem Zeichen: + markiert und die Elemente dieser Folgen, die keine Zwillinge mod 10 bilden mit dem Zeichen: - . Alle Primzahlen, die in den Elementen der Folgen auftreten wurden fett gedruckt und die zentral inmitten dieser Folgen auftretenden geraden Zahlen (n_g) wurden unterstrichen hervorgehoben.

2. Aus **Tab. 22** gewinnt man die Wahrnehmung, dass Zwillinge mod 10 in den folgenden vier Varianten auftreten können:

   Variante 1: Als Komposition eines Zwillings mod 6 und eines Zwillings mod 4. in beliebiger Folge.

   Variante 2: Als Komposition eines Zwillings mod 2, gefolgt von einem Zwilling mod 4 und einem Zwilling mod 2.

   Variante 3: Als Komposition eines Zwillings mod 4, gefolgt von einem Zwilling mod 2 und einem Zwilling mod 4.

   Variante 4: Als Zwilling mod 10, in dem keine Zwillinge mod 2, mod 4, mod 6 situiert sind..

3. Die Zwillinge mod 10 der 4. Variante (die als "reine" Zwillinge mod 10 bezeichnet werden können), treten, wie aus **Anlage 2** entnommen werden kann, überwiegend in großen Zahlen-Intervallen auf.

Variante 4: Als Zwilling mod 10, in dem keine Zwillinge mod 2, mod 4, mod 6 situiert sind..

3. Die Zwillinge mod 10 der 4. Variante (die als "reine" Zwillinge mod 10 bezeichnet werden können), treten, wie aus **Anlage 2** entnommen werden kann, überwiegend in großen Zahlen-Intervallen auf.

In **Tab. 23** wurden Zwillinge mod 10 der *Folge 1*, *Folge 2*. *Folge 3* und *Folge 4* vorgestellt, die in großen Zahlen-Intervallen auftreten, wie auch die Primzahl-Dichten im Umfeld dieser Zwillinge.

Tab. 23 - Die Bestimmung von je zwei Zwillingen mod 10 in *Folge 1*, *Folge 2*, *Folge 3*, *Folge 4* in dem Zahlen-Intervall von (Δ= 6774 und der Primzahl-Dichten im Umfeld dieser Zwillinge mod 10:

Folge :	n;=	Δ_1	Δ_2	Δ_2	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$
Folge 1								
$n_g = 6 + 100000000004730$	1000000000473	-155	-127	-5 --------	+5	+31	+133	288
$= 6 + 1000000000473 \cdot 10$								
$= 2^3 \cdot 13 \cdot 96153846199$								
$n_g = 6 + 100000000005810$	1000000000581	-77	-35	-5 --------	+5	+11	+47	124
$= 6 + 10000000000581 \cdot 10$								
$= 2^3 \cdot 11 \cdot 31 \cdot 107 \cdot 342587771$								
Folge 2								
$n_g = 8 + 100000000001410$	10000000000141	-37	-31	-5 --------	+5	+49	+131	168
$= 8 + 10000000000141 \cdot 10$								
$= 2 \cdot 3 \cdot 7^2 \cdot 34013605447$								
$n_g = 8 + 100000000002730$	10000000000273	-161	-119	-5	+5	+49	+83	244
$= 8 + 10000000000273 \cdot 10$								
$= 2 \cdot 3 \cdot 11 \cdot 19913 \cdot 76088561$								
Folge 3								
$n_g = 12 + 99999999999030$	9999999999903	-25	-5 --------	-1 --------	+5	+37	+59	84
$= 12 + 9999999999903 \cdot 10$								
$= 2 \cdot 3 \cdot 11 \cdot 1515151515137$								
$n_g = 12 + 100000000002030$	10000000000203	-83	-13	-5 --------	+5	+37	+71	154
$= 12 + 10000000000203 \cdot 10$								
$= 2 \cdot 3^2 \cdot 7 \cdot 727 \cdot 1091679221$								
Folge 4								
$n_g = 14 + 99999999999190$	9999999999919	-23	-17	-5	+5	+19	+65	88
$= 14 + 9999999999919 \cdot 10$								
$= 2^2 \cdot 3 \cdot 8333333333267$								
$n_g = 14 + 100000000003030$	10000000000303	-55	-7	-5 --------	+5	+29	+43	98
$= 14 + 10000000000303 \cdot 10$								
$= 2^2 \cdot 3 \cdot 23 \cdot 89 \cdot 407099821$								

Anmerkungen und Schlussfolgerungen:

1. Aus **Tab. 23** kann entnommen werden, dass Zwillinge mod 10 in großen Zahlen-Intervallen relativ häufig (und nur in den vier Folgen) auftreten. Im Zahlen-Intervall: n:= 99999999999042 bis 100000000005816 treten (insgesamt) acht Zwillinge mod 10 auf. Die Primzahl-Dichten, die in der Umgebung der geraden Zahlen (n_g) der vier Folgen auftreten, wurden in **Tab. 35** ermittelt, mitsamt. den Primzahl - Dichten in ihren Umgebungen auftreten.

Die Komposition des Zwillings mod 10 mit dem (direkt nachfolgenden) Zwilling mod 6 ergibt einen Zwilling mod 16, der von zwei Primzahlen der gleichen Obergruppe II abgeschlossen wird, was der Leser selbst (leicht) nachprüfen kann.

4. Von den acht Zwillingen mod 10, die in dem Zahlen-Intervall $\Delta = 6774$ auftreten, besteht der Zwilling mod 10: (**99999999999037**, 99999999999043) --- (99999999999043, 99999999999047) aus einem Zwilling mod 6, der mit einem Zwilling mod 4 (direkt) verkettet ist. Diese beiden Zwillinge, die einen (zusammengesetzten) Zwilling mod 10 bilden, wurden in **Tab. 35** durch gestrichelte Linien verbunden und (obendrein) unterstrichen.

5. Alle Zwillinge mod 10 sind stets von Primzahlen abgeschlossen, die (paarweise) zu unterschiedlichen Obergruppen: I, II gehören. Der Leser kann diesen Sachverhalt durch die Division dieser Primzahlen durch 4 nachvollziehen, wobei eine der Primzahlen den Rest 1 (R1) und die zweite den Rest 3 (R3) aufweist. Es genügt diesen (invarianten) Sachverhalt für den Zwilling mod 10; (31, **41**) zu testen. Hier hat die Primzahl $p_1 = 31$ den Rest 3 und die Primzahl $p_2 = \mathbf{41}$ den Rest 1. Die Invarianz dieses Sachverhalts kann auch aus **Anlage 2** entnommen werden, wo die Primzahlen der Obergruppe I fett gedruckt und die Primzahlen der Obergruppe II (zwecks Unterscheidung) nicht fett gedruckt wurden.

6. Der Leser kann sich davon (leicht) überzeugen, dass die Primzahlen, der Zwillinge mod 20, mod 30, ... allgemein der Zwillinge mod 10n, $n \geq 1$ von Primzahlen abgeschlossen werden, die den verschiedenen Obergruppen: I und II angehören.

7. Aus **Anlage 2** kann entnommen werden, dass Zwillinge mod 4n , $n \geq 1$ stets von Primzahlen der gleichen Obergruppe abgeschlossen werden, wogegen alle Zwillinge mod 2n, n:= 1, 3, 5, ... stets von Primzahlen abgeschlossen werden, die unterschiedlichen Obergruppen angehören.
Diese Gesetzmäßigkeiten wurden in der bisherigen Primzahl-Forschung noch nicht erkannt.

8. Aus der Analyse der Abstände (Δ_1 bis Δ_6) in denen die Primzahlen in den Abständen um die jeweiligen geraden Zahlen (n_g) der vier Folgen auftreten, erkennt man die Gesetzmäßigkeit, dass die Werte dieser

Abstände stets von den Primzahlen abweichen, aus denen die geraden Zahlen (n_g) fakultativ gebildet sind. In diesen "Fakultäts-Primzahlen" kommt in allen acht geraden Zahlen die Primzahl $p = 5$ nicht vor. Dies ist offensichtlich der tiefere Grund, dass im Umfeld dieser geraden Zahlen in den Abständen: $\Delta_3 = -5$ und $\Delta_4 = +5$ die Primzahlen auftreten, die den Zwilling mod 10 bilden.

Aufbauend auf dieser "Struktur-Analyse" kann der Beweis erbracht werden, dass alle Zwillinge der Modalitäten: mod 2n, $n \geq 1$ in einer unendlichen Zahl auftreten. Für alle Zwillinge mod 2n, $n \geq 1$ können entsprechende (unendliche) Folgen definiert werden, in denen die (jeweiligen) Zwillinge in einer unendlichen Zahl auftreten.
Anhand von **Tab. 24** werden die Folgen für die Zwillinge mod 12, mod 14, mod 16, mod 18, mod 20 und mod 100 deduziert und der (interessierte) Leser kann diese Folge-Gesetze für Zwillinge beliebiger Modularität aufbauend auf der hier demonstrierten Methodik selbst entwickeln

Tab. 24 - Die Entwicklung der Folgen in denen Zwillinge mod 12, mos 14, mod 16, mod 18, mod 20 und mod 100 in unbeschränkter Zahl auftreten können:

Zwillinge mod 12: Ableitung der aktiven Folgen:

(1, **2**, **3**, 4, **5**, 6, **7**, 8, 9, 10, **11**, 12, **13**), - *Folge 1:=* 7 + 10n, n = 0
(**11**, 12, **13**, 14, 15, 16, **17**, 18, **19**, 20, 21, 22, **23**), + *Folge 1:=* 7 + 10n, n = 1
(**211**, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, **223**), + *Folge 1:=* 7 + 10n,.n = 21

(**3**, 4, **5**, 6, **7**, 8, 9, 10, **11**, 12, **13**, 14, 15), - In dieser *Folge* sind Zwillinge mod 12 unmöglich, da hier die ungerade nicht prime Rand-Zahlen EZ5 auftritt.

(**5**, 6, **7**, 8, 9, 10, **11**, 12, **13**, 14, 15, 16, **17**), + Ein singulärer Zwilling mod 12.

(**3**, 4, **5**, 6, **7**, 8, 9, 10, **11**, 12, **13**, 14, 15), - In dieser *Folge* sind Zwillinge mod 12 unmöglich, da hier die ungerade nicht prime Rand-Zahlen EZ5 auftritt.

(**5**, 6, **7**, 8, 9, 10, **11**, 12, **13**, 14, 15, 16, **17**), + Ein singulärer Zwilling mod 12.

Tab. 24 - Fortsetzung

(15, 16, **17**, 18, **19**, 20, 21, 22, **23**, 24, 25, 26, 27), - In dieser *Folge* können keine weiteren Zwillinge mod 12 wegen der nicht primen Rand-Zahlen, EZ5, vorkommen.

(**7**, 8, 9, 10, **11**, 12, **13**, 14, 15, 16, **17**, 18, **19**), + *Folge 2:*= 13 + 10n, n = 0
(467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479), + *Folge 2:*= 13 + 10n, n = 46

(9, 10, **11**, 12, **13**, 14, 15, 16, **17**, 18, **19**, 20, 21), - *Folge 3:*= 15 + 10n, n = 0
(**19**, 20, 21, 22, **23**, 24, 25, 26, 27, 28, **29**, 30, **31**), + *Folge 3:*= 15 + 10n, n = 1
(**59**, 69, **61**, 62, 63, 64, 65, 66, **67**, 68, 69, 70, **71**), + *Folge* 3:= 15 + 10n, n = 4
(**509**, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, **521**), + *Folge 3:*= 15 + 10n, n = 50

Anmerkungen und Schlussfolgerungen:

1. Zwillinge mod 12 können in *Folge 1, Folge 2,* und in *Folge 3* in unbegrenzter Zahl auftreten.

2. Zwillinge mod 12 können, wie ersichtlich, aus Zwillingen mod 2, mod 4, mod 6 kombiniert sein oder als "reine" Zwillinge mod 12 auftreten - siehe die Zwillinge mod 12: (211, 223) in *Folge 1*, (467, 479) in *Folge 2* und **(509, 521)** *in Folge 3.*

3. Beachtlich ist, dass beide Primzahlen, .die Zwillinge mod 12 abschließen, stets Elemente der gleichen Obergruppe I bzw. II sein müssen - siehe auch **Anlage 2**.

Zwillinge mod 14:

(1, 2, **3**, 4, **5**, 6, **7**, 8, 9, 10, **11**, 12, **13**, 14, 15), - Diese Folge enthält keine Zwillinge mod 14, da die zweite Rand-Zahl nicht prim sein kann, EZ5

(**3**, 4, **5**, 6, **7**, 8, 9, 10, **11**, 12, **13**, 14, 15, 16, **17**), + *Folge 1:*= 10 + 10n, n = 0
(13, 14, 15, 16, 17, 18, 18, 20, 21, 22, 23, 24, 25, 26, 27), - *Folge 1:*= 10 + 10n, n = 1

(**23**, 24, 25, 26, 27, 28, **29**, 30, **31**, 32, 33, 34, 35, 36, **37**), + *Folge 1:*= 10 + 10n, n = 2
(**113**, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, **127**), + *Folge 1:*= 10 + 10n, n = 11

(**5**, 6, **7**, 8, 9, 10, **11**, 12, **13**, 14, 15, 16, **17**, 18, 19, 20, 21), - Diese Folge enthält keine Zwillinge mod 14, da die erste Rand-Zahl nicht prim sein kann, EZ5

(**7**, 8, 9, 10, **11**, 12, **13**, 14, 15, 16, **17**, 18, **19**, 20, 21), - *Folge 2:*= 14 + 10n, n = 0
(**17**, 18, **19**, 20, 21, 22, **23**, 24, 25, 26, 27, 28, **29**, 30, **31**), + *Folge 2:*= 14 + 10n, n = 1
(**317**, 316, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, **331**), + *Folge 2:*= 14 + 10n, n = 31

(9, 10, **11**, 12, **13**, 14, 15, 16, **17**, 18, **19**, 20, 21, 22, **23**), - *Folge 3:*= 16 + 10n, n = 0
(**19**, 20, 21, 22, **23**, 24, 25, 26, 27, 28, **29**, 30, **31**, 32, 33), - *Folge 3:*= 16 + 10n, n = 1
(**29**, 30, **31**, 32, 33, 34, 35, 36, **37**, 38, 39, 40, **41**, 42, **43**), + *Folge 3:*= 16 + 10n, n = 2
(**839**, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, **853**), + *Folge 3:*= 16 + 10n, n = 83

Anmerkungen und Schlussfolgerungen:

1. Zwillinge mod 14 können *in Folge 1, Folge 2* und *in Folge* 3 in unbegrenzter Zahl auftreten.

2. Zwillinge mod 14 können, wie ersichtlich, aus Zwillingen mod 2, mod 4, mod 6 kombiniert sein oder als "reine" Zwillinge mod 14 auftreten - siehe die Zwillinge mod 14: **(113**, 127) in *Folge 1*, **(317**. 331), in *Folge 2* und (839, **853)** in *Folge 3.*

Zwillinge mod 16:

(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17), - *Folge 1:=* 9 + 10n, n = 0
(11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27), - *Folge 1:=* 9 + 10n, n = 1
(21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 35, 37), - *Folge 1:=* 9 + 10n, n = 2
(31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47), + *Folge 1:=* 9 + 10n, n = 3
(2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237), + *Folge 1:=* 9 + 10n, n = 222

(3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19), + *Folge 2:=* 11 + 10n, n = 0
(13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29), + *Folge 2:=* 11 + 10n, n = 1
(23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39), - *Folge 2:=* 11 + 10n, n = 2
(2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229), + *Folge 2:=* 11 + 10n, n = 221

(5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21), - In dieser Zahlenfolge: 13 + 10n kommen Zwillinge mod 16 nicht vor, da die linke Rand-Zahl die EZ5. hat.

(7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23), + *Folge 3:=* 15 + 10n, n = 0
(17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33), - *Folge 3:=* 15 + 10n, n = 1
(37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53), + *Folge 3:=* 15 + 10n, n = 3
(6637, 6638, 6639, 6640, 6641, 6642, 6643, 6644, 6645, 6646, 6647, 6648, 6649, 6650, 6651, 6653), + *Folge 3:=* 15 + 10n, n = 663

(9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25), - In dieser Zahlenfolge: 17 + 10n kommen Zwillinge mod 16 nicht vor, da die rechte Rand-Zahl mit der Endziffer EZ5 nicht prim ist

Anmerkungen und Schlussfolgerungen:

1. Zwillinge mod 16 kommen (wie ersichtlich) in unbegrenzter Zahl *in Folge 1, Folge* 2 *und Folge* 3 vor.

2. In kleinen Zahlen-Intervallen treten die Zwillinge mod 16 in Verkettungen aus Zwillingen mod 2, mos 4 und mod 6 auf, wogegen in größeren Zahlenbereichen überwiegend "reine" Zwillinge mod 16 auftreten,.

3. Beide "Rand-Primzahlen" der Zwillinge mod 16 sind stets Elemente der gleichen Obergruppe - hier zwei Beispiele:
   Beispiel 1: Im Zwilling mod 16: (31, 47) sind beide Primzahlen Elemente der Obergruppe II, Rest 3.
   Beispiel 2: Im Zwilling mod 16: (13, 29) sind beide Primzahlen Elemente der Obergruppe I, Rest 1.

Zwillinge mod 18:

(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19), - *Folge 1:=* 10 + 10n, n = 0
(11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29), *Folge 1:=* 10 + 10n, n = 1
(41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59), + *Folge 1:=* 10 + 10n, n = 4
(71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89), + *Folge 1;=* 10 + 10n, n = 7
(2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179), + *Folge 1:=* 10 + 10n, n = 216

(3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21), - *Folge 2:=* 12 + 10n, n = 0
(13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31), + *Folge 2:=* 12 + 10n. n = 1
(1913, 1914, 1915, 1916, 1917, 1918, 1919, 1920, 1921, 1922, 1923, 1924, 1925, 1926, 1927, 1928, 1929, 1930, 1931), + *Folge 2:=* 12 + 10n, n = 191

(5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23), + Das einzige Element der Folge 14 + 10n, in allen weiteren Elementen dieser Folge können keine Zwillinge mod 18 auftreten, da die linke Rand-Zahl mit der Endziffer, EZ5, nicht prim ist.

(5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23), + Das einzige Element der Folge 14 + 10n, in allen weiteren Elementen dieser Folge können keine Zwillinge mod 18 auftreten, da die linke Rand-Zahl mit der Endziffer, EZ5, nicht prim ist.

(7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25), - In der Folge 18 + 10n können keine Zwilling mod 18 auftreten, da die rechte Rand-Zahl mit der Endziffer, EZ5, nicht prim ist.

(9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27), - *Folge 3*:= 18 + 10n, n = 0
(19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37), + *Folge 3*:= 18 + 10n, n = 1
(29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47), + *Folge 3*:= 18 + 10n, n = 2
(1759, 1760, 1761, 1762, 1763, 1764, 1765, 1766, 1767, 1768, 1769, 1770, 1771, 1772, 1773, 1774, 1775, 1776, 1777), + *Folge 3*:= 18 + 10n. n = 175

Anmerkungen und Schlussfolgerungen:

1. Es wurde demonstriert, dass Zwillinge mod 18 in unbegrenzter Zahl in *Folge 1, Folge 2* und in *Folge 3* auftreten können.

2. Zwillinge mod 18 werden in kleinen Zahlen-Intervallen überwiegend aus verketteten Zwillingen mod 10, mod 8, mod 6, mod 4 und mod 2 gebildet, wogegen Zwillinge mod 18 in größeren Zahlen-Intervallen überwiegend als "reine" Zwillinge auftreten, in die keine Zwillinge kleinerer Modularität eingelagert sind. Beispiele und die Begründung für diesen Sachverhalt werden in **Tab. 37** vorgestellt.

3. Aus Tab. 36 kann entnommen werden, dass die einen Zwilling mod 18 umschließenden Primzahlen stets Elemente der Obergruppe I und II sein müssen - hier das Beispiel für den Zwilling mod 18: (41, 59) in dem die Primzahl 41, R1 ein Element der Obergruppe I ist, wogegen die Primzahl 59, R3 ein Element der Obergruppe II ist. Diese Gesetzmäßigkeit wird von allen Zwillingen mod 18 erfüllt.

Zwillinge mod 20:

(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21), - *Folge 1*:= 11 + 10n, n = 0

(11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31), + *Folge 1*;= 11 + 10n, n = 1

(3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23), + *Folge 2*:= 13 + 10n, n = 0
(13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33), - *Folge 2*:= 13 + 10n, n = 2
(53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73), + *Folge 2*:= 13 + 10n, n = 5

(5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25), - In der Folge: 15 + 10n können keine Zwillinge mod 18 auftreten.

(7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27), - *Folge 3*:= 17 + 10n, n = 0
(17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37), + *Folge 3*:= 17 + 10n, n = 1
(887. 888,889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900. 901, 902, 903, 904, 905, 906, 907), + *Folge 3*:= 17 + 10n, n = 88

(9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29), - *Folge 4*:= 19 + 10n, n = 0
(59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79), + *Folge 4*:= 19 + 10n, n = 5
(7649, 7650, 7651, 7652, 7653, 7654, 7655, 7656, 7657, 7658, 7659, 7660, 7661, 7662, 7663, 7664, 7665, 7666, 7667, 7668, 7669), + *Folge 4*:= 19 + 10n, n = 765

Anmerkungen und Schlussfolgerungen:

1. Es ist leicht nachprüfbar, dass die Primzahlen, die Zwillinge mod 20 beidseitig umranden, stets Elemente der gleichen Obergruppe sein müssen und diese Gesetzmäßigkeit gilt für alle Zwillinge mod 20n, n ≥1 - hier zwei Beispiele: Im Zwilling mod 20 (11, 31) sind beide Primzahlen Elemente der Obergruppe II, R3 wogegen im Zwilling mod 20 (17, 37) beide Primzahlen Elemente der Obergruppe II, R1 sind.

Tab. 24 - Fortsetzung
Zwillinge mod 100:

(1, 2, **3**, 4, **5**, 6, **7**, 8, 9, 10, **11**, 12, **13**, 14, 15, 16, **17**, 18, **19**, 20, 21, 22, **23**, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42. 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, **53**, 54, 55, 56, 57, 58, **59**, 60, **61**, 62, 63, 64, 65, 66, **67**, 68, 69, 70, **71**, 72, **73**, 74, 75, 76, 78, **79**, 80, 81, 82, **83**, 84, 85, 86, 87, 88, **89**, 90, 91, 92, 93, 94, 95, 96, **97**, 98, 99. 100, **101**), - *Folge 1*:= 51 + 10n, n = 0

(**31**, 32, 33, 34, 35, 36, **37**, 38, 39, 40, **41**, 42, **43**, 44, 45, 46, **47**, 48, 49, 50, 51, 52, **53**, 54, 55, 56, 57, 58, **59**, 60, **61**, 62, 63, 64, 65, 66, **67**, 68, 69, 70, **71**, 72, **73**, 74, 75, 76, 77, 78, **79**, 80, 81, 82, **83**, 84, 85, 86, 87, 88, **89**, 90, 91, 92, 93, 94, 95, 96, **97**, 98, 99, 100, **101**, 102, **103**, 104, 105, 106, **107**, 108, **109**, 110, 111, 112, **113**, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, **127**, 128, 129, 130, **131**), + *Folge 1*:= 51 + 10n, n = 3

(**3**, 4, **5**, 6, **7**, 8, 9, 10, **11**, 12, **13**, 14, 15, 16, **17**, 18, **19**, 20, 21, 22, **23**, 24, 25, 26, 27, 28, **29**, 30, **31**, 32, 33, 34, 35, 36, **37**, 38, 39, 40, **41**, 42, **43**, 44, 45, 46, **47**, 48, 49, 50, 51, 52, **53**, 54, 55, 56, 57, 58, **59**, 60, **61**, 62, 63, 64, 65, 66, **67**, 68, 69, 70, **71**, 72, 73, 74, 75, 76, 77, 78. **79**, 80, 81, 82, **83**, 84, 85, 86, 87, 88, **89**, 90, 91, 92, 93, 94, 95, 96, **97**, 98, 99, 100, **101**, 102, **103**), + *Folge 2*:= 53 + 10n, n= 0

(5, 6, 7, ..., 55, 56, 57, ..., 101, 102, 103, 104, 105), - In der Folge: 55 + 10n können Zwillinge mod 100 nicht auftreten, da die abschließenden Zahlen nicht prim sind.

(**7**, 8, 9, 10, **11**, 12, **13**, 14, 15, 16, **17**, 18, **19**, 20, 21, 22, **23**, 24, 25, 26, 27, 28, **29**, 30, **31**, 32, 33, 34, 35, 36, **37**, 38, 39, 40, **41**, 42, **43**, 44, 45, 46, **47**, 48, 49, 50, 51, 52, **53**, 54, 55, 56, 57, 58, **59**, 60, **61**, 62, 63, 64, 65, 66, **67**, 68, 69, 70, **71**, 72, **73**, 74, 75, 76, 77, 78, 79, 80, 81, 82, **83**, 84, 85, 86, 87, 88, **89**, 90, 91, 92, 93, 94, 95, 96, **97**, 98, 99,

100, **101**, 102, **103**, 104, 105, 106, **107**), + *Folge 3*:= 57 + 10n, n = 0

(37, 38, 39, 40, **41**, 42, **43**, 44, 45, 46, **47**, 48, 49, 50, 51, 52, **53**, 54, 55, 56, 57, 58, **59**, 60, **61**, 62, 63, 64, 65, 66, **67**, 68, 69, 70, 71, 72, **73**, 74, 75, 76, 77, 78, **79**, 80, 81, 82, **83**, 84, 85, 86, 87, 88, **89**, 90, 91, 92, 93, 94, 95, 96, **97**, 98, 99, 100, **101**, 102, **103**, 104 105, 106, **107**, 108, **109**, 110, 111, 112, **113**, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, **127**, 128, 129, 130, **131**, 132, 133, 134, 135, 136, **137**), + *Folge 3*:= 57 + 10n, n = 3

(9, 10, **11** , 12, **13**, 14, 15, 16, **17**, 18, 19, 20, 21, 22, **23**, 24, 25, 26, 27, 28, **29**, 30, **31**, 32, 33, 34, 35, 26, **37**, 38, 39, 40, **41**, 42, **43**, 44, 45, 46, **47**, 48, 49, 50, 51, 52, **53**, 54, 55, 56, 57, 58, 59, 60, **61**, 62, 63, 64, 65, 66, **67**, 68, 69, 70, **71**, 72, **73**, 74, 75, 76, 77, 78, **79**, 80, 81, 82, **83**, 84, 85, 86, 87, 88, **89**, 90, 91, 92, 93, 94, 95, 96, **97**, 98, 99, 100, **101**, 102, **103**, 104, 105, 106, **107**, 108, **109**), - *Folge 4*:= 59 + 10n, n = 0

(79. 80, 81, 82, **83**, 84, 85, 86, 87, 88, **89**, 90, 91, 92, 93, 94, 95, 96, **97**, 98, 99, 100, **101**, 102, **103**, 104, 105, 106, **107**, 108, **109**, 110, 111, 112, **113**, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, **127**, 129, 130, **131**, 132, 133, 134, 135, 136, **137**, 138, **139**, 140, 141, 142, 143, 144, 145, 146, 147, 148, **149**, 150, **151**, 152, 153, 154, 155, 156, **157**, 158, 159, 160, 161, 162, **163**, 164, 165, 166, **167**, 168, 169, 170, 171, 172, **173**, 174, 175, 176, 177, 178, **179**), + *Folge 4*:= 59 + 10n, n = 7

Anmerkungen und Schlussfolgerungen:

1. In den vier unendlichen Folgen: *Folge 1* = 51 + 10n, *Folge 2* = 53 + 10n, *Folge 3* = 57 + 10n und *Folge 4* = 59 + 10n können unendlich viele Zwillinge mod 100 auftreten die (fallweise in kleinen Zahlen-Intervallen) aus der Verkettung von Zwillingen mod <100 konstituiert sind. In **Tab. 37** wird gezeigt, dass in großen Zahlen-Intervallen überwiegend "reine" Zwillinge mod 100 auftreten können.

2. Für alle Zwillinge mod 100 gilt die (invariante) Gesetzmäßigkeit, dass die Primzahlen von denen die Zwillinge mod 100 umschlossen werden, stets Elemente der gleichen Obergruppe I bzw. II sein müssen. Hier zwei Beispiele:

Intervallen) aus der Verkettung von Zwillingen mod <100 konstituiert sind. In **Tab. 37** wird gezeigt, dass in großen Zahlen-Intervallen überwiegend "reine" Zwillinge mod 100 auftreten können.

2. Für alle Zwillinge mod 100 gilt die (invariante) Gesetzmäßigkeit, dass die Primzahlen von denen die Zwillinge mod 100 umschlossen werden, stets Elemente der gleichen Obergruppe I bzw. II sein müssen. Hier zwei Beispiele:

Tab.24 - Fortsetzung

<u>Beispiel 1</u> Im Zwilling mod 100: (7, 107) sind beide Primzahlen Elemente der Obergruppe II.
<u>Beispiel 2</u>: Im Zwilling mod 100: **(37, 137)** sind beide Primzahlen Elemente der Obergruppe I.

3. Diese Gesetzmäßigkeit, wird in allen Zwillingen mod 10n, n ≥1 erfüllt - siehe auch das Beispiel der Zwillingen mod 20n.

4. Es gilt überdies das Gesetz, wonach für alle Zwillinge mod 10n, n ≥1 stets vier unendliche Folgen (immanent) existieren - *Folge 1, Folge 2, Folge* 3 *und Folge 4* in denen diese Zwillinge in einer unendlichen Zahl auftreten.

Den in Tab. 24 ermittelten Folgen der geraden bzw. ungeraden Zahlen (n_g bzw. n_u), die zentral inmitten der jeweiligen Zwillinge situiert sind, entsprechen jeweils zwei "begleitende" Dirichlet-Progressionen, in denen unendlich viele Primzahlen auftreten, wobei fallweise in beiden, derart "gekoppelten" Dirichlet- Progressionen Primzahlen auftreten.
In Tab. 25 wird demonstriert, dass die hier deduzierten Folgen, in denen Zwillinge mod 2n, n ≥1 auftreten, stets auch in großen Zahlen-Intervallen gelten, und dass diese Zwillinge stets nur innerhalb der definierten Folgen auftreten können. In Tab. 25 wurden auch die Primzahl-Dichten bestimmt. die in den Umgebungen der (jeweiligen) Zwillinge auftreten, wie auch die Primzahl-Fakultäten aus denen die (jeweiligen) Elemente der analysierten Folgen gebildet sind.

Tab. 25 - Zwillinge mod 2, und mod 12, 14, 16, 18, 20 und mod 100 in großen Zahlen-Intervallen und die Bestimmung der Primzahl-Dichten, die in der Umgebung dieser Zwillinge auftreten:

<u>Zwillinge mod 2:</u>

Folge 1: $n_g = 12 + 30n$	n:=	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$
n_g = 100000000000842 = 12 + 10000000000083 · 10 = 2 ·3 ·17 ·29 ·41 ·719289917	10000000000083	-65	-5	-1 --------	+1	+18	+121	186
Folge 2: $n_g = 18 + 30n$								
n_g = 100000000005288 = 18 + 10000000000527 · 10 = 2^3 ·3 ·277 ·34583 ·434957	10000000000527	-55	-29	-1 --------	+1	+41	+53	108
Folge 3: 30 + 30n								
n_g = 99999999999930 = 30 + 9999999999990 · 10 = 2 ·5 ·7 ·79259 ·18024091	99999999999990	-77	-7	-1 --------	+1	+29	+41	118

Tab. 25 - Fortsetzung
<u>Zwillinge mod 12:</u>

Folge 1: 7 + 10n

Tab. 25 - Fortsetzung

Zwillinge mod 12:

Folge 2: $13 + 10n$	n:=	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$
$n_u = 100000000001183$ $= 13 + 1000000000017 \cdot 10$ $= 31 \cdot 6481 \cdot 7321 \cdot 67987$	1000000000017	-54	-10	-6 --------	+6	+20	+90	144

Folge 3: $15 + 10n$								
$n_u = 10000000001675$ $= 15 + 1000000000166 \cdot 10$ $= 5 \cdot 263 \cdot 7604562739$	1000000000166	-42	-18	-6 --------	+6	+8	+78	120

Zwillinge mod 14:

Folge 1: $10 + 10n$								
$n_g = 100000000000950$ $= 10 + 10000000000094 \cdot 10$ $= 2 \cdot 5^3 \cdot 103 \cdot 139 \cdot 4656469$	10000000000094	-97	-67	-7 -------	+7	+17	+43	140

Folge 2: $14 + 10n$								
$n_g = 100000000001184$ $= 14 + 10000000000117 \cdot 10$ $= 2^4 \cdot 3 \cdot 19 \cdot 47 \cdot 49 \cdot 997 \cdot 1481$	10000000000117	-43	-23	-7 -------	+7	+23	+55	98

Folge 3: $16 + 10n$								
$n_g = 100000000009326$ $= 16 + 10000000000931 \cdot 10$ $= 2 \cdot 3 \;\; 23 \cdot 5233 \cdot 138474619$	10000000000931	-107	-49	-7 -------	+7	+37	+43	150

Zwillinge mod 16:

Folge 1: $9 + 10n$								
$n_u = 100000000001169$ $= 9 + 1000000000116 \cdot 10$ $= 3^2 \cdot 31 \cdot 71 \cdot 504821041$	1000000000116	-76	-28	-8 -------	+8	+22	+36	112

Folge 2: $11 + 10n$								
$n_u = 10000000002081$ $= 11 + 1000000000207 \cdot 10$ $= 3 \cdot 59 \cdot 56497175153$	1000000000207	-58	-8 --------	-2 -------	+8	+26	+40	54

Anmerkung:
Der Zwilling mod 16 besteht aus einem Zwilling mod 8, mod 6 und mod 2

Folge 3: $15 + 10n$	n:=	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$:=
$n_u = 10000000001625$ $= 15 + 1000000000161 \cdot 10$ $= 3 \cdot 5^3 \cdot 131^2 \cdot 421 \cdot 3691$	1000000000161	-58	-52	-8	+8	+32	+44	102

Anmerkung: Der verkettete Zwilling mod 16 in *Folge 2* wurde durch Unterstreichung markiert.

$= 15 + 1000000000161 \cdot 10$
$= 3 \cdot 5^3 \cdot 131^2 \cdot 421 \cdot 3691$

Anmerkung: Der verkettete Zwilling mod 16 in *Folge 2* wurde durch Unterstreichung markiert.

Tab. 25 - Fortsetzung

Zwillinge mod 18:

Folge 1: 10 + 10n	n:=	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$
n_g = 10000000000600 = 10 + 1000000000059 · 10 = $2^3 \cdot 5^2 \cdot 3947 \cdot 12667849$	1000000000059	-167	-147	-9	+9	+43	+49	216
Folge 2: 12 + 10n								
n_g = 10000000001002 = 12 + 100000000099 · 10 = $2 \cdot 11 \cdot 367 \cdot 1238543473$	100000000099	-45	-15	-9	+9	+21	+85	130
Folge 3: 18 + 10n								
n_g = 10000000000768 = 18 + 1000000000075 · 10 = $2^7 \cdot 5 \cdot 1459 \cdot 1070939$	1000000000075	-39	-17	-9	+9	+85	+115	154

Zwillinge mod 20:

Folge 1: 11 + 10n								
n_u = 10000000000401 = 11 + 1000000000039 · 10 = $3 \cdot 304643 \cdot 10941769$	1000000000039	-88	-58	-10	+10	+32	+52	140
Folge 2: 13 + 10n								
n_u = 10000000000443 = 13 + 1000000000043 · 10 = $3 \cdot 19 \cdot 175438596499$	1000000000043	-55	-32	-10	+10	48	+166	218
Folge 3: 17 + 10n								
n_u = 10000000004337 = 17 + 1000000000432 · 10 = $3^4 \cdot 7 \cdot 919 \cdot 2711 \cdot 7079$	1000000000432	-110	-38	-10	+10	+50	+62	172
Folge 4: 19 + 10n								
n_u = 10000000003749 = 19 + 1000000000373 · 10 = $3 \cdot 7 \cdot 19 \cdot 25062656651$	1000000000373	-50	-32	-10	+10	+32	+94	144

Zwillinge mod 100:	n:=	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$:=

Anmerkungen und Schlussfolgerungen:

1. In **Tab. 25** liegt (in Verbindung mit den Aussagen der Tabellen: **21, 24, 26, 29, 32, 34, 35** und **36**) die Bestätigung der bisher unbewiesenen Vermutung vor, dass Zwillinge mod 2 in einer unendlichen Zahl auftreten. Darüber hinaus liegt in **Tab. 25** die Bestätigung vor, dass Zwillinge aller Modularitäten: mod 2n, $n \geq 1$ in der unendlichen Zahlenmenge ($n \to \infty$) in einer unendlichen Zahl vorkommen.

2. In **Tab. 25** finden die in dieser Monographie deduzierten *Folge-Gesetze*, die für die (jeweiligen) Zwillinge mod 2n, $n \geq 1$ gelten, ihre volle Bestätigung.

3. Aus **Tab. 25** ergibt sich, dass "reine" Zwillinge großer Modularität (wie z.B. die Zwillinge mod 100) in hohen Zahlenbereichen auftreten, wogegen in kleineren Zahlenbereichen überwiegend verkettete Zwillinge hoher Modularität auftreten. Diese Eigenschaften der Zwillinge mod 2n, $n \geq 3$ ist für den Beweis der Goldbach'schen Vermutung von besonderer Bedeutung - siehe die Monographie des Verfassers u.d.T.: " Der Beweis der Goldbach-Vermutung. ... ". Diese Eigenschaft der Zwillinge mod 2n, $n \geq 3$ findet auch auch in **Anlage 2** eine volle Bestätigung.

4. Die einzigen Zwillinge mod 2n, die nur als "reine, nicht verkettete" Zwillinge auftreten können, sind die Zwillinge mod 2 und mod 4.

5. Die relativ hohen Primzahl-Dichten, die in den Umgebungen der inmitten von Zwillingen mod 2n zentral situierten geraden bzw. ungeraden Zahlen (n_g bzw. n_u) auftreten sind durch die "Primzahl-Genese" dieser Zahlen bedingt. Wie aus den vorgestellten Zerlegungen dieser Zahlen in ihre Primzahl-Fakultäten hervorgeht, bestehen diese Zahlen aus sehr lückenhaften Primzahl-Produkten, so dass in den Umgebungen dieser Zahlen Primzahlen in primen Abständen auftreten können, die in den Primzahl-Fakultäten dieser Zahlen nicht vorkommen.

6. Die Summen-Abstände: $\Sigma\Delta$, die in **Tab. 25** ermittelt wurden, definieren diverse verkettete Zwillinge, die (wie ersichtlich) auch in großen Zahlen-Intervallen auftreten können (müssen).

7. In **Tab. 25** wurden die jeweils auftretenden Zwillinge mod: 12, 14, 16, 18, 20 und 100 durch gestrichelte Linien markiert.

8. Die in **Tab. 25** angeführten Zwillinge unterschiedlicher Modularität bestätigten Gesetze der *Folgen*, in denen die Zwillinge mod 2n, $n \geq 1$ auftreten können werden schließlich im nachfolgenden Kapitel systematisch angeführt.

----

8. Die Bildungsgesetze der Zwillinge mod 2n, n ≥1:

Der Leser wird auf die **Anlagen 2** hingewiesen, in denen die Zwillinge mod 2, mod 4, mod 6, ..., bis mod 30 und mod 100 ausgewiesen wurden, die in unterschiedlichen Zahlen-Intervallen auftreten und des weiteren auf die in Kapitel 9 deduzierten *Folge-Gesetze*, die in **Tab. 38** zusammengefasst sind.

Die Tab. 38 vorgestellten Bildungsgesetze der unendlichen *Folgen* basieren auf den geraden bzw. ungeraden Zahlen (n_g bzw. n_u), die zentral inmitten der (jeweiligen) Zwillinge mod 2n, $n \geq 1$ situiert sind. Diese *Folgen* der Zahlen n_g bzw. n_u sind jeweils von zwei Dirichlet-Progressionen flankiert, in denen unendlich viele Primzahlen auftreten, die fallweise in den Abständen $\Delta = 2n$, $n \geq 1$ gleichzeitig auftreten und damit die Zwillinge mod 2n, $n \geq 1$ bilden. Dieser (invariante) Sachverhalt wird nachfolgend am Beispiel der *Folge 1*: 12 + 30n der Zwillinge mod 2 demonstriert:

D-Progression 1:= 11 + 30n:=	**11**, **41**, **71**, **101**, **131**, 161, **191**, 221, **251**, **281**, ..., **3371**, ...
Folge 1 = 12 + 30n =	12, 42, 72, 102, 132, 162, 192, 222, 252, 282, ..., 3372,....
D-Progression 2 = 13 + 30n:=	**13**, **43**, **73**, **103**, 133, **163**, **193**, **223**, 253, **283**, ..., **3373**, ...

D-Progression 1:= 11 + 30n:= **11**, **41**, **71**, **101**, **131**, 161, **191**, 221, **251**, **281**, ..., **3371**, ...
Folge 1 = 12 + 30n = 12, 42, 72, 102, 132, 162, 192, 222, 252, 282, ..., 3372,....
D-Progression 2 = 13 + 30n:= **13**, **43**, **73**, **103**, 133, **163**, **193**, **223**, 253, **283**, ..., **3373**, ...

In den ersten zehn Elementen der *Folge 1* := 12 + 30n treten die sechs unterstrichenen geraden Zahlen mit der EZ2 auf, die innerhalb von Zwillingen mod 2 (EZ1, EZ3) situiert sind.
Daraus ist ersichtlich, dass man die sechs (paarweise gekoppelten) Dirichlet-Progressionen, in denen jeweils unendlich viele (paarweise im Abstand Δ=2 gekoppelte) Primzahlen auftreten, durch die drei Folgen: *Folge 1* = 12 + 30n, *Folge 2* = 18 + 30n, *Folge 3* = 30 + 30n der geraden Zahlen mit den Endziffern: EZ2, EZ8, EZ0 substituieren kann.
Nach der gleichen Substitution wurde die Zahl der Dirichlet-Progressionen, in denen (paarweise und in einer unendlichen Folge) die Primzahlen auftreten, die Zwillinge mod 2n, n ≥1 abschließen, durch die halbe Zahl der Folgen der geraden bzw. ungeraden Zahlen (n_g bzw. n_u), die zentral inmitten der jeweiligen Zwillinge situiert sind, substituiert. Nach diesen Erläuterungen sind die (invarianten) *Folgen-Gesetze*, von **Tab. 26**, die für die Zwillinge mod 2n, n ≥1 gelten, leicht verständlich.

Tab. 26 - Die Bildungs-Gesetze der *Folgen* in denen Zwillinge mod 2n, n ≥1 auftreten können:

Zwillinge mod 2n, n ≥1:	*Folge 1*	*Folge 2*	*Folge 3*	*Folge 4*
Zwillinge mod 2:	12 + 30n	18 + 30n	30 + 30n	--------
Zwillinge mod 4:	5 + 10n	9 + 10n	11 + 10n	.---------
Zwillinge mod 6:	14 + 10n	16 + 10n	10 + 10n	---------
Zwillinge mod 8:	7 + 10n	13 + 10n	15 + 10n	---------
Zwillinge mod 10:	6 + 10n	8 + 10n	12 + 10n	14 + 10n
Zwillinge mod 12:	7 + 10n	13 + 10n	15 + 10n	---------
Zwillinge mod 14:	10 + 10n	14 + 10n	16 + 10n	---------
Zwillinge mod 16:	9 + 10n	11 + 10n	15 + 10n	---------
Zwillinge mod 18:	10 + 10n	12 + 10n	18 + 10n	--------
Zwillinge mod 20:	11 + 10n	13 + 10n	17 + 10n	19 + 10n
Zwillinge mod 100:	51 + 10n	53 + 10n	57 + 10n	59 + 10n

Anmerkungen und Schlussfolgerungen:

1. Die Bildungsgesetze für die *Folgen* aller weiteren Zwillinge mod 2n, n ≥1 können von interessierten Lesern auf der Basis der in dieser Monographie entwickelten Grundlagen (mühelos) vervollständigt werden mit dem Test ihrer Gültigkeit in beliebig hohen Zahlen-Intervallen durch PC-Test-Proben.

2. Aus der Übersicht von **Tab. 26** geht hervor, dass für alle Zwillinge mod 10n, n ≥1 jeweils vier Folgen: *Folge 1, Folge 2, Folge 3* und *Folge 4* auftreten, in denen unendlich viele Zwillinge mod 10n, ≥1 auftreten. Für alle Zwillinge mod 2n, die nicht vom Typ: mod 10n, n ≥1 sind, gelten jeweils nur die drei Folgen: *Folge 1, Folge 2, Folge 3*, was aus **Tab. 26** ebenfalls entnommen werden kann.

3. Aus **Tab. 26** ist des weiteren zu entnehmen, dass in den Bildungsgesetzen der drei *Folgen* aller Zwillinge mod 4n, n ≥1 ungerade (konstante) Glieder auftreten, wogegen in den drei *Folgen* aller Zwillinge mod 2 + 4n, n ≥1 stets gerade (konstante) Glieder auftreten. Die tieferen Gründe, für das Bestehen dieser Gesetzmäßigkeiten wurden bereits ausführlich erläutert (unter Beweis gestellt).

Die besonderen und invarianten Gesetze, die in allen Zwillingen mod 2n, $n \geq 1$ gelten, führen zu weiteren Erkenntnissen über das Auftreten von Mersenne- und Fermat-Primzahlen (M_p und F_p) innerhalb von Zwillingen mod 2n, $n \geq 1$. Dieser Thematik ist das nachfolgende Kapitel gewidmet.

9. Neue Aspekte der Mersenne- und Fermat-Primzahlen, M_p und F_p:

Zunächst wird die bereits in den Vorworten angesprochene Klassenteilung der geraden Zahlen $n_g = 2^n$ näher untersucht, mit der Zielsetzung einer Revision, der von Mersenne und Fermat eingeführte Restriktionen, nach denen die Primzahlen $M_p = 2^p - 1$ und $F_p = 2^{2^n} + 1$ nur innerhalb dieser Klassen der Zahlen $n_g = 2^n$ auftreten können.

Zunächst ist hervorzuheben, dass in den Umgebungen der Zahlen $n_g = 2^n$ stets Primzahlen in den Abständen $\Delta := \pm 1, \pm 3, \pm 5, \ldots, \pm n_u$ auftreten können, wobei Primzahlen vom Typ M_p bzw. F_p nur dann auftreten, wenn die Zahlen $n_g = 2^n$ extrem asymmetrisch zwischen zwei aufeinander folgenden Primzahlen (p_1, $p_2 > p_1$) situiert sind. Es soll deshalb eingangs untersucht werden in welchen der Folgen: 2^{1+4n}, 2^{2+4n}, 2^{3+4n}, 2^{4+4n} Primzahlen vom Typ M_p und F_p prinzipiell ausgeschlossen sind.

Diese Aufgabe lässt sich anhand von **Tab. 27** einsichtig lösen. Die anhand von Tab. 14 gewonnenen Einsichten werden anschließend kommentiert und führen zum Beweis, dass es unendlich viele Primzahlen der Typen: $M_p = 2^n - 1$ und $F_p = 2^n + 1$ gibt.

Tab. 27 - Die Gliederung der unendlichen Folge $n_g = 2^n$, $n \geq 1$ in vier gleichmächtige Teilfolgen:

Teilfolge 1 mit EZ2:	$2^1 = 2$, $2^{1+4} = 32$, $2^{1+8} = 512$, $2^{1+12} = 8192$, ... 2^{1+4n}, n:= 0, 1, 2, 3, ..., n
Teilfolge 2 mit EZ4:	$2^2 = 4$, $2^{2+4} = 64$, $2^{2+8} = 1024$, $2^{2+12} = 16384$, ... 2^{2+4n}, n = 0, 1, 2, 3,..., n
Teilfolge 3 mit EZ8:	$2^3 = 8$, $2^{3+4} = 128$, $2^{3+8} = 2048$, $2^{3+12} = 32768$, ... 2^{3+4n}, n = 0, 1, 2 ,3, ..., n
Teilfolge 4 mit EZ6	$2^4 = 16$, $2^{4+4} = 256$, $2^{4+8} = 4096$, $2^{4+12} = 65536$, ... 2^{4+4n}, n = 0, 1, 2, 3, ..., n

Teilfolge 1 $n_g + 1 = 2^{1+4n} + 1 :=$	Teilfolge 2 $n_g + 1 = 2^{2+4n} + 1 :=$	Teilfolge 3 $n_g + 1 = 2^{3+4n} + 1 :=$	Teilfolge 4 $n_g + 1 = 2^{4+4n} + 1 :=$
n = 0, $2^1 + 1 = \mathbf{3}$	$2^2 + 1 = \mathbf{5}$	$2^3 + 1 = \underline{9}$	$2^4 + 1 = \mathbf{17}$
n = 1, $2^5 + 1 = \underline{33}$	$2^6 + 1 = 65$	$2^7 + 1 = \underline{129}$	$2^8 + 1 = \mathbf{257}$
n = 2, $2^9 + 1 = \underline{513}$	$2^{10} + 1 = 1025$	$2^{11} + 1 = \underline{2049}$	$2^{12} + 1 = 4097$
n = 3, $2^{13} + 1 = \underline{8193}$	$2^{14} + 1 = 16385$	$2^{15} + 1 = \underline{32769}$	$2^{16} + 1 = \mathbf{65537}$
n = 4, $2^{17} + 1 = \underline{131073}$	$2^{18} + 1 = 262145$	$2^{19} + 1 = \underline{524289}$	$2^{24} + 1 = 16777217$
n = 5 $2^{21} + 1 = \underline{2097153}$	$2^{22} + 1 = 4194305$	$2^{23} + 1 = \underline{8388609}$	$2^{28} + 1 = 268435457$
n = 6 $2^{25} + 1 = \underline{33554433}$	$2^{26} + 1 = 67108865$	$2^{27} + 1 = \underline{134217729}$	$2^{32} + 1 = 4294967297$

Anmerkungen und Schlussfolgerungen:

1. Die Teilfolgen: 1, 2 und 3 bilden bezüglich der Fermat-Primzahlen inaktive Dirichlet-Progressionen. In der Teilfolge 1 ist die Fermat-Primzahlen: p = **3** die einzig mögliche Primzahl mit der Endziffer EZ3, da alle ungeraden Zahlen der Folge $2^{1+4n} + 1$ durch die Zahl 3 teilbar sind; diese Zahlen wurden deshalb durch Unterstreichung markiert.

2. In der Teilfolge 2 ist nur p = **5** eine Primzahl, alle weiteren ungeraden Zahlen dieser Teilfolge sind durch 5 teilbar, da diese Zahlen die Endziffer EZ5 aufweisen..

3. In der Teilfolge 3 sind alle ungeraden Zahlen $n_g + 1$ ebenfalls durch drei teilbar und wurden durch Unterstreichung markiert. Die Teilfolge 3 bildet eine inaktive Dirichlet-Progression.

2. In der Teilfolge 2 ist nur $p = 5$ eine Primzahl, alle weiteren ungeraden Zahlen dieser Teilfolge sind durch 5 teilbar, da diese Zahlen die Endziffer EZ5 aufweisen..

3. In der Teilfolge 3 sind alle ungeraden Zahlen $n_g +1$ ebenfalls durch drei teilbar und wurden durch Unterstreichung markiert. Die Teilfolge 3 bildet eine inaktive Dirichlet-Progression.

4. Die Teilfolge 4 bildet bezüglich der Fermat-Primzahlen die einzige aktive Dirichlet-Progression, in der unendlich viele Fermat-Primzahlen mit der Endziffer EZ7 auftreten können. Diese bisher bekannten Fermat-Primzahlen wurden in **Tab. 27** fett gedruckt markiert..

3. Für die Teilfolge 4 gilt der wichtige Hinweis, dass in dieser Teilfolge nicht nur (wie Fermat vermutete) Primzahlen vom Typ: $2^{2^n}+1$ auftreten können, vielmehr aber (fallweise) Primzahlen vom Typ: $2^{4+4n}+1$. Die Suche nach diesen Primzahlen in dieser aktiven Dirichlet-Progression - in der Teilfolge 4 sollte deshalb mit geeigneten Test-Programmen intensiviert werden, wobei die von Fermat vorgegebene Restriktion: $F_p = 2^{2^n}+1$ aufzugeben ist.

Tab. 27 - Fortsetzung

Teilfolge 1 $n_g -1 = 2^{1+4n} -1:=$	Teilfolge 2 $n_g -1 = 2^{2+4n} -1:=$	Teilfolge 3 $n_g -1 = 2^{3+4n} -1=$	Teilfolge 4 $n_g -1 = 2^{4+4n} -1=$
$n = 0$, $2^1 -1 = 1$	$2^2 -1 =$ **3**	$2^3 -1 =$ **7**	$2^4 -1 =$ <u>15</u>
$n = 1$, $2^5 -1 =$ **31**	$2^6 -1 =$ <u>63</u>	$2^7 -1 =$ **127**	$2^8 -1 =$ <u>255</u>
$n = 2$, $2^9 -1 = 511$	$2^{10} -1 =$ <u>1023</u>	$2^{11} -1 = 2047$	$2^{12} -1 =$ <u>4095</u>
$n = 3$, $2^{13} -1 =$ **8191**	$2^{14} -1 =$ <u>16383</u>	$2^{15} -1 = 32767$	$2^{16} -1 =$ <u>65535</u>
$n = 4$, $2^{17} -1 =$ **131071**	$2^{18} -1 =$ <u>262143</u>	$2^{19} -1 =$ **524287**	$2^{20} -1 =$ <u>1048575</u>

<u>Anmerkungen und Schlussfolgerungen:</u>

1. Die Teilfolgen 2 und 4 bilden bezüglich der Mersenne-Primzahlen inaktive Dirichlet-Progressionen, hingegen bilden die Teilfolgen 1 und 3 aktive Dirichlet-Progressionen, in denen unendlich viele Mersenne-Primzahlen auftreten können. Die primen Zahlen wurden hier fett gedruckt hervorgehoben.. Die Mersenne-Primzahlen können wie ersichtlich in den Endziffern: EZ1 und EZ7 auftreten.

2. Bei der Suche nach weiteren Mersenne-Primzahlen sind deshalb auch die Zahlen: $n_g -1 = 2^{1+4n} -1$ und $n_g -1 = 2^{3+4n} -1$ einzubeziehen, deren Exponenten nicht prime Zahlen sind.

3. Die von Mersenne vorgegebene Restriktion: $M_p = 2^p -1$ ist deshalb aufzugeben.

<u>Die Schlussfolgerung:</u>

In **Tab. 27** wurde nachgewiesenen, dass Mersenne-Primzahlen in zwei Dirichlet-Progressionen: vom Typ: $M_p = 2^{1+4n} -1$ und $M_p = 2^{3+4n} -1$ vorkommen und somit in einer unendlichen Zahl auftreten. Wiewohl die Fermat-Primzahlen nur in der Dirichlet-Progression vom Typ: $F_p = 2^{4+4n} +1$ vorkommen, treten diese auch diese Primzahlen (F_p) in einer unendlichen Zahl auf, was bisher nicht bewiesen werden konnte. Nachdem dieser Beweis erbracht werden konnte, empfiehlt es sich die "GIMPs-Teste" für die Suche nach weiteren Mersenne-Primzahlen zu erweitern, um auch alle Zahlen vom Typ: $2^{1+4n} -1$ und $2^{3+4n} -1$ auf ihre Primzahl-Eigenschaften testen zu können. Des weiteren gilt die Empfehlung die GIMPs-Teste so zu erweitern, dass auch alle Zahlen vom Typ: $2^{4+4n} +1$ auf ihre Primzahl - Eigenschaft getestet werden können und nicht nur die Zahlen der Untermenge: $2^{2^n}+1$.

Danach gilt die Empfehlung die "GIMPs-Teste" zu "GIM&FPs-Testen" zu erweitern und sich dabei von den "unzeitgemäßen" Restriktionen zu befreien, die Mersenne und Fermat eingeführt hatten.

In **Tab. 28** wurden alle (42) bisher gefundenen Mersenne-Primzahlen mit Angabe ihrer Endziffern und dem Datum ihrer Entdeckung angeführt. Die Primzahl $M_{p,8} = 2^{31} -1$ stammt von Euler (1772).

Tab. 28 - Die Auflistung der bisher gefundenen Mersenne-Primzahlen mit Angabe ihrer Endziffern:

Tab. 28 - Fortsetzung

$M_{p,5} = 2^{13}-1$	1	EZ1	1456	$M_{p,26} = 2^{23209}-1$	1	EZ1	1979
$M_{p,6} = 2^{17}-1$	1	EZ1	1588	$M_{p,27} = 2^{44497}-1$	1	EZ1	1979
$M_{p,7} = 2^{19}-1$	3	EZ7	1588	$M_{p,28} = 2^{86243}-1$	3	EZ7	1982
$M_{p,8} = 2^{31}-1$	3	EZ7	1772	$M_{p,29} = 2^{110503}-1$	3	EZ7	1988
$M_{p,9} = 2^{61}-1$	1	EZ1	1883	$M_{p,30} = 2^{132049}-1$	1	EZ1	1983
$M_{p,10} = 2^{89}-1$	1	EZ1	1911	$M_{p,31} = 2^{216091}-1$	3	EZ7	1985
$M_{p,11} = 2^{107}-1$	3	EZ7	1914	$M_{p,32} = 2^{756839}-1$	3	EZ7	1992
$M_{p,12} = 2^{127}-1$	3	EZ7	1876	$M_{p,33} = 2^{859433}-1$	1	EZ1	1994
$M_{p,13} = 2^{521}-1$	1	EZ1	1952	$M_{p,34} = 2^{1257787}-1$	3	EZ7	1996
$M_{p,14} = 2^{607}-1$	3	EZ7	1952	$M_{p,35} = 2^{1398269}-1$	1	EZ1	1996
$M_{p,15} = 2^{1279}-1$	3	EZ7	1952	$M_{p,36} = 2^{2976221}-1$	1	EZ1	1997
$M_{p,16} = 2^{2203}-1$	3	EZ7	1952	$M_{p,37} = 2^{3021377}-1$	1	EZ1	1998
$M_{p,17} = 2^{2281}-1$	1	EZ1	1952	$M_{p,38} = 2^{6972593}-1$	1	EZ1	1999
$M_{p,18} = 2^{3217}-1$	1	EZ1	1957	$M_{p,39} = 2^{13466917}-1$	1	EZ1	2001
$M_{p,19} = 2^{4253}-1$	1	EZ1	1961	$M_{p,40} = 2^{20996011}-1$	3	EZ7	2003
$M_{p,20} = 2^{4423}-1$	3	EZ7	1961	$M_{p,41} = 2^{24036583}-1$	3	EZ7	2004
$M_{p,21} = 2^{9689}-1$	1	EZ1	1963	$M_{p,42} = 2^{25964951}-1$	3	EZ7	2005

Anmerkungen und Schlussfolgerungen:

1. Die bisher gefundenen Mersenne-Primzahlen: $M_{p,1}$ bis $M_{p,42}$, die in **Tab. 28** aufgelistet wurden, sind eine Bestätigung der in **Tab. 27** abgeleiteten Dirichlet-Progressionen: $2^{1+4n}-1$ und $2^{3+4n}-1$, in denen "Mersenne-artige" Primzahlen (fallweise) auftreten. Die in **Tab. 28** angeführten Mersenne-Primzahlen bilden jedoch keine vollständige Liste (Folge) dieser Primahlen, da bisher nicht alle Zahlen dieser beiden Dirichlet-Progressionen auf ihre Primzahl-Eigenschaft getestet wurden, insbesondere sind noch die Zahlen mit ungeraden nicht primen Exponenten in das Testverfahren einzubeziehen.

2. Die Mersenne-Primzahl: $M_{p,1} = 2^2 - 1 = 3$, die aus der "inaktiven" Dirichlet-Progression: $2^{2+4n}-1$ stammt, ist die einzige Primzahl mit der Endziffer EZ3. Weitere Primzahlen mit der Endziffer EZ3 können in der (unendlichen) Folge der Mersenne-Primzahlen nicht auftreten.

3. Aus **Tab. 27** folgt wiederum, dass Primzahlen der "Fermat'schen Art" nur in der Folgen: $2^{4+4n}+1$ mit der Endziffer EZ7 auftreten können, bis auf die Primzahl $F_{p,2} = 2^2+1 = 5$, die solitär mit der Endziffer EZ5 auftritt.

4. Sobald man die "Fermat'sche Restriktion": $F_p = 2^{2^n}+1$ aufgibt und alle Exponenten der Folge 4 +4n:= 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, ... in das Testverfahren einbezieht, ist zu erwarten, dass man in sehr großen Zahlenbereichen weitere Fermat-Primzahlen entdecken wird.

5. Diese "Vorhersagen" mögen als ein Test des vorgestellten Beweises dienen, dass es unendlich viele Primzahlen vom Typ: $M_p = 2^{1+4n}-1$ und $M_p = 2^{3+4n}-1$ sowie $F_p = 2^{4+4n}+1$ gibt.

Die geraden Zahlen der Folge $n_g = 2^n$, $n \geq 2$ sind im gesamten Zahlenbereich ($n \to \infty$) offensichtlich stets symmetrisch oder asymmetrisch zwischen zwei aufeinander folgenden Primzahlen ($p_1 < n_g$) und ($p_2 > n_g$) situiert. Bei extrem asymmetrischer Lokalisierung von $n_g = 2^n$ zwischen den Primzahlen: p_1 und p_2 - sofern die Primzahl p_2 direkter Nachfolger der geraden Zahl $n_g = 2^n$ ist, d.h. wenn die Beziehungsgleichung: $n_g + 1 = p_2$ gilt, ergibt sich die Mersenne-Primzahl: $M_p = p_2 = 2^n - 1$ und sofern die gerade Zahl $n_g = 2^n$ ein direkter Vorgänger der Primzahl p_2 ist, gilt alternativ die Beziehungs-Gleichung: $p_2 - 1 = n_g$ bzw. $F_p = p_2 = 2^n + 1$.

Sofern die geraden Zahlen: $n_g = 2^n$ nicht extrem (links bzw. rechts) asymmetrisch zwischen zwei Primzahlen: $p_1 < p_2$ situiert sind, treten zwischen $n_g = 2^n$ und den Primzahlen: p_1 und p_2 definierte Abstände: $\Delta > \pm 1$ auf. In **Tab. 29** wurden die Primzahl-Dichten bestimmt, die im Umfeld der Zahlen: 2^n in der Folge: 2^2 bis 2^{56} auftreten. Aus **Tab. 29** ist ersichtlich, dass man durch PC-Teste leicht sechs Primzahlen $p_1 < p_2 < p_3 < p_3 < n_g = 2^n$ und $p_6 > p_5 > p_4 > n_g = 2^n$ bestimmen kann, die (jeweils) im Umfeld der Zahlen $n_g = 2^n$ auftreten.

Sofern die geraden_Zahlen: $n_g = 2^n$ nicht extrem (links bzw. rechts) asymmetrisch zwischen zwei Primzahlen: $p_1 < p_2$ situiert sind, treten zwischen $n_g = 2^n$ und den Primzahlen: p_1 und p_2 definierte Abstände: $\Delta > \pm 1$ auf. In **Tab. 29** wurden die Primzahl-Dichten bestimmt, die im Umfeld der Zahlen: 2^n in der Folge: 2^2 bis 2^{56} auftreten. Aus **Tab. 29** ist ersichtlich, dass man durch PC-Teste leicht sechs Primzahlen $p_1 < p_2 < p_3 < p_3 < n_g = 2^n$ und $p_6 > p_5 > p_4 > n_g = 2^n$ bestimmen kann, die (jeweils) im Umfeld der Zahlen $n_g = 2^n$ auftreten.

Tab. 29 - Die Primzahl-Dichten im Umfeld der Zahlen $n_g = 2^n$ (für n:= 2 bis 56):

2^n :=	Folge:		Δ_1 :=	Δ_2 :=	Δ_3 :=	Δ_4 .=	Δ_5 :=	Δ_6 :=	$\Sigma\Delta$:=
2^1 = 2	1		--	--	--	+1	+3	+7	
2^2 = 4	2				-1	+1	+3	+7	
2^3 = 8	3		-5	-3	-1	+3	+5	+9	14
2^4 = 16	4		-9	-5	-3	+1	+3	+7	16
2^5 = 32	1		-9	-3	-1	+5	+9	+11	20
2^6 = 64	2		-11	-5	-3	+3	+7	+9	20
2^7 = 128	3		-19	-15	-1	+3	+9	+11	30
2^8 = 256	4		-17	-15	-5	+1	+7	+13	40
2^9 = 512	1		-13	-9	-3	+9	+11	+29	42
2^{10} = 1024	2		-11	-5	-3	+7	+9	+15	36
2^{11} = 2048	3		-21	-19	-9	+5	+15	+21	42
2^{12} = 4096	4		-17	--5	-3	+3	+5	+31	48
2^{13} = 8192	1		-21	-15	-1	+17	+27	+29	50
2^{14} = 16384	2		-21	-15	-3	+27	+33	+37	58
2^{15} = 32768	3		-51	-49	-19	+3	+11	+15	66
2^{16} = 65536	4		-39	-17	-15	+1	+3	+7	46
2^{17} = 131972	1		-25	-13	-3	+29	+47	+75	100
2^{18} = 262144	2		-17	-11	-5	+3	+7	+9	26
2^{19} = 524288	3		-27	-19	-1	+21	+53	+59	86
2^{20} = 1948576	4		-23	-17	-5	+25	+27	+37	60
2^{21} = 2097152	1		-21	-19	-9	+17	+59	+71	92
2^{22} = 4194304	2		-27	-17	-3	+15	+25	+49	76
2^{23} = 8388608	3	Folge:	-27	-21	-15	+9	+11	+19	46
2^{24} = 16777216		4	-33	-17	-3	+43	+73	+75	108
2^{25} = 33554432		1	-61	-49	-39	+35	+41	+69	130
2^{26} = 61798864		2	-45	-35	-33	+7	+19	+25	70
2^{27} = 134217728		3	-111	-79	-39	+29	+45	+51	162
2^{28} = 268435456		4	-95	-89	-57	+3	+7	+37	132
2^{29} = 5368709112		1	-39	-9	-3	+9	+97	+99	138
2^{30} = 1073741824		2	-83	-41	-35	+3	+7	+9	92
2^{31} = 2147483648		3	-61	-19	-1	+11	+45	+65	126
2^{32} = 4294967296		4	-65	-17	-5	+15	+61	+75	140
2^{33} = 8589934592		1	-49	-39	-9	+17	+29	+35	84
2^{34} = 17179869184		2	-113	-77	-41	+25	+79	+85	198
2^{35} = 34359738368		3	-61	-49	-31	+53	+83	+99	168
2^{36} = 68719476736		4	-23	-17	-5	+31	+115	+117	140
2^{37} = 137438953472		1	-45	-31	-25	+9	+29	41	86
2^{38} = 274877906944		2	-107	-87	-45	+7	+13	+67	174
2^{39} = 549755813888		3	-67	-19	-7	+23	+39	+45	112
2^{40} = 1099511627776		4	-195	-167	-87	+15	+27	+55	250
2^{41} = 2199023255552		1	-55	-31	-21	+27	+65	+71	126
2^{42} = 4398046511104		2	-33	-17	-11	+15	+75	+87	120
2^{43} = 8796093022208		3	-117	-67	-57	+29	+39	+59	170
2^{44} = 17592186044416		4	-119	-117	-17	+7	+21	+27	146
2^{45} = 35184372088832		1	-83	-53	-11	+11	+61	+71	154
2^{46} = 70368744177664		2	-63	-57	-21	+15	+127	+139	202
2^{47} = 140737488355328		3	-147	-127	-115	+5	+9	+41	188
2^{48} = 281474976710656		4	-125	-63	-59	+7	+25	+57	182
2^{49} = 562949953421312		1	-123	-11	-81	+69	+191	+261	384
2^{50} = 1125899906842624		2	-51	-35	-27	+55	+99	+145	196

Anmerkungen und Schlussfolgerungen:

1 Die Generierung der **Tab. 29** erfordert beim Einsatz schneller PC-Rechner keinen großen Zeitaufwand. Bei sehr großen Exponenten ($n \to \infty$) wächst die Rechenzeit jedoch immens an, weshalb es bisher auch.

2. Aus **Tab. 28** findet man die in **Tab. 14** deduzierten Dirichlet-Progressionen, nach denen nur in den Folgen: $2^{1+4n}-1$ und $2^{3+4n}-1$ Mersenne-Primzahlen auftreten können, sowie in den Folgen: $2^{1+4n}+1$ und $2^{4+4n}+1$ Fermat-Primzahlen auftreten können, vollauf bestätigt. Die in **Tab. 16** auftretenden Primzahlen M_p und F_p wurden durch fett gedruckte Zahlen: **-1** bzw. **+1** hervorgehoben und stimmen mit den bereits bekannten Primzahlen M_p und F_p überein. .

3. Aus **Tab. 28** ist ersichtlich, dass die Zahlen $n_g = 2^n$ mit steigendem Exponenten (n) zunehmend in Zahlen-Intervallen mit geringer Primzahl-Dichte situiert sind und überwiegend asymmetrisch zwischen den zwei benachbarten Primzahlen: p_3 und p_4 lokalisiert sind.

4. Die im Umfeld der geraden Zahlen: $n_g = 2^n$ auftretenden Zwillinge mod 2 wurden durch gestrichelte Linien verbunden. Man erkennt, dass die Dichte der Zwillinge mod. 2 in wachsenden Zahlenbereichen geringer wird.

5. Durch geeignete Testverfahren wird es möglich Primzahlen vom Typ: M_p und F_p aufzufinden. die in den hier abgeleiteten drei Dirichlet-Progressionen auftreten, und nach denen bisher infolge der von Mersenne und Fermat aufgestellten Restriktionen überhaupt noch nicht gesucht (gefahndet) wurde.

6. Ein kurzer Vergleich mit den drei Folgen, in denen Zwillinge mod 2 auftreten können mit den vier Bildungsgesetzen der Zahlen: $n_g = 2^n$ bestätigt die (schon formulierte) These, dass im direkten Umfeld der Zweier-Potenzen: 2^n niemals Zwillinge mod 2 auftreten können. Hier der Nachweis:

Die *Folge 1* der Zwillinge mod 2:: $n_g = 12 + 30n$, EZ2	Die *Folge 2* der Zwillinge mod 2 $n_g = 18 + 30n$, EZ8	Die *Folge 3* der Zwillinge mod 2: $n_g = 30 + 30n$, EZ0
Die Zahlen $n_g = 2^n$, EZ2:: Die Zahlen $2^{1+4n} - 12$ sind nicht durch n = 30 teilbar! Beispiel: $2^5 - 12 = 32 - 12 = 20$	Die Zahlen $n_g = 2^n$, EZ8: Die Zahlen $2^{3+4n} - 18$ sind nicht durch n = 30 teilbar! Beispiel: $2^7 - 18 = 128 - 18 = 110$	Die Zahlen $n_g = 2^n$ können in der Endziffer EZ0 nicht auftreten.

Die Zahlen: 20 und 30, wie auch die Zahlen: 110 und 30 sind teilerfremde Zahlen. Dieser Konflikt mit den Folgen der geraden Zahlen, die in Zwillingen mod 2 situiert sind, ergibt sich auch im Hinblick auf die Folge: $n_g = 2^{4+4n}$, in der alle Zahlen in der Endziffer EZ6 auftreten, da in den Folgen der geraden Zahlen, die in Zwillingen mod 2 situiert sind, gerade Zahlen mit der Endziffer EZ6 bekanntlich nicht auftreten können - siehe die Ausführungen in **Kapitel 3** und die Bildungsgesetze in **Kapitel 9**..

Eine Zusammenfassung:

Die in diesem Kapitel erzielten Ergebnisse können wie folgt zusammen gefasst werden:

1. Es wurde der Beweis erbracht, dass die Mersenne- und Fermat-Primzahlem M_p und F_p in einer unendlichen Zahl auftreten.

2. Es wurde nachgewiesen, dass alle Mersenne-Primzahlen nur in den Endziffern: EZ1 und EZ7 auftreten können, ausgenommen die Primzahl $M_p = 2^2 - 1 = \mathbf{3}$.
Dieser Beweis findet seine Bestätigung in **Tab. 27** und **Tab. 28**, in denen die Endziffern aller bisher gefundenen Mersenne-Primzahlen angeführt wurden.

3. Es wurde unter Beweis gestellt, dass alle Fermat-Primzahlen nur in der Endziffern: EZ7 auftreten können, ausgenommen die Primzahl: $F_{p,2} = 2^2 + 1 = \mathbf{5}$. Dieser Beweis findet seine Bestätigung in den bisher gefundenen Primzahlen: $F_{p,1} = \mathbf{3}$, $F_{p,3} = \mathbf{17}$, $F_{p,4} = \mathbf{257}$ und $F_{p,5} = \mathbf{65537}$.

4. Es konnte nachgewiesen werden, dass die Primzahl **p = 5** ein Elemente der inaktiven Dirichlet-Progression: $2^{2+4n}-1$ ist, die hier nur solitär (für n= 0) auftreten kann.

3. Es wurde unter Beweis gestellt, dass alle Fermat-Primzahlen nur in der Endziffern: EZ7 auftreten können, ausgenommen die Primzahl: $F_{p,2} = 2^2 + 1 = 5$. Dieser Beweis findet seine Bestätigung in den bisher gefundenen Primzahlen: $F_{p,1} = 3$, $F_{p,3} = 17$, $F_{p,4} = 257$ und $F_{p,5} = 65537$.

4. Es konnte nachgewiesen werden, dass die Primzahl $p = 5$ ein Elemente der inaktiven Dirichlet-Progression: $2^{2+4n} - 1$ ist, die hier nur solitär (für n= 0) auftreten kann.
5. Es wurde der Nachweis erbracht, dass bis auf die Primzahlen: **(3, 5)** alle weiteren Primzahlen M_p und F_p keine Zwillinge mod 2 bilden können.

6. Es wurde die Notwendigkeit einer erweiterten Suche nach Mersenne- und Fermat-Primzahlen unter Aufgabe der von Mersenne und Fermat stammenden Restriktionen herausgestellt. Dies besagt, dass in der künftigen Suche nach weiteren Primzahlen: M_p und F_p alle Exponenten die in den drei Dirichlet-Progressionen auftreten können, herangezogen werden sollten.

---

10. Der Beweis, dass es unendlich viele Primzahlen $CM_p = n_g - 1$ und $CF_p = n_g + 1$ gibt:

Auf diese Primzahlen, die aus geraden Zahlen vom Typ: "composed equal numbers" resultieren, wurde bereits in den Vorworten Bezug genommen. In Kapitel 3 (Der Beweis der Vermutung, dass es unendlich viele Primzahl-Zwillinge mod 2 gibt) wurde bereits nachgewiesen, dass in den direkten Umgebungen der geraden Zahlen: $n_g = \Pi p_i$, $p_i = 2$ bis p_n und $n_g = n!$ unendlich viele Primzahlen vom Typ: $CM_p = \Pi p_i - 1$ und $CM_p = n! - 1$ auftreten, sowie unendlich viele Primzahlen vom Typ: $CF_p = \Pi p_i + 1$ und $CF_p = n! + 1$. Die geraden Zahlen, in deren direkter Umgebung diese Primzahlen auftreten, sind hier Zahlen, die aus Fakultäten von Primzahlen bestehen ("composed equal numbers").

In **Tab. 31** wird am Beispiel der Folge: $n_g = \Sigma 2^n$, n = 1 bis n = 30 demonstriert, dass auch in der direkten Umgebung dieser geraden Zahlen beliebig viele Primzahlen vom Typ CM_p und CF_p auftreten. In **Tab. 32** wurden die Primzahlen CM_p und CF_p die in der Umgebung von $n_g = 2 \cdot \Sigma 2^n$ für n = 1 bis n = 30 auftreten ermittelt.
In **Tab. 33** wird demonstriert, dass in der direkten Umgebung der geraden Zahlen $n_g = 2 \cdot p$, p = 3 bis p = 646969601 beliebig viele Primzahlen CM_p und CF_p auftreten können.
Aus diesen Tabellen (**Tab. 31** bis **Tab. 32**) folgt, dass es unendlich viele zusammengesetzte gerade Zahlen n_g ("composed equal numbers") gibt, in deren direkter Umgebung Primzahlen vom Typ CM_p und CF_p auftreten, dies sind nämlich alle (unendlich vielen) ungeraden Primzahlen.

Tab. 31 - Die Primzahl-Dichten im Umfeld der geraden Zahlen $n_g = \Sigma 2^n$, für n = 1 bis n = 30:

$\Sigma 2^n$:=	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$.=
$\Sigma 2^n$(n =1 bis 2) = 6 $n_g = 2 \cdot 3$		-3 ------	-1 ------	+1	+5 ------	+7	10
$\Sigma 2^n$(n = 1 bis 3) = 14 $n_g = 2 \cdot 7$	-7	-3 --------	-1	+3 ------	+5	+9	16
$\Sigma 2^n$(n =1 bis 4) = 30 $n_g = 2 \cdot 3 \cdot 5$	-11	-7	-1 -------	+1	+7	+11	22
$\Sigma 2^n$(n = 1 bis 5) = 62 $n_g = 2 \cdot 31$	-15	-9	-1	+5	+9 ------	+11	26
$\Sigma 2^n$(n = 1 bis 6) = 126 $n_g = 2 \cdot 3^2 \cdot 7$	-19 -----	-17	-13	+1	+5	+11	30
$\Sigma 2^n$(n = 1 bis 7) = 254 $n_g = 2 \cdot 127$	-15 -----	-13	-3	+3	+9	+15	30
$\Sigma 2^n$(n = 1 bis 8) = 510 $n_g = 2 \cdot 3 \cdot 5 \cdot 17$	-11	-7	-1	+11 -----	+13	+31	42
$\Sigma 2^n$(n = 1 bis 9) = 1.022 $n_g = 2 \cdot 7 \cdot 127$	-9	-3	-1	+9	+11	+17	26

$n_g = 2 \cdot 3^2 \cdot 5 \cdot 7 \cdot 13$

Tab. 31 - Fortsetzung

$\Sigma 2^n(n = 1 \text{ bis } 13) = 16.382$ $n_g = 2 \cdot 8191$	-19	-13	-1	+29	+35	+39	58
$\Sigma 2^n(n = 1 \text{ bis } 14) = 32.766$ $n_g = 2 \cdot 3 \cdot 43 \cdot 127$	-37	-25	-15	+3 ------	+5	+27	64
$\Sigma 2^n(n = 1 \text{ bis } 15) = 65.534$ $n_g = 2 \cdot 7 \cdot 31 \cdot 151$	-37	-15 -----	-13	+3 ------	+5	+9	46
$\Sigma 2^n(n = 1 \text{ bis } 16) = 131.070$ $n_g = 2 \cdot 3 \cdot 5 \cdot 17 \cdot 257$	-29	-11	-7	+1	+31	+41	48
$\Sigma 2^n(n = 1 \text{ bis } 17) = 262.142$ $n_g = 2 \cdot 131071$	-15	-9	-3	+5	+9 ------	+11	26
$\Sigma 2^n(n = 1 \text{ bis } 18) = 524.286$ $n_g = 2 \cdot 3^3 \cdot 7 \cdot 19$	-29	-25	-17	+1	23	+55	84
$\Sigma 2^n(n = 1 \text{ bis } 19) = 1.048.574$ $n_g = 2 \cdot 113 \cdot 4799$	-15	-3 ------	-1	+9	+15	+27	42
$\Sigma 2^n(n = 1 \text{ bis } 20) = 2.097.150$ $n_g = 2 \cdot 3 \cdot 5^2 \cdot 11 \cdot 31 \cdot 41$	-19	-17	-7	+19	+61	+73	92
$\Sigma 2^n(n = 1 \text{ bis } 21) = 4.194.302$ $n_g = 2 \cdot 7^2 \cdot 127 \cdot 337$	-25	-15	-1	+17	+27	+51	76
$\Sigma 2^n(n = 1 \text{ bis } 22) = 8.388.606$ $n_g = 2 \cdot 3 \cdot 23 \cdot 89 \cdot 683$	-25	-19	-13	+11 -----	+13	+17	42
$\Sigma 2^n(n = 1 \text{ bis } 23) = 16.777.214$ $n_g = 2 \cdot 47 \cdot 178481$	-31	-15	-1	+45	+75 -----	+77	108
$\Sigma 2^n(n = 1 \text{ bis } 24) = 33.554.430$ $n_g = 2 \cdot 3^2 \cdot 5 \cdot 7 \cdot 13 \cdot 17 \cdot 241$	-59	-47	-37	+37	+43	+71	130
$\Sigma 2^n(n = 1 \text{ bis } 25) = 67.108.862$ $n_g = 2 \cdot 31 \cdot 601 \cdot 1801$	-43	-25	-3	+17	+51	+57	100
$\Sigma 2^b(n = 1 \text{ bis } 26) = 134.217.726$ $n_g = 2 \cdot 3 \cdot 2731 \cdot 8191$	-109	-77	-37	+31	+47	+53	162
$\Sigma 2^n(n = 1 \text{ bis } 27) = 268.435.454$ $n_g = 2 \cdot 7 \cdot 73 \cdot 262657$	-93	-87	-55	+5	+9	+39	132
$\Sigma 2^n(n = 1 \text{ bis } 28) = 536.870.910$ $n_g = 2 \cdot 3 \cdot 5 \cdot 29 \cdot 43 \cdot 113 \cdot 127$	-41	-31	-1	+13	+41	+91	132
$\Sigma 2^n(n = 1 \text{ bis } 29) = 1.073.741.822$ $n_g = 2 \cdot 233 \cdot 1103 \cdot 2089$	-81	-39	-33	+5	+9 ------	+11	92
$\Sigma 2^n(n = 1 \text{ bis } 30) = 2.147.483.646$ $n_g = 2 \cdot 3^2 \cdot 7 \cdot 11\ 31 \cdot 151 \cdot 331$	-67	-59	-17	+1	+13	+47	114

Die nachfolgende Primzahl vom Typ CF_p ist $\Sigma 2^n(n = 1 \text{ bis } 60) + 1 = 2305843009213693951$

$\Sigma 2^n(n = 1 \text{ bis } 60) = 2305843009213693950$ -29 +1
$n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 10965933772081$

Anmerkung:

Im Hinblick auf den hohen Rechenaufwand wurden nur die Primzahlen: p_3 und p_4 ermittelt, die einen Zwilling mod 30 bilden, in dem die Zahl $\Sigma 2^n(n= 1 \text{ bis } 60)$ "extrem rechts" situiert ist. Die gerade Zahl, die auf die Primzahl $p_3 = \Sigma 2^n\text{-}29 = 2305843009213693921$ folgt, ist $n_{g\,1} = p_3 + 1 = 2305843009213693922$. Diese Zahl besteht aus der Primzahl-Fakultät $n_{g\,1} = 2 \cdot 3^2 \cdot 13 \cdot 9841209441084163$ und kommt in der Folge $\Sigma 2^n$ nicht vor.

Anmerkungen und Schlussfolgerungen:

1. In **Tab. 17** treten zwölf Primzahlen vom Typ CM_p und acht vom Typ CF_p auf, wobei die Primzahlen $CM_p = (\Sigma 2^n = 2^4 + 2^3 + 2^2 + 2 = 30) - 1 = \mathbf{29}$ und $CF_p = \Sigma 2^n = 30 + 1 = \mathbf{31}$ den Zwilling mod 2: **(CM_p = 29, CF_p = 31)** bilden. Einen weiterer Zwilling mod 2 bilden die Primzahlen: **(CM_p = 5, CF_p = 7)**, die auch als die Primzahlen: $M_p = 2^3 - 1 = \mathbf{7}$ und $F_p = 2^2 + 1 = \mathbf{5}$ bekannt sind.

Anmerkungen und Schlussfolgerungen:

1. In **Tab. 17** treten zwölf Primzahlen vom Typ CM_p und acht vom Typ CF_p auf, wobei die Primzahlen $CM_p = (\Sigma 2^n = 2^4 + 2^3 + 2^2 + 2 = 30) - 1 = \mathbf{29}$ und $CF_p = \Sigma 2^n = 30 + 1 = \mathbf{31}$ den Zwilling mod 2: (CM_p = **29**, CF_p = **31**) bilden. Einen weiterer Zwilling mod 2 bilden die Primzahlen: (CM_p = **5**, CF_p = **7**), die auch als die Primzahlen: $M_p = 2^3 - 1 = \mathbf{7}$ und $F_p = 2^2 + 1 = \mathbf{5}$ bekannt sind.
2 Die geraden Zahlen der (hier definierten) Folge: $\Sigma 2^n$.treten zyklisch in den vier Endziffern: EZ6, EZ4, EZ0, EZ2, EZ6, EZ4, EZ0, EZ2 usw. auf, und aus diesem Sachverhalt folgt, dass Zwillinge mod 2 vom Typ (CM_p, CF_p) in dieser Folge nur bei den geraden Zahlen $n_g = \Sigma 2^n$ möglich sind, in denen die Endziffern: EZ2 oder EZ0 auftreten.

3. Es ist offenkundig, dass in der Folge der geraden Zahlen $\Sigma 2^n$(n= 1 bis n $\rightarrow \infty$) die nachstehenden Dirichlet-Progressionen auftreten

<u>D-Progression 1:</u> $\Sigma 2^n$(n = 1, n = 1 + 4n), EZ6 - hier treten (fallweise) Primzahlen $CF_p = \Sigma 2^n + 1$ mit der Endziffer EZ7 in einer unendlichen Zahl auf.

<u>D-Progression 2:</u> $\Sigma 2^n$(n = 1 , n = 2 +4n), EZ4 - hier treten (fallweise) Primzahlen $CM_p = \Sigma 2^n$:-1 mit der Endziffer EZ3 auf.

<u>D-Progression 3:</u> $\Sigma 2^n$(n = 1, n = 3 +4n), EZ0 - hier treten (fallweise) sowohl Primzahlen vom Typ: $CM_p = \Sigma 2^n - 1$, EZ9 bzw. $CF_p = \Sigma 2^n + 1$, EZ1 auf, die auch Zwillinge mod 2 bilden können.

<u>D-Progression 4:</u> $\Sigma 2^n$(n = 1, n = 4 + 4n), EZ2 - hier treten (fallweise) sowohl Primzahlen vom Typ: $CM_p = \Sigma 2^n - 1$, EZ1 bzw. $CF_p = \Sigma 2^n + 1$, EZ3 auf, die auch Zwillinge mod 2 bilden können.

4. Nachdem in der Folge $n_g = \Sigma 2^n$ keine geraden Zahlen mit der Endziffer EZ8 auftreten können, ist es unmöglich, dass in dieser Folge Primzahlen vom Typ: CM_p, EZ7 auftreten. Des weiteren können in dieser Folge auch keine Primzahlen vom Typ: CF_p, EZ9 auftreten.

5. Aus **Tab. 31** gewinnt man die wichtige Einsicht, dass die hier auftretenden Differenzen (Δ_1 bis Δ_6), die zwischen den Primzahlen (p_1 bis p_6) und den (jeweiligen) geraden Zahlen $\Sigma 2^n$auftreten, niemals gleich den Primzahlen sein können, die in den jeweiligen fakultativen Bildungsgesetzen von $\Sigma 2^n$auftreten. Diese Differenz-Abstände können stets nur Primzahlen sein, die in den Primzahl-Fakultäten von $\Sigma 2^n$ nicht auftreten, bzw. aus Fakultäten dieser Primzahlen komponiert sein. Nach dem gleichen Muster werden die Abstände (Δ_1 bis Δ_6) auch in den Umgebungen der geraden Zahlen $n_g = \Pi p_i$ gebildet.

6. In der nachfolgenden **Tab. 32** werden die Primzahl-Dichten bestimmt, die in den Umgebungen der geraden Zahlen $n_g = 2 \cdot \Sigma 2^n$auftreten, die also in den (so) vergrößerten Zahlen-Intervallen auftreten.

Tab. 32 - Die Primzahl-Dichten im Umfeld der geraden Zahlen $n_g = 2 \cdot \Sigma 2^n$, für n = 1 bis n = 30):

$2 \cdot \Sigma 2^n$:=	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma \Delta$:=
$2 \cdot \Sigma 2^n$(n = 1 bis 2) = 12	-7 --------	-5	-1 -------	+1	+5 ------	+7	14
$n_g = 2^2 \cdot 3$							
$2 \cdot \Sigma 2^n$(n = 1 bis 3) = 28	-11 -----	-9	-5	+1 ------	+3	+9	20
$n_g = 2^2 \cdot 7$							
$2 \cdot \Sigma 2^n$(n = 1 bis 4) = 60	-13	-7	-1 -------	+1	+7	+11	24
$n_g = 2^2 \cdot 3 \cdot 5$							
$2 \cdot \Sigma 2^n$(n = 1 bis 5) = 124	-17 -----	-15	-11	+3	+7	+13	30
$n_g = 2^2 \cdot 31$							
$2 \cdot \Sigma 2^n$(n = 1 bis 6) = 252	-13 -----	-11	-1	+5	+11	+17	30
$n_g = 2^2 \cdot 3^2 \cdot 7$							
$2 \cdot \Sigma 2^n$(n = 1 bis 7) = 508	-17	-9	-5	+1	+13	+15	32

Tab. 32 - Fortsetzung

$2 \cdot \Sigma 2^n$(n= 1 bis 10) = 4.092 $n_g = 2^2 \cdot 3 \cdot 11 \cdot 31$	-19	-13	-1 -------+1		7	+19	38
$2 \cdot \Sigma 2^n$(n = 1 bis 11) = 8,188 $n_g = 2^2 \cdot 23 \cdot 89$	-21	-17	-9	+3	+21	+31	52
$2 \cdot \Sigma 2^n$(n = 1 bis 12) = 16.380 $n_g = 2^2 \cdot 3^2 \cdot 5 \cdot 7 \cdot 13$	-19 ----- -17		-11	+1	+31	+37	56
$2 \cdot \Sigma 2^n$(n = 1 bis 13) = 32.764 $n_g = 2^2 \cdot 8191$	-47	-45	-15	+7	+23	+29	76
$2 \cdot \Sigma 2^n$(n = 1 bis 14) = 65.532 $n_g = 2^2 \cdot 3 \cdot 43 \cdot 127$	-35	-13	-11	+5	+7	+15	50
$2 \cdot \Sigma 2^n$(n = 1 bis 15) = 131.068 $n_g = 2^2 \cdot 7 \cdot 31 \cdot 151$	-27	-9	-5	+3	+33	+43	70
$2 \cdot \Sigma 2^n$(n = 1 bis 16) = 262.140 $n_g = 2^2 \cdot 3 \cdot 5 \cdot 17 \cdot 257$	-13	-7	-1	+7	+11 -----+13		26
$2 \cdot \Sigma 2^n$(n = 1 bis 17) = 524.284 $n_g = 2^2 \cdot 131071$	-27	-23	-15	+3	+25	+57	84
$2 \cdot \Sigma 2^n$(n = 1 bis 18) = 1.048.572 $n_g = 2^2 \cdot 3^3 \cdot 7 \cdot 19$	-23	-13	-1 ------ +1		+11	+17	40
$2 \cdot \Sigma 2^n$(n = 1 bis 19) = 2.097.148 $n_g = 2^2 \cdot 113 \cdot 4799$	-17 ----- -15		-5	+21	+63	+75	92
$2 \cdot \Sigma 2^n$(n = 1 bis 20) = 4.194.300 $n_g = 2^2 \cdot 3 \cdot 5^2 \cdot 11 \cdot 31 \cdot 41$	-29	-23	-13	+1	+19	+29	58
$2 \cdot \Sigma 2^n$(n= 1 bis 21) = 8.388.604 $n_g = 2^2 \cdot 7^2 \cdot 127 \cdot 337$	-23	-17	-11	+13	+15	+19	42
$2 \cdot \Sigma 2^n$(n = 1 bis 22) = 16.777.212 $n_g = 2^2 \cdot 3 \cdot 23 \cdot 89 \cdot 683$	-59	-29	-13	+1	+47	+77	136
$2 \cdot \Sigma 2^n$(n = 1 bis 23) = 33.554.428 $n_g = 2^2 \cdot 47 \cdot 178481$	.57	-45	-35	+39	+45	+75	132
$2 \cdot \Sigma 2^n$(n = 1 bis 24) = 71.118.860 $n_g = 2^2 \cdot 3^2 \cdot 5 \cdot 7 \cdot 13 \cdot 17 \cdot 241$	-51 ----- -49		-39	+9	+17	+39	90
$2 \cdot \Sigma 2^n$(n = 1 bis 25) = 134.217.724 $n_g = 2^2 \cdot 31 \cdot 601 \cdot 1801$	-107	-75	-35	+33	+49	+55	162
$2 \cdot \Sigma 2^n$(n = 1 bis 26) = 268.435.452 $n_g = 2^2 \cdot 3 \cdot 2731 \cdot 8191$	-91	-65	-53	+7	+11	+41	132
$2 \cdot \Sigma 2^n$(n = 1 bis 27) = 536.870.908 $n_g = 2^2 \cdot 7 \cdot 73 \cdot 262657$	-59	-39	-29	+1	+15	+43	102
$2 \cdot \Sigma 2^n$(n= 1 bis 28) = 1.073.741.820 $n_g = 2^2 \cdot 3 \cdot 5 \cdot 29 \cdot 43 \cdot 113 \cdot 127$	-79	-37	-31	+7	+11 -----+13		92
$2 \cdot \Sigma 2^n$(n = 1 bis 29) = 2.147.483.644 $n_g = 2^2 \cdot 233 \cdot 1103 \cdot 2089$	-65	-57	-15	+3	+15	+49	114
$2 \cdot \Sigma 2^n$(n = 1 bis 30) = 4.294.967.292 $n_g = 2^2 \cdot 3^2 \cdot 7 \cdot 11 \cdot 31 \cdot 151 \cdot 331$	-61	-13	-1	+19	+65	+79	140

Anmerkungen und Schlussfolgerungen:

1. In der Folge $n_g = 2 \cdot \Sigma 2^n$(n = 1 bis 30) treten fünf Primzahlen CM_p und CF_p auf, die Zwillinge mod 2 bilden und des weiteren drei Primzahlen CM_p sowie sechs Primzahlen CF_p, die keine Zwillinge mod 2 bilden.

2. In den Umgebungen der Zahlen $n_g = 2 \cdot \Sigma 2^n$(n = 1 bis n = 30) treten weitere zwölf Zwillinge mod 2 auf, die durch gestrichelte Linien markiert wurden. Eine Prüfung ergibt, dass alle (siebzehn) Zwillinge mod 2 stets (alternativ) in den Folgen: n_g = 12 + 30n, 18 + 30n, 30 + 30n auftreten.
Auf die Anführungen dieser (jeweiligen) geraden Zahlen (n_g) kann hier verzichtet werden - ein kritischer Leser ist in der Lage diese Zahlen (n_g) und ihre fakultativen Kompositionen selbst zu ermitteln.

auf, die durch gestrichelte Linien markiert wurden. Eine Prüfung ergibt, dass alle (siebzehn) Zwillinge mod 2 stets (alternativ) in den Folgen: $n_g = 12 + 30n$, $18 + 30n$, $30 + 30n$ auftreten.
Auf die Anführungen dieser (jeweiligen) geraden Zahlen (n_g) kann hier verzichtet werden - ein kritischer Leser ist in der Lage diese Zahlen (n_g) und ihre fakultativen Kompositionen selbst zu ermitteln.

3. Aus **Tab. 32** kann entnommen werden, dass die Abstände: Δ_1 bis Δ_6 zwischen den geraden Zahlen (n_g) niemals den Primzahlen gleichen können, die fakultativ in diesen geraden Zahlen auftreten. Insoweit wird diese, auch in den Tabellen: **Tab. 5, Tab. 6, Tab. 7, Tab. 17** erkannte Gesetzmäßigkeit bestätigt. Die Abstände $\Delta = \pm 1$ wurden in **Tab. 32** fett gedruckt hervorgehoben.

2. In **Tab. 32** treten 29 Zwillinge mod 2 auf, die durch gestrichelte Linien markiert wurden. Diese Zwillinge sind allesamt Elemente der Folgen: $n_g = 12 + 30n$, $18 + 30n$ bzw. $30 + 30n$, was in der nachfolgenden **Tab. 33** demonstriert wird.
In **Tab. 33** werden auch die fakultativen Komposition der geraden Zahlen dieser Folgen abgeleitet.

3. Aus den Schwankungen der Primzahl-Dichten, die in den Umgebungen der Zahlen $n_g = 2 \cdot p_i$ auftreten, ist die Schlussfolgerung abzuleiten, dass diese geraden Zahlen in Intervallen mit unterschiedlichen Primzahl-Dichten situiert sind und dabei symmetrische oder asymmetrisch zwischen den direkt anrainenden Primzahlen: p_3 und p_4 lokalisiert sind.

4. Daraus ist die Schlussfolgerung abzuleiten, dass auch die Mersenne- und Fermat-Primzahlen, die in den Dirichlet-Progressionen: $2^{1+4n}-1$, $2^{3+4n}-1$ bzw. $2^{4+4n}+1$ auftreten ebenfalls in Zahlen-Intervallen auftreten, in denen unterschiedliche Primzahl-Dichten vorkommen.
Sofern die geraden Zahlen der Folgen: 2^{1+4n} und 2^{3+4n} direkt auf die (jeweilige) Primzahl p_3 folgt, ergibt sich eine Mersenne-Primzahl: $M_p = 2^n - 1$. EZ1 bzw. EZ7.
Sofern die geraden Zahlen der Folge: 2^{4+4n} direkt vor der (jeweiligen) Primzahl p_4 situiert ist, ergibt sich eine Fermat-Primzahl: $F_p = 2^n + 1$. EZ7.

5. Durch diese Untersuchungen kann das Problem der Primzahlen: M_p und F_p als gelöst angesehen werden.

In **Tab. 33** sind die Primzahl-Dichten in den Umgebungen der geraden Zahlen $n_g = 2 \cdot p_i$ (von $p_i = 3$ bis $p_i = 5175756821$) abgeleitet worden. Aus **Tab. 33** ist ersichtlich, dass in den Abständen $\Delta_{\cdot} = \pm p_i$ keine primen Zahlen im Umfeld der geraden Zahlen $n_g = 2 \cdot p_i$ auftreten können, wohl aber Primzahlen vom Typ: CM_p. und vom Typ: CF_p.

Tab. 33 - Die Primzahl-Dichten in den Umgebungen der Zahlen $n_g = 2\ p_i$ ($p_i = 3$ bis $p_i = 5175756821$):

$n_g = 2 \cdot p_i :=$		Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta :=$
$n_g = 2 \cdot 11$	= 22	-9	-5	-3	+7	+9	+15	24
$n_g = 2 \cdot 13$	= 26	-9	-7	-3	+3	+5	+11	20
$n_g = 2 \cdot 17$	= 34	-15	-5	-3	+3	+7	+9	24
$n_g = 2 \cdot 19$	= 38	-9	-5	**-1**	+3	+5	+9	18
$n_g = 2 \cdot 23$	= 46	-9	-5	-3	**+1**	+7	+13	24
$n_g = 2 \cdot 29$	= 58	-15	-9	-3	**+1**	+3	+9	24
$n_g = 2 \cdot 31$	= 62	-9	-3	**-1**	+5	+9	+11	20
$n_g = 2 \cdot 37$	= 74	-7	-3	**-1**	+5	+9	+13	20
$n_g = 2 \cdot 41$	= 82	-9	-7	-3	**+1**	+5	+17	26
$n_g = 2 \cdot 43$	= 86	-13	-7	-3	+3	+11	+15	28
$n_g = 2 \cdot 47$	= 94	-15	-11	-5	+3	+7	+9	24
$n_g = 2 \cdot 53$	= 106	-9	-5	-3	**+1**	+7	+21	30
$n_g = 2 \cdot 59$	= 118	-11	-9	-5	+9	+13	+19	30
$n_g = 2 \cdot 61$	= 122	-15	-13	-7	+5	+9	+15	30
$n_g = 2 \cdot 67$	= 134	-21	-7	-3	+3	+5	+15	36
$n_g = 2 \cdot 71$	= 142	-11	-5	-3	+7	+9	+15	26
$n_g = 2 \cdot 73$	= 146	-15	-9	-7	+3	+5	+11	26
$n_g = 2 \cdot 79$	= 158	-9	-7	**-1**	+5	+9	+15	24
$n_g = 2 \cdot 83$	= 166	-15	-9	-3	**+1**	+7	+13	28

Tab. 33 - Fortsetzung

$n_g = 2 \cdot 103$	=206	-13	-9	-7	+5	+17	+23	36
$n_g = 2 \cdot 107$	=214	-17 -----	-15	-3	+9	+13 ----	+15	32
$n_g = 2 \cdot 113$	= 226	-27	-15	-3	**+1** ------	+3	+7	34
$n_g = 2 \cdot 131$	= 262	-21	-11	-5	**+1**	+7 ------	+9	30
$n_g = 2 \cdot 139$	= 278	-9 -------	-7	**-1**	+3 ------	+5	+15	24
$n_g = 2 \cdot 12041$	= 24.082	-21	-11	-5	**+1**	+9	+15	36
$n_g = 2 \cdot 12049$	= 24098	-15	-7	**-1**	+5	+9 ------	+11	26
$n_g = 2 \cdot 102149$	= 204298	-125	-65	-47	**+1**	+3	13	138
$n_g = 2 \cdot 102181$	= 204362	-9	-3	**-1**	+5	+9	+15	24
$n_g = 2 \cdot 646969601$	= 1293939202	-41	-35------	-33	**+1**	+9	15	96
$n_g = 2 \cdot 5175756821$	= 15351513642	-85	-35	**-1**	+35	+59 -----	+61	146

Anmerkungen und Schlussfolgerungen:

1 In **Tab. 33** treten neun Primzahlen vom Typ CM_p und CF_p auf - die Abstände Δ_3= **-1** und Δ_4= **+1** wurden fett gedruckt markiert.

2. In **Tab. 33** treten 31 Zwillinge mod 2 auf, die "strichliniert" markiert wurden.

In **Tab. 34** wurden die Zwillinge mod 2 aufgelistet, die in *Folge 1* (12 + n x30), in *Folge 2* (18 + n x30) und in *Folge 3* (30 + n x30) in wachsenden Zahlen-Bereichen auftreten.

Tab. 34 - Die fakultative Darstellung der geraden Zahlen, die in **Tab. 19** in den Zwillingen mod 2 liegen:

Zwilling mod 2: :	(p, n_g, p+2):=	*Folge:*	$n_g = \prod p_i$.=	n:=
1. Zwilling,(x1)	**(11, 13)**	12 + n x 30	$12 = 2^2 \cdot 3$	0
1. Zwilling,(x 2)	**(17, 19)**	18 + n x 30	$18 = 2 \cdot 3 \cdot 5$	0
2. Zwilling,(x 2)	**(29, 31)**	30 + n x 30	$30 = 2 \cdot 3 \cdot 5$	0
3. Zwilling,(x 2)	**(41, 43)**	12 + n x 30	$42 = 2 \cdot 3 \cdot 7$	1
4. Zwilling,(x 2)	**(59, 61)**	30 + n x 30	$60 = 2^2 \cdot 3 \cdot 5$	1
5. Zwilling,(x 1)	**(71, 73)**	12 + n x 30	$72 = 2^3 \cdot 3^2$	2
6. Zwilling,(x 2)	**(101, 103)**	12 + n x 30	$102 = 2 \cdot 3 \cdot 17$	3
7. Zwilling,(x 2)	**(107, 109)**	18 + n x 30	$108 = 2^2 \cdot 3^3$	3
8. Zwilling,(x 1)	**(137, 139)**	18 + n x 30	$138 = 2 \cdot 3 \cdot 23$	4
9. Zwilling,(x 3)	**(149, 151)**	30 + n x 30	$150 = 2 \cdot 3 \cdot 5^2$	4
10. Zwilling,(x 1)	**(179, 181)**	30 + n x 30	$180 = 2^2 \cdot 3^2 \cdot 5$	5
11. Zwilling,(x 1)	**(191, 193)**	12 + n x 30	$192 = 2^6 \cdot 3$	6
12. Zwilling,(x 1)	**(227, 229)**	18 + n x 30	$228 = 2^2 \cdot 3 \cdot 19$	7
13. Zwilling,(x 2)	**(269, 271)**	30 + n x 30	$270 = 2 \cdot 3^2 \cdot 5$	8
14. Zwilling,(x 1)	**(281, 283)**	12 + n x 30	$282 = 2 \cdot 3 \cdot 47$	9
15. Zwilling,(x 1)	**(24107, 24109)**	18 + n x 30	$24108 = 2^2 \cdot 3 \cdot 73 \cdot 233$	803
16. Zwilling,(x 1)	**(204299, 204301)**	30 + n x 30	$204300 = 2^2 \cdot 3^2 \cdot 5^2 \cdot 227$	6809
17. Zwilling,(x 1)	**(1293939167, 1293939169)**	18 + n x 30	$1293939168 = 2^5 \cdot 3 \cdot 29 \cdot 464777$	43131305
18. Zwilling,(x 1)	**(15351513701, 15351513703)**	12 + n x 30	$15361513702 = 2 \cdot 3 \cdot 7 \cdot 19 \cdot 1923749$	511717123

Anmerkungen und Schlussfolgerungen:

1 In **Tab. 34** werden die Dirichlet-Progressionen, die dem Beweis der Vermutung, dass Zwillinge mod 2 in einer unendlichen Zahl auftreten (und dies nur in den Umgebungen der drei Folgen: n_g = 1 2 + 30n, 18 + 30n, 30 + 30n) eineindeutig bestätigt.

2. Aus **Tab. 34** sind die (differenten) fakultativen Kompositionen der geraden Zahlen der drei Folgen. n_g = 12 + 30n, 18 + 30n, 30 + 30n ersichtlich, die man per PC-Programme für alle bekannten Zwillinge mod 2 ableiten kann.

2. Aus **Tab. 34** sind die (differenten) fakultativen Kompositionen der geraden Zahlen der drei Folgen. $n_g = 12 + 30n$, $18 + 30n$, $30 + 30n$ ersichtlich, die man per PC-Programme für alle bekannten Zwillinge mod 2 ableiten kann.

3. Es ist offensichtlich, dass man für beliebige Primzahl-Fakultäten $n_g = \Pi p_i$ (von $p_i = 2$ bis $p_i = p_n$), in denen nicht alle Primzahlen $p_i > 2$ vorkommen, die präsenten aber in beliebigen Potenzen auftreten können, die in ihren Umgebungen auftretenden Primzahl-Dichten bestimmen kann, wie auch die in denen außer der Primzahl $p_1 = 2$ weitere Primzahlen $p_i \geq 3$ auftreten müssen, die lückenhaft auftreten können und (fallweise) in Potenzen vorkommen.

4. Eine volle Bestätigung findet in **Tab. 34** auch das in dieser Monographie deduzierte Gesetz, dass inmitten der (aller) Zwillinge mod 2 stets nur gerade Zahlen der Folgen: $n_g = 12 + 30n$, $18 + 30n$ und $30 + 30n$ auftreten können, die in der Zahlenfolge $n \to \infty$ in einer unendlichen Zahl auftreten.
Eine unendliche Teilmenge dieser geraden Zahlen (n_g) ist inmitten von Zwillingen mod 2 situiert. Damit liegt der Beweis für die Vermutung vor, dass Zwillinge mod 2 in unendlicher Zahl vorkommen.

5 Die drei Folgen der geraden Zahlen (n_g), die inmitten der (aller) Zwillinge mod 2 situiert sind, können auch in der folgenden Notation zusammengefasst werden, in der die ersten drei Primzahlen: **2, 3, 5** sich zyklisch wiederholend auftreten.

$n_g = p \times 6 + 30n$, $p := 2, 3, 5$ und $n :\geq 0$ ... das zyklische Folgen-Gesetz der Zwillinge mod 2

Aus dem "zyklischen Folgen-Gesetz der Zwillinge mod 2" geht hervor, dass die ersten drei Primzahlen: p = **2, 3, 5** in alle Zwillinge mod 2 eingehen, diese Zwillinge in gewissem Sinne determinieren.

6. Aus **Tab. 34** ist ersichtlich, dass die "Dichten der Zwillinge mod 2" in steigenden Zahlen-Intervallen eine abnehmende Tendenz aufweisen, was auch aus **Anlage 1** und aus **Anlage 2** hervorgeht. Dabei gilt, dass außer dem Drilling: **(3, 5)**, (5, 7) keine weiteren Zwillinge mod 2 direkt aufeinander folgen können. Zwischen allen Zwillingen mod 2 müssen ab dem Zwilling **(11, 13)** stets Abstände mod 2n, $n \geq 2$ auftreten, d.h. die Abstände 4, 6, 8, 10, 12 usw.

----

11. Der Beweis, dass es unendlich viele Primzahlen $SM_p = n_u - 2$ und $SF_p = n_u + 2$ gibt:

Der Begriff der Primzahlen vom Typ: $SM_p = n_u - 2$ und $SF_p = n_u + 2$ wurde bereits in den Vorworten eingeführt. Diese Primzahlen können "Semi-Mersenne - und Semi-Fermat-Primzahlen" genannt werden. Die folgenden Beispiele verdeutlichen, dass zwei Primzahlen SM_p und SF_p einen Zwilling mod 4 bilden können bzw. einen Zwilling mod 6 oder einen Zwilling mod 2n, $n \geq 2$.

<u>Beispiel 1</u>: $SM_p = 15 - 2 = \mathbf{13}$, $SF_p = 15 + 2 = \mathbf{17}$ - **(13, 17)** ist ein Zwilling mod 4

<u>Beispiel 2</u>: $SM_p = 25 - 2 = \mathbf{23}$, $SF_p = 27 + 2 = \mathbf{29}$ - **(23, 29)** ist ein Zwilling mod 6

<u>Beispiel 3</u>: $SM_p = 91 - 2 = \mathbf{89}$, $SF_p = 95 + 2 = \mathbf{97}$ - **(89, 97)** ist ein Zwilling mod 8

<u>Beispiel 4</u>: $SM_p = (3021 - 2 = \mathbf{3019}$, $SF_p = 3117 + 2 = \mathbf{3119}$ - **(3019, 3119)** ein Zwilling mod 100

In den Vorworten wurden die Kriterien: (3) ... $n_u + 2 = \Pi p_i + 2$, $p_i \geq 3$ und (4) ... $n_u = \Pi p_i - 2$, $p_i \geq 3$ angegeben, aus denen sich (ebenfalls) der Beweis des Hauptsatzes der Primzahl-Theorie ableiten lässt. Des weiteren wurde in **Kapitel 5** der Beweis erbracht, dass es unendlich viele Zwillinge mod 4 gibt, in denen zwischen den benachbarten Primzahlen (p_1, $p_2 = p_1 + 4$) jeweils eine nicht prime ungerade Zahl, n_u und zwei gerade Zahlen: $n_{g,1}$ und $n_{g,2}$ situiert sind, was die Folge: (p_1, $n_{g,1}$, n_u, $n_{g,2}$, p_2) ergibt. Für die ungeraden Zahlen (n_u), die inmitten der Zwillinge mod 4 situiert sind, wurden in **Kapitel 5** die (hier wiederholt angegebenen Dirichlet-Prrogressionen) deduziert:

In **Tab. 35** wurden jeweils elf ungerade Zahlen (n_u) dieser sechs Folgen aufgelistet mit Angabe ihre Primzahl-Fakultäten Πp_i.

Tab. 35 - Die ersten elf ungeraden Zahlen, die in den Folgen: 1, 2, 3, 4, 5, 6 auftreten mit Angabe ihrer Primzahl- Fakultäten, Πp_i:

Folge 1	Folge 2	Folge 3	Folge 4	Folge 5	Folge 6	n:=
n_u = **19** = **19**	n_u = 15 = 3 · 5	n_u = **11** = **11**	n_u = 5 = 5	n_u = 9 = 3^2	n_u = 21 = 3 · 7	0
n_u = 39 = 3 · 13	n_u = 35 = 5 · 7	n_u = **31** = **31**	n_u = 25 = 5^2	n_u = **29** = **29**	n_u = **41** = **41**	1
n_u = **59** = **59**	n_u = 55 = 5 · 11	n_u = 51 = 3 · 17	n_u = 45 = 3^2 · 5	n_u = 49 = 7^2	n_u = **61** = **61**	2
n_u = **79** = **79**	n_u = 75 = 3 · 5^2	n_u = 171 = 3^2 · 19	n_u = 65 = 5 · 13	n_u = 69 = 3 · 23	n_u = 81 = 3^4	3
n_u = 99 = 3^2 · 11	n_u = 95 = 5 · 19	n_u = **71** = **71**	n_u = 85 = 5 · 17	n_u = **89** = **89**	n_u = **101** = **101**	4
n_u = 119 = 7 · 17	n_u = 115 = 5 · 23	n_u = 91 = 91 = 7 · 13	n_u = 105 = 3 · 5 · 7	n_u = **109** = **109**	n_u = 121 = 11^2	5
n_u = **139** = **139**	n_u = 135 = 3^3 5	n_u = 111 = 3 · 37	n_u = 125 = 5^3	n_u = 129 = 3 · 43	n_u = 141 = 3 · 47	6
n_u = 159 = 3 · 53	n_u = 155 = 5 · 31	n_u = **131** = **131**	n_u = 145 = 5 · 29	n_u = **149** = **149**	n_u = 161 = 7 · 23	7
n_u = **199** = **199**	n_u = 175 = 5^2 · 7	n_u = **151** = **151**	n_u = 165 = 3 · 5 · 11	n_u = 169 = 13^2	n_u = **181** = **181**	8
n_u = 219 = 3 · 73	n_u = 195 = 3 · 5 · 13	n_u = 171 = 3 · 57	n_u = 185 = 5 · 37	n_u = 189 = 3^3 · 7	n_u = 201 = 3 · 67	9
n_u = **239** = **239**	n_u = 215 = 5 · 43	n_u = **191** = **191**	n_u = 205 = 5 · 41	n_u = 209 = 11 · 19	n_u = 221 = 13 · 17	10

Anmerkungen:

1. Die in **Tab. 35** vorgenommene Zerlegung der ungeraden Zahlen der sechs Folgen in ihre Primzahl-Fakultäten weist zahlreiche Primzahlen vom Typ: $n_u = p$ auf. Diese Zahlen können offensichtlich nicht inmitten von Zwillingen mod 4 situiert sein. In den nahen Umgebungen dieser Zahlen ($n_u = p$) können jedoch Zwillinge mod 4 auftreten, was in **Tab. 35** gezeigt wird. In der direkten Nachbarschaft der Primzahlen, die in den Folgen: Folge 1, Folge 3, Folge 5 und Folge 6 auftreten, kommen jedoch prime Zahlen vom Typ: $SM_p = p - 2$ und $SF_p = p + 2$ vor, was aus **Tab. 35** entnommen werden kann.

2. Nicht alle nicht primen ungeraden Zahlen, die in **Tab. 35** in den sechs Folgen (für n = 0 bis n = 11) angeführt wurden, sind inmitten von Zwillingen mod 4 situiert; dann können jedoch Zwillinge mod 4 in den Umgebungen dieser ungeraden Zahlen (n_u) auftreten, was ebenfalls aus **Tab. 35** folgt.

3. Aus **Tab. 35** ist zu entnehmen, dass die Folge 2 und die Folge 4 "primzahl-passive" Dirichlet-Progressionen bilden, in denen keine Primzahlen auftreten.

Tab. 36 Die Primzahl-Dichten in den Umgebungen der ungeraden Zahlen der sechs Folgen von **Tab. 22**:

Folge 1:	n:=	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$:=
n_u = **19**	0	-8	-6	**-2**	+4	+10	+12	20
n_u = 39 = 3 · 13	0	-10	-8	**-2**	**+2**	+4	+8	18
n_u = **59**	1	-16	-12	-6	+2	+8	+12	28
n_u = **79**	2	-12	-8	-6	+4	+10	+18	30
n_u = 99 = 3^2 · 11	3	-16	-10	**-2**	**+2**	+4	+8	24
n_u = 119 = 7 · 17	4	-12	-10	-6	+8	+12	+18	30
n_u = **139**	5	-12	-8	**-2**	+10	+12	+18	30
n_u = 159 = 3 · 53	6	-10	-8	**-2**	+4	+8	+14	24
n_u = **179**	7	-16	-12	-6	**+2**	+12	+14	30

n_u = **139**	5	-12	-8	-2	+10	+12	+18	30
$n_u = 159 = 3 \cdot 53$	6	-10	-8	-2	+4	+8	+14	24
n_u = **179**	7	-16	-12	-6	+2	+12	+14	30

Tab. 36 - Fortsetzung

Folge 1:	n:=	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$.=
n_u = **199**	8	-8	-6	-2	+12	+24	+28	36
$n_u = 219 = 3 \cdot 73$	9	-22	-20	-8	+4	+8	+10	32
Folge 2:								
$n_u = 15 = 3 \cdot 5$	0	-8	-4	-2	+2	+4	+8	16
$n_u = 35 = 5 \cdot 7$	1	-12	-6	-4	+2	+6	+8	20
$n_u = 55 = 5 \cdot 11$	2	-12	-8	-2	+4	+6	+12	24
$n_u = 75 = 3 \cdot 5^2$	3	-8	-4	-2	+4	+8	+14	22
$n_u = 95 = 5 \cdot 19$	4	-16	-12	-6	+2	+6	+8	24
$n_u = 115 = 5 \cdot 23$	5	-14	-4	-2	+2	+4	+16	30
$n_u = 135 = 3^3 \cdot 5$	6	-22	--8	-4	+2	+4	+14	36
$n_u = 155 = 5 \cdot 31$	7	-16	-6	-4	+2	+8	+12	28
$n_u = 175 = 5^2 \cdot 7$	8	-12	-8	-2	+4	+6	+16	28
$n_u = 195 = 3 \cdot 5 \cdot 13$	9	-14	-4	-2	+2	+4	+16	30
$n_u = 215 = 5 \cdot 43$	10	-18	-16	-4	+8	+12	+14	32
Folge 3:	n:=							
n_u = **11**	0	-8	-6	-4	+2	+6	+8	16
n_u = **31**	1	-12	-8	-2	+6	+10	+12	24
$n_u = 51 = 3 \cdot 17$	2	-10	-8	-4	+2	+8	+10	20
n_u = **71**	3	-12	-10	-4	+2	+8	+12	24
$n_u = 91 = 7 \cdot 13$	4	-18	-8	-2	+8	+10	+12	30
$n_u = 111 = 3 \cdot 37$	5	-8	-4	-2	+2	+16	+20	28
n_u = **131**	5	-22	-18	-4	+6	+8	+18	40
n_u = **151**	7	-14	-12	-2	+6	+12	+16	30
$n_u = 171 = 3^2 \cdot 19$	8	-14	-8	-4	+2	+8	+10	24
n_u = **191**	9	-18	-12	-10	+2	+6	+8	26
n_u = **211**	10	-18	-14	-12	+12	+16	+18	36
Folge 4:								
n_u = **5**	0			-2	+2	+6	+8	10
$n_u = 25 = 5^2$	1	-8	-6	-2	+4	+6	+12	20
$n_u = 45 = 3^2 \cdot 5$	2	-8	-4	-2	+2	+8	+14	22
$n_u = 65 = 5 \cdot 13$	3	-12	-6	-4	+2	+6	+8	20
$n_u = 85 = 5 \cdot 17$	4	-12	-6	-2	+4	+12	+16	28
$n_u = 105 = 3 \cdot 5 \cdot 75$	5	-8	-4	-2	+2	+4	+8	16
$n_u = 125 = 5^3$	6	-18	-16	-12	+2	+6	+12	30
$n_u = 145 = 5 \cdot 29$	7	-14	-8	-6	+4	+6	+12	26
$n_u = 165 = 3 \cdot 5 \cdot 11$	8	-14	-8	-2	+2	+8	+14	28
$n_u = 185 = 5 \cdot 37$	9	-12	-6	-4	+6	+8	+12	24
$n_u = 205 = 5 \cdot 41$	10	-12	-8	-6	+6	+18	+22	34
Folge 5:								
$n_u = 9 = 3^2$	0	-6	-4	-2	+2	+4	+8	14
n_u = **29**	1	-12	-10	-6	+2	+8	+12	24
$n_u = 49 = 7^2$	2	-8	-6	-2	+4	+10	+12	20
$n_u = 69 = 3 \cdot 23$	3	-10	-8	-2	+2	+4	+10	20
n_u = **89**	4	-16	-10	-6	+8	+12	+14	30
n_u = **109**	5	-8	-6	-2	+4	+18	+22	30
$n_u = 129 = 3 \cdot 43$	6	-20	-16	-2	+2	+8	+10	30

Tab. 36 - Fortsetzung.

	n:=	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$:=
Folge 5:								
$n_u = 209 = 11 \cdot 19$	10	-16 -----	-12	-10	**+2**	+14 -----	+18	34
Folge 6:								
$n_u = 21 = 3 \cdot 7$	0	-8 -------	-4	**-2** -------	**+2**	+8	+10	18
$n_u =$ **41**	1	-12	-10	-4	**+2** ------	+6	+12	24
$n_u =$ **61**	2	-14	-6 -------	**-2**	+6 ------	+10	+12	26
$n_u = 81 = 3^4$	3	-10	-8	**-2** -------	**+2**	+8	+16	26
$n_u =$ **101**	4	-18	-12	-4	**+2** ------	+6	+8	26
$n_u = 121 = 11^2$	5	-14	-12	-8	+6 ------	+10	+16	30
$n_u = 141 = 3 \cdot 47$	6	-10	-4	**-2**	+8	+10	+16	26
$n_u = 161 = 7 \cdot 23$	7	-12	-10	-4	**+2** ------	+6	+12	24
$n_u =$ **181**	8	-14	-8	**-2**	+10	+12 -----	+16	30
$n_u = 201 = 3 \cdot 67$	9	-8 ------	-4	**-2**	+10	+22	+26	34
$n_u = 221 = 13 \cdot 17$	10	-24	-22	-10	**+2** ------	+6	+8	32

Anmerkungen:

1. Die Primzahlen vom Typ SM_p und SF_p wurden durch die fett gedruckten Abstände Δ= **- 2** und **+ 2** markiert. und alle Zwillinge mod 4 wurden durch gestrichelte Linien gekennzeichnet.

2. Die Primzahlen: SM_p und SF_p können (wie aus **Tab. 23** ersichtlich) fallweise Zwillinge mod 4 bilden. In den sechs dargestellten Folgen treten 15 Zwillinge mod 4 vom Typ: (SM_p, SF_p) auf und in den weiteren Umgebungen dieser ungeraden Zahlen (die in den sechs Folgen auftreten), sind weitere 61 Zwillinge mod 4 situiert.

3. In **Tab. 36** wird der Sachverhalt (die Gesetzmäßigkeit) bestätigt, dass Zwillinge mod 4 nicht direkt aufeinander folgen können. Die einzige Ausnahme ist der "Drilling mod 4": **(3, 7) — (7, 11)**.

4. Eine leicht durchführbare Prüfung, die der Leser selbst vollziehen kann, führt zu dem Ergebnis, dass alle Zwillinge mod 4, die in **Tab. 36** auftreten, stets Elemente einer der sechs Folgen sind.

Für den Nachweis der Validität des Gesetzes, dass Zwillinge mod 4 in den hier deduzierten sechs Dirichlet-Progressionen in unendlicher Zahl auftreten, werden in **Tab. 37** weitere Elemente dieser Folgen untersucht, die in weit gelegenen Zahlen-Intervallen dieser Folgen situiert sind,

Tab. 37 Die Primzahl-Dichten in den Umgebungen der sechs Folgen in großen Zahlen-Intervallen:

	n:=	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$:=
Folge 1:								
$n_u = 2001519 =$ $= 19 + 1000075 \cdot 20 =$ $= 3^3 \cdot 457 \cdot 1621$	1000075	-68	-28	**-2** -------	**+2**	+32	+38	106
Folge 2:								
$n_u = 1000455 =$ $= 15 + 500022 \cdot 20 =$ $= 3 \cdot 5 \cdot 666697$	500022	-16	-4	**-2** ------	**+2**	+26	+56	72
Folge 3:								
$n_u = 20000571 =$ $= 1000028 \cdot 20 =$ $= 3 \cdot 7 \cdot 349 \cdot 2729$	1000028	-34	-4	**-2** -------	**+2**	+20	+26	60
Folge 4:								
$n_u = 2000145 =$ $= 5 + 100007 \cdot 20 =$ $= 3 \cdot 5 \cdot 7 \cdot 43 \cdot 443$	100007	-52	-38	**-2** -------	**+2**	+8	+32	84

$n_u = 2000145 =$ $= 5 + 100007 \cdot 20 =$ $= 3 \cdot 5 \cdot 7 \cdot 43 \cdot 443$	100007	-52	-38	-2 ------- +2		+8	+32	84

Tab. 37 - Fortsetzung

Folge 5:	n:=	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$:=
$n_u = 1001529 =$ $= 9 + 50076 \cdot 20 =$ $= 3^2 \cdot 257 \cdot 433$	50076	-38	-28	-2 -------	+2	+20	+22	60
Folge 6:								
$n_u = 20000841 =$ $= 21 + 1000041 \cdot 20 =$ $= 3 \cdot 7 \cdot 349 \cdot 2729$	1000041	-20	-4	-2 -------	+2	+20	+26	46

Anmerkungen und Schlussfolgerungen:

1. In **Tab. 37** wird der Beweis, dass es unendlich viele Zwillinge mod 4 gibt, vollauf bestätigt.

2. Die Aussage, die **Tab. 37** liefert, besagt, dass Zwillinge mod 4 stets nur in den sechs Dirichlet-Progressionen auftreten können, die stets in der Modularität 20 auftreten.

3. Einen tiefen Einblick vermitteln die (hier angeführten) Primzahl-Fakultäten der ungeraden Zahlen (n_u), die in den sechs Folgen auftreten. In allen diesen Fakultäten treten stets die Primzahlen: p = **3**, **5**, **7** auf, sofern diese in der Fakultät der ersten (bzw. zweiten) ungeraden Zahl der jeweiligen Folge auftreten.

4. Aus **Tab. 37** ist zu entnehmen, dass in den Umgebungen der relativ großen ungeraden Zahlen (n_u), die aus den 6 Folgen, in denen Zwillinge mod 4 auftreten können, relativ hohe Primzahl-Dichten auftreten. Dieser Sachverhalt wird in **Tab. 38** für eine 18-stellige Zahl der Folge 2 bestätigt.

Tab. 38 - Die Bestimmung der Primzahl-Dichte im Umfeld der Zahl n_u = 2.000.000.000.000..011.005

Folge 2:	n;=	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta$:=
$n_u = 20000000000000011005$ $= 5 + 4000000000002201 \cdot 20 =$ $= 3^2 \cdot 5 \cdot 44444444444689 = 3^2 \cdot 5 \cdot p_n$	400000000002201	-112	-104	-86	+2	+16	+44	156

Anmerkungen und Schlussfolgerungen:

1. In der Primzahl-Fakultät der in **Tab. 38** analysierten ungeraden Zahl der Folge 2 treten die beiden Primzahlen: p = **3** und p = **5** auf, die bereits im ersten Glied dieser Folge n_u = 15 = **3** · **5** auftreten

2. Ein Vergleich dieser Primzahl-Dichte, die in der Umgebung dieser 18-stelligen Zahl auftritt, mit der Primzahl-Dichte, die im Umfeld der ebenfalls 18-stelligen Zahl n_g = 19! auftritt (siehe **Tab. 7**) zeigt, dass die im Umfeld der Zahl n_u = 5 + 4000000000002201 · 20 eine 1,9872 -fach höhere Primzahl-Dichte auftritt, als im Umfeld der geraden Zahl n_g = 19! - man vergleiche die Summen-Abstände: $\Sigma\Delta$= 156 und $\Sigma\Delta$= 310, in denen jeweils sechs Primzahlen situiert sind. Der Quotient: 310 / 156 = 1,9872 ist hier das Maß für die höhere Primzahl-Dichte, die im Umfeld dieser ungeraden Zahl der Folge 2 vorliegt.

3. Die (tiefere) Ursache für die relativ hohe Primzahl-Dichte ist der enorme Abstand zwischen der Primzahl 5 und der Primzahl p_n, die in die ungerade Zahl (n_u) der Folge 2 eingehen.
In die gerade Zahl n_g = 19! gehen hingegen lückenlos alle Primzahlen von p = 2 bis p =19 ein, was für die geringe Primzahl-Dichte im Umfeld der Zahl n_g = 19! ursächlich ist.

----

1. Erkenntnis:
Alle Zwillinge mod 4, die in **Tab. 35** in Folge 1, Folge 2 und in Folge 3 auftreten, werden aus Primzahlen der Obergruppe I gebildet, die den Klassen: 1, 2, 3, 4 angehören. Diese Primzahlen haben bei der Teilung durch die Zahl 4 den Rest 1.

2. Erkenntnis:
Alle Zwillinge mod 4, die in **Tab. 35** in Folge 4, Folge 5 und in Folge 6 auftreten, werden aus Primzahlen der Obergruppe II gebildet, die den Klassen: 5, 6, 7, 8 angehören. Diese Primzahlen haben bei der Teilung durch die Zahl 4 den Rest 3.

3. Erkenntnis:
Der einzige Zwilling mod 4, in dem eine Primzahl situiert sein kann, ist der Zwilling mod 4 (3, 7) in dem die Primzahl p = 5 situiert ist, da alle zweistelligen ungeraden Zahlen mit der Endziffer EZ5 nicht prim sind.

4. Erkenntnis:
Die Zwillinge mod 4: (3, 7) --- (7, 11) sind die einzigen "verketteten" Zwillinge mod 4, die in der unendlichen Folge der Zwillinge mod 4 auftreten können, da innerhalb von Zwillingen mod 4 (mit der Ausnahme der Zwillinge mod 2 (3, 5) und (5,7), die in den Zwilling mod 4 (3. 7) eingehen, keine Zwillinge mod 2 situiert sein können.

5. Erkenntnis:
Ab dem Zwilling mod 4 (13, 17) sind alle direkt zu den Zwillingen mod 4 "benachbarten" Zwillinge mod 2 aus den Zwillingen mod 4 "ausgegliedert" - hier zwei Beispiele:

Beispiel 1:
Der Zwilling mod 2 (11,13) grenzt links an den Zwilling mod 4 (13, 17), wobei die Primzahl p = 13 diese beiden Zwillinge verbindet (in beiden vorkommt), was auch wie folgt notiert werden kann:
(11, 13) --- (13, 17).

Beispiel 2:
Der Zwilling mod 2 (17, 19) grenzt rechts an den Zwilling mod 4 (13, 17), wobei die Primzahl p = 17 diese beiden Zwillinge verbindet - in beiden vorkommt -, was auch wie folgt notiert werden kann:
(13, 17) --- (17, 19).

6. Erkenntnis:
Nachdem ab dem Zwilling mod 4 (3, 7), in dem die Primzahl p = 5 situiert ist, in allen nachfolgenden Zwillingen mod 4 keine Primzahlen situiert sein können, scheiden alle primen Zahlen der Folgen: 1, 3, 5 und 6 für das Auftreten von Zwillingen mod 4 in ihren direkten Umgebungen aus. Dies besagt, dass nur in den direkten Umgebungen der nicht primen Zahlen dieser Folgen Zwillinge mod 4 auftreten können.

Diese invarianten Eigenschaften der Zwillinge mod 4 wurden in **Anlage 2** dadurch hervorgehoben, dass die Primzahlen der Obergruppe I fett gedruckt wurden und die Primzahlen der Obergruppe II normal gedruckt wurden. Aus **Anlage 2** ist auch noch zu entnehmen in welchen Abständen Zwillinge mod 4 aufeinander folgen, und dass Zwillinge mod 4 der gleichen Obergruppe mehrfach aufeinander folgen können. Diese hier erkannten wichtigen Eigenschaften der Zwillinge mod 4 geben den Anlass, die invarianten Eigenschaften der Zwilling mod 6, mod 8 und mod 10 näher zu analysieren. Diesem Ziel ist das nachfolgende **Kapitel 12** gewidmet.

12. Die tieferen Ursachen für das Auftreten von Mersenne- und Fermat-Primzahlen:

In **Kapitel 9** wurde der Beweis erbracht, dass Mersenne- und Fermat-Primzahlen (M_p und F_p) nur in den folgenden Dirichlet-Progressionen auftreten können:

M_p: *Teilfolge 1*: = $2^{1+4n}-1$, EZ1 und *Teilfolge 3* = $2^{3+4n}-1$, EZ7

F_p: *Teilfolge 4*: = $2^{4+4n}+1$, EZ7

Die bisher gefundenen (42) Primzahlen vom Typ M_p und die (5) Primzahlen vom Typ F_p bestätigen diesen Sachverhalt, wobei die Primzahlen: $M_p = 2^2 - 1 = 3$, EZ3 und $F_p = 2^2 + 1 = 5$, EZ5, die den Zwilling mod 2 (3, 5) bilden, nur solitär in den Endziffern: EZ3, EZ5 auftreten. Alle folgenden Primzahlen: M_p können nur in den Endziffern: EZ1 oder EZ7 auftreten und alle Primzahlen F_p ausschließlich in der Endziffer EZ7. Nachdem alle geraden Zahlen der Folge $n_g = 2^n$, bis auf die Primzahl $p = 2$ aus keinen weiteren Primzahl-Faktoren gebildet werden, treten im Umfeld dieser Zahlen $n_g = 2^n$ stets relativ hohe Primzahl-Dichten auf - siehe **Tab. 16**.

Die Zahlen $n_g = 2^n$ sind dabei stets symmetrisch oder asymmetrisch inmitten von Zwillingen mod 2n, n > 1 situiert. Sofern die Zahlen $n_g = 2^n$ extrem "links-asymmetrisch" in einem Zwilling mod 2n, n>1 (p_1, p_2) situiert sind, ergibt sich eine Mersenne - Primzahl: $p_1 + 1 = 2^n$ bzw. $p_1 = 2^n - 1 = M_p$.

Sofern eine Zahl $n_g = 2^n$ wiederum extrem "rechts-asymmetrisch" in einem Zwilling mod 2n, n>1 (p_1,p_2) situiert ist, ergibt sich eine Fermat - Primzahl: $p_2 - 1 = 2^n$ bzw. $p_2 = 2^n + 1 = F_p$.

Sofern eine Zahl $n_g = 2^n$ "maximal-symmetrisch" inmitten eines Zwillings mod 2n, n>1, (p_1, p_2) situiert ist, ergeben sich die Beziehungen: $2^n - n_u = p_1$ und $2^n + n_u = p_2$. Aus diesen Beziehungen folgt, dass die Primzahlen (p_1, p_2) dann im Abstand: $2 \times n_u + 1$ auftreten müssen.

Aus **Tab. 39** kann entnommen werden, dass diese Bedingungen nur in Zwillingen mod 6+4n, n:= 1, 2, 3, ... erfüllt werden können. Man beachte, dass nur für Zwillinge dieser Modul - Werte *Folgen* mit gerad- zahligen konstanten Gliedern auftreten. Hier vier Beispiele, die aus **Tab. 38** entnommen wurden:

Beispiel 1: In Zwillingen mod 6 treten die geraden Zahlen n_g := (10, 14, 16) + 10n inmitten situiert auf.

Beispiel 2: In Zwillingen mod 10 treten die geraden Zahlen n_g .= (6, 8, 12) + 10n inmitten situiert auf.

Beispiel 3: In Zwillingen mod 14 treten die geraden Zahlen n_g := (10, 14, 16) + 10n inmitten situiert auf.

Beispiel 4: In Zwillingen mod 18 treten die geraden Zahlen n_g .= (10, 12, 18) + 10n inmitten situiert auf.

Immer nur dann, wenn die gerade Zahl $n_g = 2^n$ mit einer geraden Zahl dieser *Folgen* identisch ist, ist diese Zahl symmetrisch inmitten eines Zwillings mod 6 +4n situiert. Hierzu drei Beispiele:

Beispiel 1: $n_g = 2^6 = 64$ ist symmetrisch inmitten des Zwillings mod 6: (61, 67) situiert.

Beispiel 2: $n_g = 2^{12} = 4096$ ist symmetrisch inmitten des Zwillings mod 6: (4093, 4099) situiert.

Beispiel 3: $n_g = 2^{45} = 35184372088832$ ist symmetrisch inmitten des Zwillings mod 22 = 6 +4n = 6 + 16: (3518437208821, 3518437208843) situiert.

Tab. 39 Die Anordnung (Positionierung) der Zahlen $n_g = 2^3$ bis 2^{80} innerhalb von Zwillingen mod 2n, n >1:

2^n :=	Δ_1	Δ_2	$\Sigma\Delta$:=	Zwilling mod:=	*Folge:=*
$2^3 = 8$	-1	+3	4	Zwilling mod 4	3
$2^4 = 16$	-3	+1	4	Zwilling mod 4	4
$2^5 = 32$	-1	+5	6	Zwilling mod 6	1
$2^6 = 64$	-3 ----------------	+3	6	Zwilling mod 6	2
$2^7 = 128$	-1	+3	4	Zwilling mod 4	3
$2^8 = 256$	-5	+1	6	Zwilling mod 6	4
$2^9 = 512$	-3	+9	12	Zwilling mod 12	1
$2^{10} = 1024$	-3	+7	10	Zwilling mod 10	2
$2^{11} = 2048$	$\underline{-9}$	+5	14	Zwilling mod 14	3
$2^{12} = 4096$	-3 ----------------	+3	6	Zwilling mod 6	4
$2^{13} = 8192$	-1	+17	18	Zwilling mod 18	1
$2^{14} = 16384$	-3	+27	30	Zwilling mod 30	2
$2^{15} = 32768$	-19	+3	22	Zwilling mod 22	3
$2^{16} = 65536$	-15	+1	16	Zwilling mod 16	4
$2^{17} = 131972$	$\underline{-3}$	+29	32	Zwilling mod 32	1
$2^{18} = 262144$	-5	+3	8	Zwilling mod 8	2
$2^{19} = 524288$	-1	+21	22	Zwilling mod 22	3
$2^{20} = 1948576$	-5	+25	30	Zwilling mod 30	4
$2^{21} = 2097152$	-9	+17	26	Zwilling mod 26	1
$2^{22} = 4194304$	-3	+15	18	Zwilling mod 18	2
$2^{23} = 8388608$	-15	+9	24	Zwilling mod 24	3
$2^{24} = 16777216$	-3	+43	46	Zwilling mod 46	4
$2^{25} = 33554432$	-39	+35	74	Zwilling mod 74	1
$2^{26} = 61798864$	-33	+7	40	Zwilling mod 40	2
$2^{27} = 134217728$	-39	+29	68	Zwilling mod 68	3
$2^{28} = 268435456$	-57	+3	60	Zwilling mod 60	4
$2^{29} = 5368709112$	-3	+11	14	Zwilling mod 14	1

Tab. 39 - Fortsetzung	Δ_1	Δ_2	$\Sigma\Delta.$=	Zwilling mod:=	*Folge:=*
2^{30} = 1073741824	-35	+3	38	Zwilling mod 38	2
2^{31} = 2147483648	-1	+11	12	Zwilling mod 12	3
2^{32} = 4294967296	-5	+15	20	Zwilling mod 20	4
2^{33} = 8589934592	-9	+17	26	Zwilling mod 26	1
2^{34} = 17179869184	-41	+25	66	Zwilling mod 66	2
2^{35} = 34359738368	-31	+53	84	Zwilling mod 84	3
2^{36} = 68719476436	-5	+31	36	Zwilling mod 36	4
2^{37} = 137438953472	<u>-25</u>	+9	34	Zwilling mod 34	1
2^{38} = 274877906944	-45	+7	52	Zwilling mod 52	2
2^{39} = 549755813888	-7	+23	30	Zwilling mod 30	3
2^{40} = 1099511627776	-87	+15	102	Zwilling mod 102	4
2^{41} = 2199023255552	<u>-21</u>	+27	48	Zwilling mod 48	1
2^{42} = 4398046511104	-11	+15	26	Zwilling mod 26	2
2^{43} = 8796093022208	<u>-57</u>	+29	88	Zwilling mod 88	3
2^{44} = 17592186044416	-17	+7	24	Zwilling mod 24	4
2^{45} = 35184372088832	-11 ---------------	+11	22	Zwilling mod 22	1
2^{46} = 70368744177664	-21	+15	36	Zwilling mod 36	2
2^{47} = 140737488355328	<u>-115</u>	+5	120	Zwilling mod 120	3
2^{48} = 281474976710656	-59	+7	66	Zwilling mod 66	4
2^{49} = 562949953421312	-81	+69	150	Zwilling mod 150	1
2^{50} = 1125899906842624	-27	+55	82	Zwilling mod 82	2
2^{51} = 2251799813685248	-129	+21	150	Zwilling mod 150	3
2^{52} = 4503599627370496	-47	+21	68	Zwilling mod 68	4
2^{53} = 9007199254740992	<u>-111</u>	+5	116	Zwilling mod 116	1
2^{54} = 18014398509481984	-33	+85	118	Zwilling mod 118	2
2^{55} = 36028797018963968	-55	+3	58	Zwilling mod 58	3
2^{56} = 72057594037927936	-5	+81	86	Zwilling mod 86	4
2^{57} = 144115188075855872	-19	+9	28	Zwilling mod 28	!

Tab. 39 - Fortsetzung	Δ_1	Δ_2	$\Sigma\,\Delta$:=	Zwilling mod:=	*Folge:=*
2^{58} = 288230376151711744	-27	+45	72	Zwilling mod 72	2
2^{59} = 576460752303423488	-57	+81	138	Zwilling mod 138	3
2^{60} = 1152921504606846976	-77	+33	110	Zwilling mod 110	4
2^{61} = 2305843009213693952	-1	+15	16	Zwilling mod 16	1
2^{62} = 4611686018427387904	-27	+25	52	Zwilling mod 52	2
2^{63} = 9223372036854775808	-25	+9	34	Zwilling mod 34	3
2^{64} = 18446744073709551616	-39	+3	42	Zwilling mod 42	4
2^{65} = 36893488147419103232	-49	+5	54	Zwilling mod 54	1
2^{66} = 73786976294838206464	-5	+7	12	Zwilling mod 12	2
2^{67} = 147573952589676412928	-19	+3	22	Zwilling mod 22	3
2^{68} = 295147905179352825856	-5	+33	38	Zwilling mod 38	4
2^{69} = 590295810358705651712	-19	+29	48	Zwilling mod 48	1
2^{70} = 1180591620717411303424	-15	+25	40	Zwilling mod 40	2
2^{71} = 2361183241434822606848	-159	+11	170	Zwilling mod 170	3
2^{72} = 4722366482869645213696	-23	+15	38	Zwilling mod 38	4
2^{73} = 9444732965739290427392	-49	+9	58	Zwilling mod 58	1
2^{74} = 18894659314785808547 84	-15 --------------	+15	30	Zwilling mod 30	2
2^{75} = 37778931862957161709568	-31	+21	52	Zwilling mod 52	3
2^{76} = 75557863725914323419136	-15 --------------	+15	30	Zwilling mod 30	4
2^{77} = 151115727451828646838272	-33	+11	44	Zwilling mod 44	1
2^{78} = 302231454903657293676544	-11	+7	18	Zwilling mod 18	2
2^{79} = 604462909807314587353088	-7	+23	30	Zwilling mod 30	3
2^{80} = 1208925819614629174706176	-39	+5	44	Zwilling mod 44	4

Anmerkungen und Schlussfolgerungen:

1. Die Teilfolgen 2 und 4 bilden bezüglich der Mersenne-Primzahlen inaktive Dirichlet-Progressionen, hingegen bilden die Teilfolgen 1 und 3 aktive Dirichlet-Progressionen, in denen unendlich viele Mersenne-Primzahlen auftreten können. Die primen Zahlen wurden hier fett gedruckt hervorgehoben.. Die Mersenne-Primzahlen können wie ersichtlich nur in den Endziffern: EZ1 und EZ7 auftreten.

2. Bei der Suche nach weiteren Mersenne-Primzahlen sind deshalb auch die Zahlen: $n_g - 1 = 2^{1+4n} - 1$ und $n_g - 1 = 2^{3+4n} - 1$ einzubeziehen, deren Exponenten nicht prime Zahlen sind.

3. Die von Mersenne vorgegebene Restriktion: $M_p = 2^p - 1$ ist deshalb aufzugeben.

4. Aus **Tab. 39** ist ersichtlich, dass die Zahlen $n_g = 2^n$ überwiegend asymmetrisch zwischen den benachbarten Primzahlen (den Zwillingen mod 2n, n > 1) situiert sind. Von den 78 Zahlen $n_g = 2^n$, die in **Tab. 39** angeführt wurden, sind nur die fünf Zahlen: $n_g := 2^5,\ 2^{11},\ 2^{45},\ 2^{74},\ 2^{76}$ (ideal) symmetrisch in den Zwillingen mod 6, mod 22 und mod 30 situiert. Diese benachbarten Primzahlen wurden in der **Tab. 39** durch gestrichelte Linien verbunden.

Die Schlussfolgerung:

In **Tab. 39** wurde bestätigt, dass Mersenne-Primzahlen in zwei Dirichlet-Progressionen: vom Typ: $M_p = 2^{1+4n} - 1$ und $M_p = 2^{3+4n} - 1$ vorkommen und somit in einer unendlichen Zahl auftreten. Wiewohl die Fermat-Primzahlen nur in der Dirichlet-Progression vom Typ: $F_p = 2^{4+4n} + 1$ vorkommen kann, treten auch diese Primzahlen (F_p) in einer unendlichen Zahl auf, was bisher nicht bewiesen werden konnte. Nachdem dieser Beweis erbracht werden konnte, empfiehlt es sich die "GIMPs-Teste" für die Suche nach weiteren Mersenne-Primzahlen zu erweitern, um auch alle Zahlen vom Typ: $2^{1+4n} - 1$ und $2^{3+4n} - 1$ auf ihre Primzahl-Eigenschaften testen zu können. Des weiteren gilt die Empfehlung die GIMPs-Teste so zu erweitern, dass auch alle Zahlen vom Typ: $2^{4+4n} + 1$ auf ihre Primzahl - Eigenschaft getestet werden können und nicht nur die Zahlen der Untermenge: $2^{2^n}+1$.
Die vom Verfasser mit dem MuPAD- Test vorgenommenen Prüfungen ergaben, dass zwischen der Fermat-Primzahl $F_p = 2^{16}+1$ und der Zahl 2^{65544} keine weiteren Primzahlen vom Typ F_p situiert sind. Die nachfolgenden Primzahlen F_p sind daher in wesentlich höheren Zahlen-Intervallen situiert.
Danach gilt die Empfehlung die "GIMPs-Teste" zu "GIM&FPs-Testen" zu erweitern und sich dabei von den "unzeitgemäßen" Restriktionen zu befreien, die Mersenne und Fermat eingeführt hatten.

Eine Zusammenfassung:
Die in diesem Kapitel erzielten Ergebnisse können wie folgt zusammen gefasst werden:

1. Es wurde der Beweis erbracht, dass die Mersenne- und Fermat-Primzahlem M_p und F_p in einer unendlichen Zahl auftreten.

2. Es wurde nachgewiesen, dass alle Mersenne-Primzahlen nur in den Endziffern: EZ1 und EZ7 auftreten können, ausgenommen die Primzahl $M_p = 2^2 - 1 = \mathbf{3}$.

   Dieser Beweis findet seine Bestätigung in **Tab. 28**, in der die Endziffern aller bisher gefundenen Mersenne-Primzahlen angeführt wurden.

3. Es wurde unter Beweis gestellt, dass alle Fermat-Primzahlen nur in der Endziffern: EZ7 auftreten

4. Aus den Endziffern: EZ1 und EZ7 in denen die Mersenne- Primzahlen (M_p) auftreten und aus der Endziffer: EZ7 der Fermat-Primzahlen (F_p) ergibt sich (zwangsweise), dass die Primzahlen F_p und M_p niemals Zwillinge mod 2 bilden können. Der tiefere Grund für diesen (invarianten) Sachverhalt wird in **Tab. 40** offenkundig.

Tab. 40 - Die Primzahl-Dichten in den Umgebungen der Zwillinge mod 2, die in den Folgen $n_g := 12 + n \times 30$, $n_g = 18 + n \times 30$ und $n_g = 30 + n \times 30$ auftreten:

$n_g = \Pi p_i :=$	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta :=$
$n_g = 12 = 2^2 \cdot 3$	-7 --------	-5	-1 -------	+1	+5 ------	+7	14
$n_g = 18 = 2^2 \cdot 3$	-7 --------	-5	-1 -------	+1	+5	+11	18
$n_g = 30 = 2 \cdot 3 \cdot 5$	-11	-7	-1 -------	+1	+7	+11	22

Tab. 40 - Fortsetzung

	Δ_1	Δ_2	Δ_3	Δ_4	Δ_5	Δ_6	$\Sigma\Delta:=$
$n_g = 42 = 2 \cdot 3 \cdot 7$	-11	-5	-1 -------	+1	+5	+11	22
$n_g = 60 = 2^2 \cdot 3 \cdot 5$	-13	-7	-1 -------	+1	+7	+11	24
$n_g = 72 = 2^3 \cdot 3^2$	-11	-5	-1 -------	+1	+7	+11	22
$n_g = 102 = 2 \cdot 3 \cdot 17$	-13	-5	-1 -------	+1	+5 ------	+7	20
$n_g = 108 = 2^2 \cdot 3^3$	-7 ------	-5	-1 -------	+1	+5 ------	+7	26
$n_g = 138 = 2 \cdot 3 \cdot 23$	-25	-11	-1 -------	+1	+11 -----	+13	38
$n_g = 150 = 2 \cdot 3 \cdot 5^2$	-13 -----	-11	-1 -------	+1	+7	+13	26
$n_g = 180 = 2^2 \cdot 3^2 \cdot 5$	-13	-7	-1 -------	+1	+11 -----	+13	26
$n_g = 192 = 2^6 \cdot 3$	-11 -----	-9	-1 -------	+1	+7 ------	+9	20
$n_g = 228 = 2^2 \cdot 3 \cdot 19$	-17	-5	-1 -------	+1	+5	+11	28
$n_g = 270 = 2 \cdot 3^3 \cdot 5$	-13	-7	-1 -------	+1	+7	+11	24
$n_g = 282 = 2 \cdot 3 \cdot 47$	-11	-5	-1 -------	+1	+11	+25	36
$n_g = 24.108 = 2^2 \cdot 3 \cdot 7^2 \cdot 41$	-11	-5	-1 -------	+1	+5	+13	24
$n_g = 204.300 = 2^2 \cdot 3^2 \cdot 5^2 \cdot 227$	-67	-47	-1 -------	+1	+11	+19	86
$n_g = 1.293.939.168 = 2^5 \cdot 3 \cdot 29 \cdot 464777$	-55	-7	-1 -------	+1	+35	+43	98
$n_g = 15.351.613.702 = 2 \cdot 3 \cdot 7 \cdot 19 \cdot 1923749$	-61	-25	-1 -------	+1	+17	+61	102

Anmerkungen und Schlussfolgerungen:

1. Aus **Tab. 40** ergibt sich zunächst die Bestätigung, dass die Mersenne- und Fermat-Primzahlen: M_p und F_p ab dem Zwilling mod 2 (3, 5) = ($M_p = 2^2 - 1$, $F_p = 2^2 + 1$) keine weiteren Zwillinge mod 2 bilden können. Dieses hier gefundene Gesetz findet in **Tab. 40** eine volle Bestätigung. da alle geraden Zahlen, die inmitten der Zwillinge mod 2 situiert sind, aus Primzahl-Fakultäten vom Typ: $n_g = \Pi p_i$ bestehen.

2. Die Mersenne- und Fermat-Primzahlen (M_p und F_p) treten dagegen in den (direkten) Umgebungen der geraden Zahlen $n_g = 2^n$ auf. In diesen geraden Zahlen ($n_g = 2^n$) treten (fakultativ) keine ungeraden Primzahlen auf.

---

13. Die Bestimmung großer Primzahlen $p = 2^n \pm 1$, $p = 2^n \pm 3$, $p = 2^n \pm 5$, $2^n \pm 7$, $p = 2^n \pm 9$ und $p = 2^n \pm 11$ mit dem MuPAD-Test:

Nachdem aus **Tab. 39** entnommen werden kann, dass Primzahlen in den (direkten) Umgebungen der geraden Zahlen $n_g = 2^n$ überwiegend in Abständen $\Delta > 3, 5, 7, 9, 11, \ldots$ auftreten und die Abstände $\Delta = \pm 1$ nur selten auftreten, lassen sich extrem große Primzahlen nach dem MuPAD-Test bestimmen. In **Tab. 41, 42, 43, 44, 45, 46** wurden die Primzahlen $p = 2^n \pm 1$, $p = 2^n \pm 3$, $p = 2^n \pm 5$, $p = 2^n \pm 7$, $p = 2^n \pm 9$ und $p = 2^n \pm 11$ für die Exponenten n = 4 bis n = 4256 bestimmt, die in den Folgen: *Folge 1*, *Folge 2*, *Folge 3*, *Folge 4* auftreten.

Für die Bestimmung der 19 Primzahlen, die in **Tab. 41** vorgestellt wurden, ist bei Einsatz der MuPAD - Testmethode ein Zeitaufwand von (etwa) zwei Stunden erforderlich. Für die Ermittlung der ersten 16 Mersenne- Primzahlen, die in **Tab. 41** angeführt sind, benötigten die Mathematiker dagegen eine Zeit von rd. 500 Jahren und Euler fand die Primzahl $M_{p,8} = 2^{31} - 1$ im Jahr 1771.

Tab. 41 Die Primzahlen : $p = 2^n - 1$ und $p = 2^n + 1$ im Exponenten-Bereich $4 \leq n \leq 4256$

Folge 1:	*Folge 2:*	*Folge 3:*	*Folge 4:*
2^5 - 1, EZ1	$2^{2+4n}-1$, EZ3 QS durch 3 teilbar	2^7 -1, EZ7	$2^{4+4n}-1$, EZ5
2^13 -1, EZ1		2^19 -1, EZ7	
2^61 -1, EZ1		2^31 -1, EZ7	
2^89 -1, EZ1		2^107 -1, EZ7	
2^521 -1, EZ1		2^127 -1, EZ7	
2^2281 -1, EZ1		2^607 -1, EZ7	
2^3217 -1, EZ1		2^1279 -1, EZ7	
2^4253 -1, EZ1		2^2203 -1, EZ7	

Anmerkungen:

1. Die *Folge 2* und *Folge 4* sind Primzahl - inaktive Folgen: $2^n - 1$
2. In den Folgen: *Folge 1* und *Folge 3* treten im Exponenten-Bereich $4 \leq n \leq 4256$ 15 Primzahlen auf

---

Folge 1:	*Folge 2:*	*Folge 3:*	*Folge 4;*
$2^{1+4n}+1$, EZ3 QS durch 3 teilbar	$2^{2+4n}+1$, EZ5	$2^{3+4n}+1$, EZ9 QS durch 3 teilbar	2^4 +1 = 17, EZ7
			2^8 +1 = 257, EZ7
			2^16 +1 = 65537, EZ7

Anmerkungen:

1. Die Folgen: *Folge 1*, *Folge 2* und *Folge 3* : $2^n + 1$ sind Primzahl - inaktiv.
2. In *Folge 4* treten im Exponenten-Bereich $4 \leq n \leq 4096$ drei Primzahlen $p = 2^n + 1$ auf.
3. Mit dem MuPAD - Test wurde geprüft, dass in *Folge 4* bis $n \leq 65537$ keine weiteren Primzahlen $2^n + 1$ auftreten.

Tab. 42 Die Bestimmung der Primzahlen $p = 2^n - 3$ und $p = 2^n + 3$ für die Exponenten: $4 \leq n \leq 4256$:

Folge 1:	*Folge 2:*	*Folge 3:*	*Folge 4:*
2^5 -3. EZ9	2^6 -3, EZ1	$2^{3+4n}-3$, EZ5	2^4 -3, EZ3
2^9 -3, EZ9	2^10 -3, EZ1		2^12 -3, EZ3
2^29 -3, EZ9	2^14 -3, EZ1		2^20 -3, EZ3
2^213 -3, EZ9	2^22 -3, EZ1		2^24 -3, EZ3

Tab. 42 - Fortsetzung

Folge 1:	*Folge 2:*	*Folge 3:*	*Folge 4:*
2^221 -3, EZ9	2^94 -3, EZ1		2^116 -3, EZ3
2^233 -3, EZ9	2^122 -3, EZ1		2^336 -3, EZ3
2^266 -3, EZ9	2^150 -3, EZ1		2^452 -3, EZ3
2^545 -3, EZ9	2^174 -3, EZ1		2^1736 -3, EZ3
2^689-3, EZ9	2^266 -3, EZ1		
2^2321 -3, EZ9	2^694-3, EZ1		
2^3237 -3, EZ9	2^850-3, EZ1		
	2^3954 -3, EZ1		

Anmerkungen:

1. Die *Folge 3*: $2^{3+4n}-3$, EZ5 ist für $n>0$ Primzahl - inaktiv.
2. Im "Exponenten-Bereich": $4 \leq n \leq 4256$ treten insgesamt 31 Primzahlen $p = 2^n - 3$ auf.

Folge 1:	*Folge 2:*	*Folge 3:*	*Folge 4:*
$2^{1+4n}+3$, EZ5	2^6 +3, EZ7	2^7 +3, EZ1	2^4 +3, EZ9
	2^18 +3, EZ7	2^15 +3, EZ1	2^12 +3, EZ9
	2^30 +3, EZ7	2^55 +3, EZ1	2^16 +3, EZ9
	2^390 +3, EZ7	2^67 +3, EZ1	2^28 +3, EZ9
	2^1110 +3, EZ7	2^2139 +3, EZ1	2^84 +3, EZ9
	2^2008 +3, EZ7	2^2191 +3, EZ1	2^228 +3, EZ9
	2^2370 +3, EZ7	2^2367 +3, EZ1	2^784 +3, EZ9
	2^4002 +3, EZ7		2^1704 +3, EZ9
	2^4062 +3, EZ7		2^4060 +3, EZ9

Anmerkungen:

1. Die *Folge 1*: $2^{1+4n}+3$. EZ5 ist für $n>0$ Primzahl-inaktiv..
2. Im "Exponenten-Bereich" $4 \leq n \leq 4256$ treten insgesamt 25 Primzahlen $p = 2^n + 3$ auf.

Tab. 43 Die Bestimmung der Primzahlen $p = 2^n - 5$ und $p = 2^n + 5$ für die Exponenten $4 \leq n \leq 4256$:

Folge 1:	*Folge 2:*	*Folge 3:*	*Folge 4:*
$2^{1+4n}-5$, EZ7 QS teilbar durch 3	2^6 -5, EZ9	$2^{3+4n}-5$, EZ5	2^4 -5, EZ1
	2^10 -5, EZ9		2^8 -5, EZ1
	2^18 -5, EZ9		2^12 -5, EZ1
	2^26 -5, EZ9		2^20 -5, EZ1
	2^66 -5, EZ9		2^32 -5, EZ1

Tab. 43 - Fortsetzung

Folge 1:	*Folge 2:*	*Folge 3:*	*Folge 4:*
	2^118 -5, EZ9		2^36 -5, EZ1
	2^130 -5, EZ9		2^56 -5, EZ1
	2^150 -5, EZ9		2^1136 -5, EZ1
	2^166 -5, EZ9		2^1228 -5, EZ1
	2^206 -5, EZ9		2^2368 -5, EZ1
	2^226 -5, EZ9		2^2400 -5, EZ1
	2^550 -5, EZ9		2^3128 -5, EZ1
	2^706 -5, EZ9		
	2^810 -5, EZ9		
	2^1818 -5, EZ9		

Anmerkungen:

1. Die *Folge 1*, EZ7 ist für $n \geq 1$ Primzahl - inaktiv, da alle Zahlen $2^{1+4n} -5$ durch die Zahl 3 teilbar sind.
2. Die *Folge 3*, EZ5 ist für $n > 1$ Primzahl - inaktiv.
3. Im Exponenten-Bereich $4 \leq n \leq 4256$ treten 27 Primzahlen $p = 2^n - 5$ auf.

Folge 1:	*Folge 2:*	*Folge 3:*	*Folge 4:*
2^5 +5, EZ7	$2^{2+4n}+5$, EZ9 QS teilbar durch 3	2^11 +5, EZ3	$2^{4+4n}+5$, EZ1 QS Teilbar durch 3
2^53 +5, EZ7		2^47 +5, EZ3	
2^141 +5, EZ7		2^143 +5, EZ3	
2^273 +5, EZ7		2^191 +5, EZ3	
2^341 +5, EZ7			

Anmerkungen:

1. *Folge 2* und *Folge 4* sind Primzahl - inaktiv, da die Quersummen der Zahlen $2^{2+4n}+5$ und $2^{4+4n}+5$ durch 3 teilbar.
2. In den Folgen: *Folge 1* und *Folge 3* treten im Exponenten-Bereich $4 \leq n \leq 4256$ insgesamt neun Primzahlen vom Typ: $2^n +5$ auf.

Die Schlussfolgerungen aus **Tab. 41**, **Tab. 42** und **Tab. 43**:

1. Aus **Tab. 41, Tab. 42** und **Tab. 43** ergibt sich die Schlussfolgerung, dass in den Folgen $2^n \pm 1$, $2^n \pm 3$ und $2^n \pm 5$ für $n \to \infty$ unendlich viele Primzahlen auftreten.

2. Aus diesen Tabellen ist zu entnehmen welche der Folgen: *Folge 1, Folge 2, Folge 3, Folge 4* in Abhängigkeit von den addierten bzw. subtrahierten Zahlen (±1, ±3, ±5 Primzahl - inaktiv sind.

3. In den vorgestellten Tabellen wurden die Ursachen offen gelegt, die zu "Primzahl - Inaktivitäten" bei einzelnen Folgen führen. - in diesen Folgen treten entweder nicht prime Zahlen mit der Endziffer EZ5 auf oder aber ungerade Zahlen mit der durch die Zahl 3 teilbaren Quersummen (QS).

4. Aus den nachfolgenden Tabellen ist zu entnehmen, dass abhängig von den (addierten bzw. subtrahierten) Zahlen $n_u := \pm 7, \pm 9, \pm 11$ (und so weiter) einzelne Folgen Primzahl - inaktiv werden.

Tab. 44 Die Bestimmung der Primzahlen $p = 2^n - 7$ und $p = 2^n + 7$ für die Exponenten $4 \leq n \leq 4256$

Folge 1:	*Folge 2:*	*Folge 3:*	*Folge 4:*
$2^{1+4n} - 7$, EZ5	$2^{2+4n} - 7$, EZ7 QS durch 3 teilbar	2^715 -7, EZ1	$2^{4+4n} - 7$, EZ9 QS durch 3 teilbar
		2^1983 -7, EZ1	
		2^2319 -7, EZ1	
		2^2499 -7, EZ1	
		2^3775 -7, EZ1	

Anmerkungen:

1, Die Folgen: *Folge 1, Folge 2 und Folge 4* sind für alle Zahlen $2^n - 7$ Primzahl - inaktiv

2. In *der Folge* 3 treten im Exponenten-Bereich $4 \leq n \leq 4256$ fünf Primzahlen $p = 2^n - 7$, EZ1 auf.

Folge 1:	*Folge 2:*	*Folge 3:*	*Folge 4:*
$2^{1+4n} + 7$, EZ9 QS durch 3 teilbar	2^6 +7, EZ1	$2^{3+4n} + 7$, EZ5	2^4 +7, EZ3
	2^10 +7, EZ1		2^8 +7, EZ3
	2^18 +7, EZ1		2^16 +7, EZ3
	2^30 +7, EZ1		2^20 +7, EZ3
	2^38 +7, EZ1		2^28 +7, EZ3
	2^78 +7, EZ1		2^44 +7, EZ3
	2^98 +7, EZ1		2^88 +7, EZ3
	2^126 +7, EZ1		2^160 +7, EZ3
	2^174 +7, EZ1		2^204 +7, EZ3
	2^214 +7, EZ1		2^588 +7, EZ3
	2^610+7, EZ1		2^1888 +7, EZ3
	2^798 +7, EZ1		2^2648 +7, EZ3
	2^926 +7, EZ1		2^3864 +7, EZ3
	2^1190 +7, EZ1		
	2^1198 +7, EZ1		
	2^1806 +7, EZ1		
	2^3454 +7, EZ1		
	2^3510 +7, EZ1		
	2^3870 +7, EZ1		

Anmerkungen:

1. Die *Folge 1* und *Folge 3* sind Primzahl - inaktive Folgen, da die Zahlen $2^n + 7$ dieser Folgen nicht prim sein können

2. In *Folge 2* und *Folge 4* treten im Exponenten-Bereich $4 \leq n \leq 4256$ insgesamt 32 Primzahlen vom Typ: $p = 2^n + 7$ auf.

3. In der unendlichen Folge der Exponenten $(n \rightarrow \infty)$ treten daher unendlich viele Primzahlen $p = 2^n + 7$ auf.

Tab. 45 Die Bestimmung der Primzahlen $p = 2^n -9$ und $p = 2^n +9$ für die Exponenten $4 \leq n \leq 4256$:

Folge 1;	*Folge 2:*	*Folge 3:*	*Folge 4:*
2^5 -9, EZ3	2^{2+4n} -9, EZ5	2^11 -9, EZ9	2^4 -9, EZ7
2^9 -9, EZ3		2^243 -9, EZ9	
2^17 -9, EZ3		2^251 -9, EZ9	
2^21 -9, EZ3		2^563 -9, EZ9	
2^33 -9, EZ3		2^699 -9, EZ9	
2^125 -9, EZ3			
2^141 -9, EZ3			
2^285 -9, EZ3			
2^321 -9, EZ3			
2^537 -9,. EZ3			
2^729 -9, EZ3			
2^2841 -9, EZ3			
2^3365 -9, EZ3			

Anmerkungen:

1. Die *Folge 2* ist Primzahl - inaktiv, da alle Zahlen $2^n -9$ die EZ5 aufweisen.
2. In den Folgen: *Folge 1*, *Folge 2* und *Folge 4* treten insgesamt 19 Primzahlen auf.
3. In der *Folge 4* tritt im Exponenten-Bereich $4 \leq n \leq 4256$ nur die Primzahl $p = 2^4 -9 = 7$ auf. Dies schließt nicht aus, dass sich für höhere Exponenten Primzahlen des Typs: $p = 2^{4+4n} -9$ ergeben.

Folge 1:	*Folge 2:*	*Folge 3:*	*Folge 4:*
2^5 +9,. EZ1	2^6 +9, EZ3	2^7 +9, EZ7	2^{4+4n} +9, EZ5
2^9 +9, EZ1	2^10 +9, EZ3	2^23 +9, EZ7	
2^37 +9, EZ1	2^18 +9, EZ3	2^47 +9, EZ7	
2^57 +9, EZ1	2^30 +9, EZ3	2^95 +9, EZ7	
2^317 +9, EZ1	2^66 +9, EZ3	2^119 +9, EZ7	
2^697 +9, EZ1	2^670 +9, EZ3	2^175 +9, EZ7	
2^1717 +9, EZ1	2^886 +9, EZ3	2^263 +9, EZ7	
2^3229 +9, EZ1	2^1342 +9, EZ3	2^295 +9, EZ7	
2^3253 +9, EZ1	2^2394 +9, EZ3	2^319 +9. EZ7	
2^3749 +9, EZ1	2^2710 +9, EZ3	2^327 +9, EZ7	
		2^1855 +9, EZ7	

Anmerkungen:

1. Die *Folge 4:* 2^{4+4n} +9, EZ5 ist Primzahl - inaktiv
2. Im Exponenten-Bereich $4 \leq n \leq 4256$ treten 30 Primzahlen $p = 2^n +9$ auf.

Tab. 46 Die Bestimmung der Primzahlen $p = 2^n - 11$ und $p = 2^n + 11$ für die Exponenten $4 \leq n \leq 4256$

Folge 1:	*Folge 2:*	*Folge 3:*	*Folge 4:*
$2^{1+4n}-11$, EZ1 QS durch 3 teilbar	2^6 -11, EZ3	$2^{3+4n}-11$, EZ7 QS durch 3 teilbar	$2^4 - 11 = 5$, EZ5
	2^10 -11, EZ3		
	2^18 -11, EZ3		
	2^42 -11, EZ3		
	2^78 -11, EZ3		
	2^94 -11, EZ3		
	2^114 -11, EZ3		
	2^190 -11, EZ3		
	2^322 -11, EZ3		
	2^546 -11, EZ3		
	2^3894 -11, EZ3		

Anmerkungen:

1, *Folge 1* und *Folge 3* sind Primzahl - inaktiv, da alle Zahlen $2^n - 11$ dieser Folgen durch 3 teilbar sind,

2. In der *Folge 4* ist die Zahl $p = 2^4 - 11 = 5$ eine solitäre Primzahl.

3. Im Exponenten-Bereich $4 \leq n \leq 4256$ treten insgesamt 11 Primzahlen $p = 2^n - 11$ auf.

Folge 1;	*Folge 2:*	*Folge 3:*	*Folge 4:*
2^5 +11, EZ3	$2^{2+4n}+11$, EZ5	2^7 +11, EZ9	$2^{4+4n}+11$, EZ7 QS durch 3 teilbar
2^9 +11, EZ3		2^15 +11, EZ9	
2^29 +11, EZ3		2^23 +11, EZ9	
2^77 +11, EZ3		2^31 +11, EZ9	
2^287 +11, EZ3		2^55 +11, EZ9	
2^573 +11, EZ3		2^71 +11, EZ9	
2^1301 +11, EZ3		2^1555 +11, EZ9	
2^1661 +11, EZ3			

Anmerkungen:

1. *Folge 2*, EZ5 *und Folge 4*, EZ7 sind Primzahl - inaktive Folgen $2^n + 11$
2. Im Exponenten-Bereich $4 \leq n \leq 4256$ treten 15 Primzahlen $p = 2^n + 11$ auf.

Ein Vergleich der Menge der Primzahlen, die im Exponenten-Bereich $4 \leq n \leq 4256$ auftreten:

$2^n - 1$:
16 Primzahlen (8 x EZ1 und 8 x EZ7)
$2^n - 3$:
31 Primzahlen (12 x EZ1, 8 x EZ3, 11 x EZ9)
$2^n - 5$:
27 Primzahlen (12 x EZ1, 15 x EZ9)
$2^n - 7$:
5 Primzahlen (alle mit EZ1)
$2^n - 9$:
19 Primzahlen (13 x EZ3, 1 x EZ7, 5 x EZ9)
$2^n - 11$:
11 Primzahlen (alle mit EZ3)

$2^n + 1$:
3 Primzahlen mit EZ7
$2^n + 3$:
25 Primzahlen (7 x EZ1, 9 x EZ7, 9 x EZ9)
$2^n + 5$:
9 Primzahlen (4 x EZ3, 5 x EZ7)
$2^n + 7$.
32 Primzahlen (19x EZ1, 13 x EZ3)
$2^n + 9$:
30 Primzahlen (10 x EZ1, 10 x EZ3, 11 x EZ7)
$2^n + 11$:
15 Primzahlen (8 x EZ3, 7 x EZ9)

Der vorgestellte Vergleich der Primzahl-Mengen führt zu der Erkenntnis, dass Mersennesche und Fermatsche Primzahlen (M_p und F_p) in geringerer Häufigkeit auftreten als Primzahlen vom Typ: $p = 2^n \pm 3.\ 2^n \pm 5,\ 2^n \pm 7,\ 2^n \pm 9,\ 2^n \pm 11, \ldots$

Dieses Faktum sollte zum Anlass genommen werden, um das "Lucas-Lehmer-Testverfahren" derart zu modifizieren, dass es auch für die Suche nach extrem großen Primzahlen der Typen: $2^n \pm n_u > 1$ anwendbar ist. Da diesem Verfahren die "Restklassen - Methode" zugrunde liegt, dürften seiner Modifizierung keine besonderen Schwierigkeiten im Wege stehen. Mit einem modifizierten Lucas-Lehmer - Test kann man (unter Beachtung der Primzahl-inaktiven Folgen) die größten unbekannten Primzahlen mit den Endziffern: EZ1, EZ3, EZ7 und EZ9 bestimmen.

In **Tab. 47** werden die Zwillinge mod 2n, n >1 vorgestellt, inmitten welcher die größten Primzahlen situiert sind, die in den Tabellen **Tab. 41** bis **Tab. 46** angeführt wurden. Die Bestimmung dieser Zwillinge mod 2n, n>1 erfordert (trotz des Einsatzes der MuPAD-Testmethode) ein Zeitaufwand von mehreren Tagen.

Tab. 47 - Die Zwillinge mod 2n, n>1 in denen die Zahlen $2^n \pm 1$, $2^n \pm 3$, $2^n \pm 5$, $2^n \pm 7$, $2^n \pm 9$. $2^n \pm 11$ in den Folgen: *Folge 1, Folge 2, Folge 3* und *Folge 4* situiert sind:

$2^n - 1$:	Δ_1	Δ_2	$\Sigma\Delta$:=	$2^n + 1$:	Δ_1:	Δ_2	$\Sigma\Delta$:=
2^4253:=	**-1**	+4691	4692	2^16:=	**+1** -------	+3	2
$2^n -3$:				$2^n +3$			
2^3954	-3	+1429	1432	2^4060	-825	+3	828
$2^n -5$:				$2^n +5$:			
2^3128	-5	+841	846	2^341	-229 -	+5	334
$2^n -7$:				$2^n +7$:			
2^3775	-7	+531	538	2^3870	-143	+7	150
$2^n -9$:				$2^n +9$:			
2^3365	-9	+999	1008	2^3749	-2245	+9	2254
$2^n -11$:				$2^n +11$:			
2^3894	-11	+1059	1070	2^1661	-6583	+11	6594
2^n - x, 2^n +y:	x:=	y:=	(x+y):=	<u>Anmerkung:</u>			
2^4256	-1325	+723	2048	Letztes Beispiel der Suche nach Primzahlen in der Umgebung einer definierten Zahl $n_g = 2^n$ (hier 2^4256)			

<u>Anmerkungen und Schlussfolgerungen:</u>

1. Aus **Tab. 47** ist ersichtlich, dass die geraden Zahlen $n_g = 2^n$ überwiegend asymmetrisch innerhalb von Zwillingen mod 2n, n > 1 situiert sind.

2. Die "Modularität" der in Tab. 47 ausgewiesenen Zwillinge schwankt zwischen $\Sigma\Delta$:= 150 und 6594.

3. Aus **Tab. 47** ist ersichtlich, dass man große Primzahlen durch die Suche nach dem Exponenten (n) bei Vorgabe der ungeraden Zahlen n_u:= ±3, ±5, ... bestimmen kann oder aber (wie im letzten Beispiel von **Tab. 47** demonstriert wurde) durch die Variation der ungeraden Zahlen: n_u: = "-x" und "+y".

4. Sofern sehr große Primzahlen $p = 2^n \pm 3$, $p = 2^n \pm 5$, $p = 2^n \pm 7$, ... mit definierten Endziffern (EZ1, EZ3 EZ7, EZ9) bestimmt werden sollen, genügt es die Suche nach diesen Primzahlen in den (jeweiligen) Primzahl - aktiven Folgen (*Folge 1, Folge 2, Folge 3, Folge 4*) vorzunehmen und die inaktiven Folgen außer Acht zu lassen.

5. Nach der GIMPs - Methode können derweil (bekanntlich) nur sehr große Primzahlen mit den Endziffern EZ1 und EZ7 bestimmt werden, nicht aber Primzahlen mit den Endziffern EZ3 und EZ9.

---

14. Der Nachweis, dass für Primzahlen vom Typ $2^n \pm n_u$ Gesetz 1 und Gesetz 2 der Faktor-Zerlegung gelten:

Nach Gesetz 1 können bei der Addition bzw. Subtraktion von Zahlen, die als Faktoren in einer vorgegebenen Zahl (n_g bzw. n_u) auftreten, niemals Primzahlen generiert werden - siehe Kapitel 4. Nach Gesetz 2 können Primzahlen nur durch die Addition bzw. Subtraktion von Zahlen gewonnen werden, die in vorgegebenen Zahlen (n_g bzw. n_u) als Faktoren nicht auftreten. Des weiteren können (bekanntlich) Primzahlen vom Typ: $n_g -1$ bzw. $n_g +1$ und vom Typ: $n_u -2$ bzw. $n_u +2$ (fallweise) generiert werden. Nach Gesetz 2 erfolgt bei der Addition bzw. Subtraktion von Zahlen, die als Faktoren in den (vorgegebenen) Zahlen (n_g bzw. n_u) nicht auftreten (fallweise) eine "Verschmelzung" zu größeren Primzahl-Faktoren, sofern sich keine Primzahlen ergeben.
An einigen Beispielen soll (nun) demonstriert werden, dass Gesetz 1 und Gesetz 2 (uneingeschränkt) auch für die geraden Zahlen $n_g = 2^n$ gelten, in denen (als einziger Primzahl-Faktor) nur die Primzahl $p = \mathbf{2}$ auftritt.

Beispiel 1: Die Faktor-Zerlegung von $n_g = 2\text{^}8 = 256$ bei Addition und Subtraktion ungerader Zahlen:

$n_u = 2\text{^}8 -1 = 3 \cdot 5 \cdot 17 = 255$, $n_u = 2\text{^}8 +1 = \mathbf{257}$, $n_u = 2\text{^}8 -3 = 11 \cdot 23 = 253$

$n_u = 2\text{^}8 +3 = 7 \cdot 37 = 259$, $n_u = 2\text{^}8 -5 = \mathbf{251}$, $n_u = 2\text{^}8 +5 = 3^2 \cdot 29 = 261$

$n_u = 2\text{^}8 -7 = 3 \cdot 83 = 249$, $n_u = 2\text{^}8 +7 = \mathbf{263}$, $n_u = 2\text{^}8 - 13 = 3^5 = 243$

$n_u = 2\text{^}8 +13 = \mathbf{269}$

$n_g = 2\text{^}8 -2 = 2 \cdot 127 = 254$, $n_g = 2\text{^}8 +2 = 2 \cdot 3 \cdot 43 = 258$

$n_g = 2\text{^}8 -2^4 = 2^4 \cdot 3 \cdot 5 = 240$, $n_g = 2\text{^}8 +2^4 = 2^4 \cdot 17 = 272$

Beispiel 2: Die Faktor-Zerlegung von $n_g = 2\text{^}32 = 4294967296$ bei Addition und Subtraktion ungerader Zahlen

$n_u = 2\text{^}32 -1 = 3 \cdot 5 \cdot 17 \cdot 257 \cdot 65537 = 4294967295$, $n_u = 2\text{^}32 +1 = 641 \cdot 6700417 = 4294967297$

$n_u = 2\text{^}32 -3 = 9241 \cdot 464773 = 4294967293$, $n_u = 2\text{^}32 +3 = 7 \cdot 613566757 = 4294967299$

$n_u = 2\text{^}32 -5 = \mathbf{4294967291}$, $n_u = 2\text{^}32 +5 = 3^3 \cdot 47 \cdot 3384529 = 4294967301$

$n_u = 2\text{^}32 -15 = 11 \cdot 181 \cdot 241 \cdot 8951 = 4294967281$, $n_u = 2\text{^}32 +15 = \mathbf{4294967311}$

$n_g = 2\text{^}32 -2 = 2 \cdot 2147483647 = 4294967294$, $n_g = 2\text{^}32 +2 = 2 \cdot 3 \cdot 715827883 = 4294967298$

$n_g = 2\text{^}32 -2^4 = 2^4 \cdot 3 \cdot 5 \cdot 29 \cdot 43 \cdot 113 \cdot 127 = 4294967280$, $n_g = 2^4 \cdot 17 \cdot 15790321 = 4294967312$

Anmerkungen und Schlussfolgerungen:

1. Die in den vorgestellten Beispielen auftretenden Primzahlen wurden fett gedruckt hervorgehoben.
2. Man kann sich davon überzeugen, dass Gesetz 1 und Gesetz 2 für alle Zahlen $n_g = 2^n$ (stets) gelten.

15. Die Primzahl-Dichten im Umfeld der Fakultät $n_g = \prod p_i$ (von $p_1 = 2$ bis $p_n = 267$) im Intervall n_g -3300 bis n_g + 3300 - eine Analyse der Schwankungen der Primzahl-Dichten:

In diesem Kapitel werden die Primzahl-Dichten in einem Intervall von $\Sigma\ \Delta = 6600$ vorgestellt, die in der Umgebung der geraden Zahl $n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot ... \cdot 251$ auftreten. In diesem Intervall treten 29 Primzahlen auf, die mit dem MuPAD-Testprogramm bestimmt wurden - siehe **Anlage 4**.

In **Tab. 48** wurden diese Primzahlen mit Angabe der zwischen ihnen auftretenden Abstände (Δ_1 bis Δ_{27}) in ihrer Aufeinanderfolge wiedergegeben.

Tab. 48 - Die Folge der Primzahlen im Intervall $\pm\Delta = 3300$ im Umfeld der Zahl $n_g = \prod p_i$ ($p_1 = 2$, $p_n = 251$)

(Δ) **- 2887** , (344) **- 2543**, (12) **-2531**, (84) **- 2447**, (58) **- 2389**, (252) **- 2137**, (50) **-2087**, (208)

(Δ) **- 1879**, (156) **- 1741**, (122) **- 1619**, (958) **- 661**, (314) **- 347**, (624) **+277**, (72) **+349**, (4) **+353**,

(Δ) (14) **+367**, (16) **+383**, (318) **+701**, (96) **+797**, (16) **+881**, (96) **+977**, (306) **+1283**, (144)

(Δ) **+ 1427**, (156) **+ 1583**, (86) **+ 1669**, (360) **+ 2029**, (60) **+ 2089**, (64) **+ 2153**, (1076) **+ 3229**

Anmerkungen:

1. Alle Abstände der 29 Primzahlen von der Zahl $n_g = \prod p_i (p_1 = 2$ bis $p_n = 251)$ sind prime Zahlen und wurden in **Tab. 48** fett gedruckt.

2. Die Primzahl-Dichten unterliegen "wellenartigen" Schwankungen, was sich in den aufeinander folgenden Zwillingen mod 2n, n > 1 manifestiert.

3. Die größten Zwillinge: mod 624, mod 958 und mod 1076 wurden in **Tab. 48** unterstrichen markiert.

4. Infolge der Größe des gewählten Intervalls (±3300) können (hier) keine Primzahl-Fakultäten vom Typ: $257 \cdot 257 = 66049$, $257 \cdot 263 = 67591$, $257 \cdot 269 = 69133$ usf. auftreten - derartige Abstände können jedoch innerhalb von Intervallen $\pm\Delta = 100.000$ auftreten.

5. Der MuPAD-Test, der für die Abstände: Δ:= - 100.000 bis - 66.100 durchgeführt wurde, ergab, dass in diesem Intervall sechs Primzahlen auftreten, die in den Abständen der folgenden Primzahl-Fakultäten situiert sind:

   $-(98431 = 257 \cdot 383)$, $-(97507 = 281 \cdot 347)$, $-(96571 = 269 \cdot 359)$,
   $-(86701 = 277 \cdot 313)$, $-(81161 = 277 \cdot 293)$, $-(78961 = 281 \cdot 281)$

   In diesem Intervall treten des weiteren 126 Primzahl auf, denen Primzahl-Abstände entsprechen.

6. Der MuPAD-Test, der für die Abstände Δ = +66.100 bis + 100.000 durchgeführt wurde, ergab, dass in diesem Intervall fünf Primzahlen auftreten, die in den Abständen der folgenden Primzahl-Fakultäten situiert sind:

   $+(74429 = 263 \cdot 283)$ $+(77059 = 263 \cdot 293)$ $+(87953 = 281 \cdot 313)$
   $+(96673 = 277 \cdot 349)$ $+(98723 = 269\ \ 367)$

   In diesem Intervall treten des weiteren 129 Primzahlen auf, denen Primzahl-Abstände entsprechen.

7. Die Verteilungen der Primzahlen in diesen beiden Intervallen weisen ähnliche Schwankungen der Primzahl-Dichten auf, wie in **Tab. 48** - auf Primzahlen, die in relativ kleinen Abständen aufeinander folgen, treten (wellenartig) Primzahlen in großen Abständen auf.

8. Es ist eine volle Bestätigung von Gesetz 2 der Primzahl-Bildung zu konstatieren.

In **Tab. 49** wurde die Primzahl-Verteilung in der Umgebung der Fakultät $n_g = \Pi p_i(p_1 = 2, p_n = 257)$ für das Intervall von $\Delta = 6600$ dargestellt, in **Tab. 50** wurde für das gleiche Intervall die Primzahl-Verteilung in der Umgebung der Fakultät $n_g = \Pi p_i(p_1 = 2, p_n = 263)$ dargestellt und in **Tab. 51** für die Fakultät $n_g = \Pi p_i(p_1 = 2, p_n = 269)$. Alle diese Primzahlen, die (allesamt) stets in Primzahl-Abständen gegenüber den Fakultäten $n_g = \Pi p_i$ auftreten, wurden mit dem MuPAD-Test bestimmt,

Tab. 49 Die Folge der Primzahlen im Intervall $\pm\Delta$= 3300 im Umfeld der Zahl $n_g = \Pi p_i(p_1 = 2, p_n = 257)$

(Δ):= **-3103**, (244) **-2859**, (210) **-2511**, (224) **-2287**, (34) **-2253**, (32) **-2221**, (874) **-1347**, (420)

(Δ):= **-927**, (104) **-823**, (262) **-561**, (30) **-531**, (1190) **+659**, (1220) **+1637**, (66) **+1751**, (58)

(Δ):= **+1809**, (600) **+2409**, (32) **+2441**, (208) **+2649**, (470) **+3119**, (168)

Anmerkungen:

1. Im Intervall ±3300 treten in der Umgebung der Zahl $n_g = \Pi p_i(p_1 = 2, p_n = 257)$ 22 (fett gedruckte) Primzahlen auf, die (allesamt) gegenüber der Zahl (n_g) Primzahl-Abständen aufweisen.

2. Diese Primzahlen bilden Zwillinge variabler Modularität - die größten Zwillinge wurden unterstrichen markiert. Die Zahl n_g ist im Zwilling mod 1190 situiert.

Tab, 50 Die Folge der Primzahlen im Intervall $\pm\Delta$= 3300 im Umfeld der Zahl $n_g = \Pi p_i(p_1 = 2, p_n = 263)$

(Δ):= **-3169**, (512) **-2657**, (268) **-2389**, (456) **-1933**, (212) **-1721**, (178) **-1543**, (104) **-1439**, (162)

(Δ):= **-1277**, (96) **-1181**, (112) **-1069**, (128) **-941**, (268) **-673**, (72) **-601**, (98) **-383**, (16) **-367**,

(Δ):= (60) **-307**, (708) **+401**, (428) **+859**. (202) **+1061**, (490) **+1531**, (376) **+1907**, (6) **+1913**,

(Δ):= (168) **+2081**, (170) **+2251**, (142) **+2393**, (74) **+2467**, (36) **+2503**, (54) **+2557**, (64)

(Δ):= **+2621**, (156) **+2777**, (284) **+3061**, (190) **+3251**

Anmerkungen:

1. Im Intervall ±3300 treten in der Umgebung der Zahl $n_g = \Pi p_i(p_1 = 2, p_n = 263)$ 32 (fett gedruckte) Primzahlen auf, die (allesamt) gegenüber der Zahl (n_g) Primzahl-Abstände aufweisen.

2. Diese Primzahlen bilden Zwillinge variabler Modularität - die größten Zwillinge wurden unterstrichen markiert. Im größten Zwilling mod 708 dieser Primzahl-Folge ist die Zahl $n_g = \Pi p_i$ situiert.

Tab. 51 Die Folge der Primzahlen im Intervall ±3300 im Umfeld der Zahl $n_g = \Pi p_i(p_1 = 2, p_n = 269)$

(Δ):= **-3271**, (300) **-2971**, (84) **-2887**, (330) **-2557**, (8) **-2549**, (10) **-2539**, (92) **-2447**, (108) **-2339**,

(Δ):= (66) **-2273**, (52) **-2221**, (512) **-1709**, (82) **-1627**, (74) **-1553**, (106) **-1447**, (470) **-977**, (156)

(Δ):= **-821**, (144) **-677**, (984) **+307**, (90) **+397**, (82) **+479**, (140) **+619**, (64) **+683**, (596) **+1279**,

(Δ):= (18) **+1297**, (720) **+2017**, (66) **+2083**, (154) **+2237**, (180) **+2417**, (200) **+2617**, (54) **+2671**,

(Δ):= (120) **+2791**, (228) **+3019**

Anmerkungen:

1. Im Intervall ±3300 treten in der Umgebung der Zahl $n_g = \Pi p_i(p_1 = 2, p_n = 269)$ 32 (fett gedruckte) Primzahlen auf, die Allesamt) gegenüber der Zahl (n_g) Primzahl-Abstände aufweisen.

2. Diese Primzahlen bilden Zwillinge variabler Modularität - die größten Zwillinge wurden unterstrichen markiert. Im größten Zwilling mod 720 dieser Primzahl-Folge ist die Zahl $n_g = \Pi p_i$ situiert.

Die Schlussfolgerungen, die aus den Tabellen: Tab. 48, Tab. 49, Tab. 50 und Tab. 51 folgen:

1. Das Gesetz 2 der "Primzahl-Genese" wird für alle Primzahlen erfüllt, die in den Umgebungen der (lückenlosen) Fakultäten $n_g = \Pi p_i$ situiert sind.

2. Die Zahlen ($n_g = \Pi p_i$) sind stets inmitten sehr großer Zwillinge mod 2n, n>1 situiert.

3. In allen untersuchten Intervallen treten im Umfeld der Zahlen ($n_g = \Pi p_i$) große Schwankungen der Primzahl-Dichten auf.

4. Aus den (durchgeführten) MuPAD - Testen folgt, dass sich ähnliche Dichte - Schwankungen der Primzahlen auch im Umfeld der Fakultäten $n_g = \Pi p_i^{a_i}$ ergeben, in denen einzelne Primzahlen in (beliebigen) Potenzen (a_i) auftreten. Die "periodischen" Schwankungen der Primzahl-Dichten finden so ihre gesetzmäßige Grundlage.

5. In Zahlen-Intervallen ($n \to \infty$) können daher beliebig große primzahlfreie Zonen auftreten, denen Zonen mit größeren Primzahl-Dichten vorausgehen und folgen.
   Ein Beispiel liefert hier der in **Tab. 47** ermittelte Zwilling mod 6594 in welchem asymmetrisch die Zahl $n_g = 2$^1661 situiert ist.

---

16. Die tieferen Ursachen für die Variabilität und die zyklischen Verteilungen der Primzahl - Dichten im Umfeld gerader Zahlen $n_g = \Pi p_i$:

Aus dem bereits unter Beweis gestellten Gesetz 1 folgt, dass eine Addition bzw. Subtraktion von Primzahlen bzw. Primzahl-Produkten zu einer geraden Zahl (n_g) in deren Primzahl-Fakultät diese Primzahlen (p_i) bzw. Primzahl-Produkte ($p_i \cdot p_j$) eingehen, niemals zu Primzahlen führen kann. Dabei treten diese Primzahlen bzw. Primzahl-Produkte in den Primzahl-Fakultäten der Zahlen: $n_g \pm p_i \cdot p_j$ stets auf. Aus dem ebenfalls bereits unter Beweis gestellten Gesetz 2 folgt, dass eine Addition bzw. Subtraktion von Primzahlen bzw. Primzahl-Produkten zu einer geraden Zahl (n_g) in deren Primzahl-Fakultät diese Primzahlen bzw. Primzahl-Produkte nicht eingehen, zu Primzahlen führen kann.
An fünf Beispielen sollen nun die (tieferen) Ursachen erforscht werden, die zu unterschiedlichen Primzahl-Dichten im direkten Umfeld annähernd gleich großer gerader Zahlen (n_g) führen.
In **Tab. 53** wurden die Primzahlen ausgewiesen, die im Intervall $\Delta = \pm 3300$ im Umfeld der geraden Zahl $n_{g,1} = 2 \cdot 3 \cdot 5 \cdot 7 \cdot \ldots \cdot 103 = 23984823528925228172706521638692258396210$ auftreten,
In **Tab. 54** wurden die Primzahlen ausgewiesen, die im Intervall $\Delta = \pm 3300$ im Umfeld der geraden Zahl $n_{g,2} = 2 \cdot 3 \cdot 5 \cdot 7 \cdot \ldots \cdot 101 \cdot 101 = 2351909888008233450906395004931243670070$ auftreten.
In **Tab. 55** wurden im gleichen Intervall die Primzahlen bestimmt, die im Umfeld der geraden Zahl $n_{g,3} = 2 \cdot 5 \cdot 7^2 \cdot 11 \cdot \ldots \cdot 101 \cdot 101 = 54877897200485878052114921678172901896830$ auftreten.
In **Tab. 56** wurden die Primzahlen ausgewiesen, die im Intervall $\Delta = \pm 3300$ im Umfeld der geraden Zahl $n_{g,4} = 2^{19} \cdot 3^{10} \cdot 5^{19} \cdot 7^{20} = 47116535326076910470490000000000000000000$ auftreten und in **Tab. 57** die Primzahlen, die im gleichen Intervall von $\Delta = \pm 3300$ im Umfeld der geraden Zahl $n_{g,5} = 2$^136 = 87112285931760246646623899502532662132736 auftreten.
Diese geraden Zahlen: $n_{g,1}$, $n_{g,2}$, $n_{g,3}$, $n_{g,4}$ und $n_{g,5}$ weisen alle 41 Dezimalstellen auf, treten aber in beträchtlichen Abständen von einander auf - siehe die folgende Gegenüberstellung dieser Zahlen:

$n_{g,1}$ = 23984823528925228172706521638692258396210 ... 41 Dezimalstellen
$n_{g,2}$ = 23519098800208233450906395004931243670070 ... 41 Dezimalstellen
$n_{g,3}$ = 54877897200485878052114921678172901896830 ... 41 Dezimalstellen
$n_{g,4}$ = 47116535326076910470490000000000000000000 ... 41 Dezimalstellen
$n_{g,5}$ = 87112285931760246646623899502532662132736 ... 41 Dezimalstellen

Die Verteilung der Primzahlen in den Umgebungen dieser geraden Zahlen wurde mit dem MuPAD-Test bestimmt (und kann so vom Leser nachvollzogen werden). Die nun folgenden Tabellen wurden so gestaltet, dass aus ihnen die Ursachen für die (unterschiedlichen) Primzahl-Dichten in den gleichen Intervallen, Δ= ±3300 entnommen werden können.

Tab. 53 - Die Primzahlen und Primzahl-Dichten im Umfeld der geraden Zahl $n_{g\,1}$:

$n_{g\,1}$:-/+ **- 3187**, (6) **-3181**, (12) **-3169**, (480) **-2689**, (242) **-2447**, (36) **-2411**, (144) **-2267**, (16) **- 2251**, (140) **-2111**, (114) **-1997**, (298) **-1699**, (380) **-1319**, (40) **-1279**, (48) **-1231**, (18) **-1213**, (104) **-1109**, (70) **-1039**, (18) **-1021**, (50) **-971**, (18) **-953**, (132) **-821**, (34) **-787**, (126) **-661**, (30) **-631**. (132) **-499**, (110) **-389**, (96) **-293**, (156) **-113**, (346) **+223**, (10) **+233**, (60) **+293**, (104) **+397**, (52) **+449**, (12) **+461**, (2) **+463**, (46) **509**, (90) **+599**, (32) **+631**, (118) **+751**, (128) **+877**, (60) **+937**, (4) **+941**, (110) **1051**, (46) **+1087**, (66) **+1163**, (18) **+1181**, (68) **+1249**, (48) **+1297**, (126) **+1423**, (184) **+1607**, ((60) **+1667**, (210) **+1877**, (126) **+2003**, (24) **+2027**, (62) **+2089**, (54) **+2143**, (96) **+2239**, ((12) **2251**, (166) **+2417**, (216) **+2633**, (80) **+2713**, (90) **+2803**. (58) **+2861**, (48) **+2909**, (92) **+3001**, (108) **+3109**

Anmerkungen:

1. Im Intervall Δ= ±3300 treten 69 Primzahlen in wellenartigen Abständen auf, die (allesamt) gegenüber der geraden Zahl $n_{g\,1}$ in "Primzahl-Abständen" auftreten.

2- In der direkten Umgebung von $n_{g\,1}$ erscheint der Zwilling mod 346 .

3. Im analysierten Intervall treten drei weitere große Zwillinge: mod 480, mod 242 und mod 216 auf.

Tab. 54 - Die Primzahlen und Primzahl-Dichten im Umfeld der geraden Zahl $n_{g\,2}$:

$n_{g\,2}$ -/+ **-3089**, (22) **-3067**, (96) **-2971**, (242) **-2729**, (150) **-2579**, (120) **-2459**, (36) **-2423**, (112) **-2311**, (30) **-2281**, (168) **-2113**, (60) **-2053**, (14) **-2039**, (162) **-1877**, (30) **-1847**, (24) **-1823**, (102) **-1721**, (102) **-1619**, (12) **-1607**, (40) **-1567**, (68) **-1499**, (48) **-1451**, (28) **-1423**, (222) **-1201**, (38) **-1163**, (144) **-1019**, (166) **-853**, (14) **839**, (10) **-829**, (8) **-821**, (82) **-739**, (48) **-691**, (48) **-643**, (86) **-557**, (34) **-523**, (14) **-509**, (172) **-337**, (206) **-131**, (28) **-103**, (354) **+251**, (98) **+349**, (90) **+439**, (4) **+443**, (18) **+461**, (26) **+487**, (34) **+521**, (156) **+677**, (62) **+739**, (58) **+797**, (12) **+809**, (98) **+907**, (30) **+937**, (94) **+1031**, (246) **+1277**, (84) **+1361**, (258) **1619**, (44) **+1663**, (90) **+1753**, (6) **+1759**, (244) **+1997**, (20) **+2017**. (66) **+2083**, (130) **+2213**, (54) **+2267**, (156) **+2423**, (50) **+2473**, (184) **+2657**, (14) **+2671**, (60) **+2731**, (126) **+2857**, (52) **+2909**, (180) **+3089**, (102) **+3191**

Anmerkungen:

1. Im Intervall $\Delta = \pm 3300$ treten 72 Primzahlen, allesamt ausschließlich in Primzahl-Abständen zu der geraden Zahl $n_{g,2}$ auf, da Primzahl-Faktoren-Abstände erst im Abstand von $\geq 103 \cdot 103 = 10609$ auftreten können.

2. In der direkten Nachbarschaft der Zahl $n_{g\,2}$ ist der Zwilling mod 354 situiert. In weiteren Abständen von $n_{g\,2}$ treten die Zwillinge mod 242 und mod 222 auf.

3. Die "Genese" der (aller) Primzahlen im Intervall $\Delta = \pm 3300$ ist davon abhängig, ob sich bei der Addition bzw. Subtraktion von Primzahlen $p > 103$ zur geraden Zahl $n_{g\,2}$ Primzahlen einstellen oder Fakultäten aus Primzahlen $p >> 103$.

Tab. 55 - Die Primzahlen und Primzahl-Dichten im Umfeld der geraden Zahl $n_{g\,3}$

$n_{g\,3}$ -/+	-3039, (3 · 1013)	(186)	-2853,	(20)	**-2833,**	(240)	**-2593,**	(16)	-2577,	(26)	**-2551,**
	(82)	-2469,	(86)	**-2383,**	(144)	-2259,	(260)	**-1999,**	(16)	**-1983,**	(90)
	-1893,	(120)	-1773,	(74)	**-1699,**	(136)	-1563,	(54)	-1467,	(8)	**-1459,**
	(130)	-1329,	(8)	**-1321,**	(28)	-1293,	(44)	**-1249,**	(202)	-1047,	(6)
	-1041,	(74)	**-967,**	(16)	-951,	(68)	**-883,**	(24)	**-859,**	(130)	-729, (3^6)
	(12)	-717,	(48)	-669,	(90)	-579,	(6)	-573, ...	(2)	**-571,**	(190)
	-381,	(278)	**-103,**	(100)	-3,	(140)	+137,	(90)	**+227,**	(162)	**+389,**
	(28)	+417,	(50)	**+467,**	(12)	**+479,**	(24)	**+503,**	(250)	+753,	(18)
	+771,	(86)	**+857,**	(22)	+879,	(8)	**+887,**	(96)	**983,**	(108)	**+1091,**
	(60)	**+1151,**	(52)	+1203,	(90)	+1293,	(68)	**+1361,**	(132)	**+1493,**	(10)
	+1503,	(6)	+1509,	(98)	**+1607,**	(4)	+1611,	(26)	**+1637,**	(100)	+1737,
	(156)	+1893,	(48)	+1941,	(62)	**+2003,**	(148)	+2151,	(6)	+2157,	(12)
	+2169,	(140)	**+2309,**	(52)	**+2351,**	(10)	+2361,	(50)	**+2411,**	(36)	**+2447,**
	(184)	+2631,	(98)	**2729,**	(48)	**+2777,**	(406)	+3183,	(68)	**+3251**	

Anmerkungen:

1. Im Intervall $\Delta = \pm 3300$ um die gerade Zahl $n_{g\,3}$ treten 72 Primzahlen auf, wobei die fett gedruckten Primzahl-Abstände prime Zahlen bilden und die nicht fett geduckten Abstände aus Primzahl-Fakultäten gebildet werden, in denen stets die in der Fakultät $n_{g\,3} = \Pi p_i$ ausgeschlossene Primzahl $p = 3$ eingeht.

2. Von den 72 Primzahlen treten nur 35 in Primzahl-Abständen auf, dagegen (aber) 37 in nicht primen Abständen, die allesamt aus Fakultäten gebildet werden, in denen die Primzahl $p = 3$ auftritt. Alle nicht primen Abstände weisen deshalb eine durch 3 teilbare Quersumme (QS) auf.

3. Die gerade Zahl $n_{g\,3}$ ist asymmetrisch in einem Zwilling mod 140 situiert, in größeren Abständen treten hingegen (wellenartig) geringere oder aber höhere Primzahl-Dichten auf. Dabei ist das Auftreten eines Zwillings mod 2 beachtlich, der durch eine punktierte Linie markiert wurde.

4. Beachtlich ist auch das Auftreten zahlreicher Zwillinge mod 4, mod 6, mod 8, mod 10, mod 12. Auf die Zonen mit hoher Primzahl-Dichte folgen zyklisch Zonen mit geringer Primzahl-Dichte.

Tab. 56 - Die Primzahlen und Primzahl-Dichten im Umfeld der geraden Zahl $n_{g\,4}$

$n_{g\,4}$ -/+	-3293, (37·89)	(250)	-3043, (17·179)	(104)	**-2939,**	(12)	**-2927,**	(18)	**-2909,**	(156)	**-2753.**
	(22)	**-2731,**	(144)	-2587, (13·199)	(56)	**-2531,**	(28)	**-2503,**	(132)	**-2371,**	(72)
	-2299, $(11^2\cdot 19)$	(30)	**-2269,**	(12)	-2257, (37·61)	(180)	-2077, (31·67)	(290)	**-1787,**	(10)	-1777,
	(210)	**-1567,**	(120)	**-1447,**	(168)	**-1279,**	(86)	**-1193,**	(144)	**-1049,**	(168)
	-881,	(30)	-851, (23 37)	(10)	-841, (29^2)	(42)	-799, (17·47)	(48)	**-751,**	(54)	-697, (17·41)
	(24)	**-673,**	(312)	-361, (19^2)	(90)	**-271,**	(14)	**-257,**	(246)	**-11,**	(150)*
	+139,	(72)	**+211,**	(18)	+253, (11·23)	(84)	**+337,**	(100)	+437, (19·23)	(14)	+451, (11·41)
	(196)	**+647, ...**	**(2) ....**	+649, (11·59)	(40)	+689, (13·53)	(8)	+697, (17·41)	(4)	**+701,**	(8)
	+709,	(120)	**+829,**	(72)	+901, (17·53)	(60)	+961, (31^2)	(6)	**+967,**	(10)	**+977,**
	(30)	+1007, (19·53)	(62)	**+1069,**	(118)	**+1187,**	(384)	**+1571,**	(170)	**+1741,**	(36)
	+1787,	(2)	**+1789,**	(114)	+1903, (11·173)	(28)	**+1931,..**	**(2) ....**	**+1933,**	(28)	+1961, (37·53)
	(92)	**+2053,**	(90)	**+2143,**	(4)	+2147, (19·113)	(290)	**+2437,**	(136)	+2573, (31·83)	(20)
	+2593,	(300)	+2893, (11·263)	(118)	**+3011,**	(8)	**+3019,**	(4)	**+3023,**	(60)	**+3083,**
	(54)	**+3137**									

Anmerkungen:

1. Im Intervall $\Delta = \pm 3300$ treten im Umfeld der geraden Zahl $n_{g\,4}$ 72 Primzahlen auf, von denen nur die 25 fett gedruckten in Primzahl-Abständen auftreten, wogegen 47 in Primzahl-Faktor-Abständen auftreten, die nicht fett gedruckt wurden.

2. Die Fakultäten der Primzahl-Faktor-Abstände wurden jeweils angeführt.

3. Die gerade Zahl $n_{g\,4}$ ist asymmetrisch im Zwilling mod 150* situiert, der nicht zu den acht großen Zwillingen: mod 384, .mod 312, mod 300, 2x mod 290, mod 250, mod 246, mod 210 zählt, die im Intervall $\Delta = \pm 3300$ auftreten.

4. Im analysierten Intervall treten zwei Zwillinge mod 2 auf, die durch punktierte Linien markiert wurden. Des weiteren treten zahlreiche Zwillinge kleiner Modularität auf.

5. Diese schnell aufeinander folgenden Dichte-Schwankungen sind offensichtlich durch die "Genese" der geraden Zahl $n_{g\,4}$ bedingt, in welcher nur die Primzahlen: p = 2, p =3, p = 5, p = 7 mehrfach (in Potenzen) auftreten, jedoch keine größeren Primzahlen fakultativ eingehen.

5. Durch die "fakultative Genese" der geraden Zahl $n_{g\,4}$ kommen im Umfeld dieser Zahl zahlreiche Primzahl-Faktor-Abstände aus Primzahlen p > 7 (abstandsmäßig) vor, die sich durch Addition bzw. durch Subtraktion von zu $n_{g\,4}$ "benachbarten" Primzahlen ergeben.

Tab. 57 - Die Primzahlen und Primzahl-Dichten im Umfeld der geraden Zahl $n_{g\,5}$:

$n_{g\,5}$ -/+						
	-3219, (72) $(3\cdot29\cdot37)$	-3147, (178) $(3\cdot1049)$	**-2969**, (60)	**-2909**, (56)	-2853, (106) $(3^2\cdot317$	-2747, (90) $(41\cdot(67)$
	-2657, (32)	-2625, (16) $(3\cdot5^3\cdot7)$	**-2609**, (20)	-2589, (30) $(3\cdot863)$	**-2559**, (76)	-2535, (60) $(3\cdot5\cdot13^2)$
	-2475, (22) $(3^2\cdot5^2\cdot11)$	-2453, (56) $(11\cdot223)$	-2397, (312) $(3\cdot17\cdot47)$	-2085, (40) $(3\cdot5\cdot139)$	**-1949**, (206)	-1743, (366) $(5\cdot409)$
	-1377, (12) $(3^4\cdot17)$	-1365, (48) $(3\cdot5\cdot7\ 13)$	-1317, (30) $(3\cdot439)$	-1287, (42) $(3^2\cdot11\cdot13)$	-1245, (52) $(3\cdot5\cdot83)$	**-1193**, (48)
	-1145, (252) $(5\cdot229)$	-893, (150) $(19\cdot47)$	**-743**, (134)	-609, (130) $(3\cdot7\cdot29)$	**-479**, (54)	-425, (2) $(5^2\cdot17)$
	-423, (46) $(3^2\cdot47)$	-377, (8) $(13\ 29)$	-369, (4) $(3^2\ 41)$	-365, (30) $(5\cdot73)$	-335, (36) $(5\cdot67)$	-299, (2) $(13\cdot23)$
	-297, (40) $(3^2\cdot11)$	**-257**, (14)	-243, (130) (3^5)	**-113**, (198)*	+85, (8) $(5\cdot17)$	+93, (28) $(3\cdot31)$
	+121, (28) (11^2)	+147, (60) $(3\cdot7^2)$	+207, (90) $(3^2\cdot23)$	+297, (186) $(3^3\cdot11)$	+483, (10) $(3\cdot7\cdot23)$	**+571**, (164)
	+571,. (164)	+735, (112) $(3\cdot5\cdot7^2)$	+847, (14) $(7\cdot11^2)$	+861, (24) $(3\cdot7\cdot41)$	+885, (6) $(3\cdot5\cdot59)$	+891, (160) $(3^4\cdot11)$
	+1051, (66)	**+1117**. (30)	+1371, (42) $(3\cdot457)$	+1413, (102) $(3^2\cdot157)$	+1515, (18) $(3\cdot5\cdot101)$	+1533, (168) $(3\ 7\cdot73)$
	+1701, (42) $(3^5\cdot7)$	+1745, (140) $(3\cdot7\cdot83)$	+1885, (56) $(5\cdot13\cdot29)$	+1941, (126) $(3\cdot647)$	+2067, (144) $(3^2\cdot257)$	+2211, (102) $(3\cdot11\cdot67)$
	+2313, (42) $(3^2\cdot257)$	+2355, (76) $(3\cdot5\ 157)$	+2431, (86) $(11\cdot13\cdot17)$	+2517, (186) $(3\cdot839)$	+2703, (22) $(3\cdot17\cdot53)$	+2725, (110) $(5^2\cdot109)$
	+2835, (90) $(3^4\cdot5\cdot7)$	+2925, (30) $(3^2\cdot5^2\cdot13)$	+2955, (78) $(3\cdot5\cdot197)$	+3033, (64) $(3^2\cdot337)$	+3097, (24) $(19\cdot163)$	**+3121**, (80)
	+3201, (60) $(3\cdot11\cdot97)$	+3261 $(3\cdot1087)$				

Anmerkungen:

1. Im Intervall $\Delta = \pm 3300$ treten 83 Primzahlen auf, die überwiegend in Primzahl-Faktor-Abständen zu der geraden Zahl $n_{g\,4}$ situiert sind. Lediglich 15 Primzahlen treten in Primzahl-Abständen zu $n_{g\,4}$ auf, 68 primzahlen treten hingegen in Primzahl-Faktor-Abständen auf. Ursächlich dafür ist, dass in der geraden Zahl $n_{g\,5} = 2$^136 keine der ungeraden Primzahlen $p \geq 3$ fakultativ auftritt.

2. Die Primzahl-Faktoren der ungeraden Abstands-Zahlen wurden (jeweils) unterhalb dieser ungeraden Abstands-Zahlen notiert.

3. Die Primzahl-Abstände wurden fett gedruckt hervorgehoben, derweil die nicht primen Abstände nicht fett gedruckt notiert wurden.

4. Nach Gesetz 2 können (fallweise) alle Primzahl-Faktoren (ausgenommen die Primzahl $p = 2$) zur Genese neuer Primzahlen bei der Addition bzw. der Subtraktion von der geraden Zahl $n_{g,\,4} = 2$^136 führen.

5. Ein Vergleich mit den Dichte-Schwankungen in den direkten Umgebungen der Zahlen $n_{g\,1}$ und $n_{g\,2}$ (wo in den direkten. Nachbarschaften dieser Zahlen stets große Primzahl-freie Zonen auftreten) zeigt hier ein umgekehrtes Bild. In der direkten Nachbarschaft der geraden Zahl $n_{g\,4}$ manifestiert sich eine besonders hohe Primzahl-Dichte. Es kommen des weiteren zwei (neu generierte) Zwillinge mod 2 auf.

6. Die Dichte-Verteilung der Primzahlen ist in der Umgebung der geraden Zahl $n_{g\,5} = 2$^136 relativ ausgeglichen - ursächlich dafür ist die hohe Zahl der "Primzahl-Faktor-Abstände" der Primzahlen von der geraden Zahl $n_{g\,5}$.

Ein Vergleich der Primzahl-Dichte im direkten Umfeld der geraden Zahlen $n_{g,1} = 2 \cdot 3 \quad 5 \cdot 7 \cdot ... \cdot 103$ (siehe **Tab. 53**) mit der Primzahl-Dichte im direkten Umfeld der geraden Zahl $n_{g,5} = 2$ ^136 (siehe **Tab. 57**) zeigt ein antizyklisches Verhalten der Primzahl-Dichten. Im direkten Umfeld der Zahl $n_{g,1}$ liegt eine wesentlich geringere Primzahl-Dichte vor als im Umfeld der Zahl $n_{g,5}$.

Die nachfolgenden Untersuchungen der Primzahlen im Intervall Δ= ±3300 und in den Intervallen Δ= -13081 bis -10609 sowie Δ= +10609 bis +13081 der folgenden geraden Zahlen: $n_{g,6}$, $n_{g,7}$, $n_{g,8}$ und $n_{g,9}$ führt zu der Erkenntnis, dass dieses Verhalten der Primzahl - Dichten einer Gesetzmäßigkeit entspricht - siehe **Tab. 58**, **Tab. 59**, **Tab. 60** und **Tab. 61**:

$n_{g,6} = 2 \cdot 3 \cdot 5 \cdot 7 \cdot ... \cdot 101 =$ 232862364358497360900063316880507363070 ... 39 Dezimalstellen
$n_{g,7} = 2 \cdot 3^2 \cdot 5 \cdot 7 \cdot ... \cdot 101 =$ 698587093075492082700189950641522089210 ... 39 Dezimalstellen
$n_{g,8} =$ 2^128 = 340282366920938463463374607431768211456 ... 39 Dezimalstellen
$n_{g,9} =$ 2^129 = 680564733841876926926749214863536422912 ... 39 Dezimalstellen

Tab. 58 - Die Primzahlen und Primzahl-Dichten im Umfeld der geraden Zahl $n_{g,6}$:

$n_{g,6}$-/+ **-3181**, (72) **-3109**, (20) **-3089**, (22) **-3067**, (30) **-3037**, (66) **-2971**,
(128) **2843**, (10) **-2833**, (140) **-2693**, (412) **-2281**, (12) **-2269**, (26)
-2243, (244) **-1999**, (48) **-1951**, (18) **1833**, (132 **1801**, (14) **-1787**,
(130) **-1657**, (60) **-1597**, (30) **-1567**, (120) **-1447**, (20) **-1427**, (60)
-1367, (60) **-1307**, (10) **-1297**, (248) **-1049**, (66) **983**, (30) **-953**,
(96) **-857**, (118) **-739**, (30) **-709**, (186) **523**, (14) **-509**, (18)
-491, (28) **-463**, ... (2) .... **-461**, (222) **-239**, (72) **167**, (189 **-149**,
(10) **-139**, (8) **-131**, (364)* **+233**, (290) **+523**, (48) **+571**, (76)
+647, (54) **+701**, (38) **+739**, (22) **+761**, (158) **+919**, (234) **+1153**,
(126) **+1279**, (28) **1307**, (54) **+1361**, (92) **+1453**, (30) **+1483**, (28)
+1511, (108) **+1619**, (2) **+1621**, (168) **+1789**, (84) **+1873**, (180) **+2053**,
(34) **+2087**, (44) **+2131**, (22) **+2153**, (98) **+2251**, (88) **+2339**, (18)
+2357, (26) **+2383**, (54) **+2437**, (22) **+2459**, (162) **+2621**, (92) **+2713**,
(148) **+2861**, (18) **+2879**, (92) **+2999**, (258) **+3257**

$n_{g,6}$ - **-13063**, (96) **-12967**, (48) **-12919**, (138) **12781**, (24) **-126757**, (120) **-12637**,
(96) **-12541**, (24) **12517**, (14) **-12503**, (12) **-12491**, (114) **-12377**, (30)
-12347, (106) **12241**, (92) **-12149**, (58) +12091, (150) **11941**, (308) **11633**,
(107 · 113)
(46) **-11587**, (164) **-11423**, (54) **-11369**, (108) **-11261**, 112) **-11149**, (182)
-10987, (50) **-10937**, (29) **-10909**, (170) **-10739**

Tab. 58 - Fortsetzung

$n_{g,6}$ ± **+10613**, (<u>296</u>) **+10909**, (48) **+10957**, (<u>240</u>) **+11197**, (30) +11227, (24) (103 · 109) **+11251**, (102) **+11353**, (40) **+11393**, (56) +11449, (34) (107 · 107) **+11483**, (138) **+11621**, (62) +11663, (14) **+11677**, (22) **11699,...(2) ....** **+11701**, (16) **11717**, (96) **+11813**, (107 · 109) (8) **+11821**, (66) **+11887**, (94) **+11981**, (62) **+12043**, (58) **+12101**, (60)

+12161, (78) **12239**, (<u>300</u>) **+12539**, (132) **+12671**, (18) **+12689**, (68) **+12699**, (66) **+12823**, (66) **+12823**, (88) **+12911**, (126) **13037**, (44) +13081 = 103 · 127

Anmerkungen::

1. Im Intervall Δ= ±3300 treten ausschließlich Primzahl-Abstände zwischen der Zahl $n_{g,6}$ und den 76 Primzahlen auf.

2. Die "variierenden" Abstände zwischen den benachbarten Primzahlen resultieren aus den folgenden Gründen:

   Grund 1: Die Primzahlen der Folge: p_j = 103 bis 3259 treten bereits in variierenden Abständen auf.

   Grund 2: Zahlreiche ungerade Zahlen vom Typ $n_u = n_{g,6} \pm p_j > 101$ sind aus Primzahl-Faktoren konstituiert deren einzelne Faktoren stets größer als p_n =101 sind - hier zwei Beispiele::

   Beispiel 1: $n_{g,6} - 103 = 26883996799249 \cdot 86617464693714117871997 83$
   Beispiel 2: $n_{g,6} + 103 = 131 \cdot 93515873911 \cdot 19008273420385089638465153$

   Diese Beispiele bestätigen Gesetz 2. Nach Gesetz 1 kann bei der Addition bzw. Subtraktion von Primzahlen $p \leq p_n$ bzw. ihren Produkten niemals eine Primzahl vom Typ: $n_{g,6} \pm (p \leq p_n)$ entstehen - hier vier Beispiele:

   Beispiel 3: $n_{g,6} - 101 = 101 \quad 6683 \cdot 34750485722825671423466572789 43$
   Beispiel 4: $n_{g,6} + 101 = 101 \cdot 2336993 \cdot 13848803 \cdot 71237436024091007473549$
   Beispiel 5: $n_{g,6} - 37 \cdot 61 = 37 \cdot 61 \cdot 179 \cdot 199 \cdot 10151093 \cdot 11557627 \cdot 2468767294263439$
   Beispiel 6: $n_{g,6} + 61 \cdot 97 = 61 \cdot 97 \cdot 604859 \cdot 745027 \cdot 226201253039 \cdot 386079345193$

3. Ab den Zahlen-Bereichen: $n_{g,6} < n_{g,6} - 103 \cdot 103 = n_{g,6} - 10609$ und $n_{g,6} > n_{g,6} + 10609$ können, wie aus Tab. 58 ersichtlich (fallweise) Primzahlen in den Faktor-Abständen ±103 · 103, 103 · 107, ... von $n_{g,6}$ auftreten. Sofern die Addition bzw. Subtraktion von Primzahl-Faktoren: (p > 101) keine Primzahlen ergeben, erfolgt eine "Verschmelzung" dieser Faktoren zu Fakultäten in denen ausschließlich größere Primzahlen auftreten - dies möge an zwei Beispielen demonstriert werden:

   Beispiel 7: $n_{g,6} - 107 \cdot 113 = 2328623643584973609000633160805073509 79$ ... eine Primzahl
   Beispiel 8: $n_{g,6} + 107 \cdot 113 = 9041 \cdot 6192349 \cdot 961956517 \cdot 432386341910808613 7$

4. Die hier kommentierten Sachverhalte erfüllen (ohne Ausnahme) die Gesetze: Gesetz 1 und Gesetz und erklären, warum im direkten Umfeld gerader Zahlen $n_g = \Pi p_i.(p_1 = 2$ bis $p_n)$ eine Zone mit einer geringen Primzahl-Dichte (unvermeidbar) auftreten muss.

5. Die Ursachen für dass "wellenartige" Auftreten von Zonen mit geringer bzw. hoher Primzahl-Dichte resultieren aus den unter Pkt. 2 genannten Gründen: Grund 1 und Grund 2.
   In Tab. 58 wurden die großen Zwillinge mod (>200) unterstrichen markiert und der Zwilling mod 2 durch eine punktierte Linie hervorgehoben.

Aus **Tab. 59** ist ersichtlich, dass ähnliche Primzahl-Dichte-Verteilungen im Umfeld der geraden Zahl $n_{g,7}$ im Intervall Δ= ±3300 auftreten mit einer geringen Dichte im direkten Umfeld von $n_{g,7}$.

Tab. 59 Die Primzahlen und Primzahl-Dichten im Umfeld der geraden Zahl $n_{g\ 7}$:

$\underline{n_{g\ 7}\text{-/+}}$ **-3259**, (6) **-3253**, (86) **-3167**, (100) **-3067**, (26) **-3041**, (<u>288</u>) **-2753**, (90) **-2663**, (16) **-2647**, (30) **-2617**, (26) **-2591**, (12) **-2579**, (28) **-2551**, (84) **-2467**, (134) **-2333**, (40) **-2293**, (42) **-2251**, (108) **-2143**, (14) **-2129**, (42) **-2087**, (70) **-2017**, (186) **-1831**, (134) **-1697**, (60) **-1637**, (70) **-1567**, (8) **-1559**, (100) **-1459**, (12) **-1447**, (<u>356</u>) **-1091**, (28) **-1063**, (122) **-941**, (112) **-829**, (60) **-769**, (78) **-691**, (170) **-521**, (12) **-509**, (46) **-463**, (80) **-383**, (126) **-257**, (118) **-139**, (30) <u>**-109**, (258)* **+149**,</u> (24) **+173**, (18) **+191**, (78) **+269**, (38) **+307**, (10) **+317**, (66) **+383**, (14) **+397**, (82) **+479**, (90) **+569**, (38) **+607**, (6) **+613**, (6) **+619**, (28) **+647**, (6) **+653**, (30) **+683**, (104) **+787**, (10) **+797**, (12) **+809**, (158) **+967**, (16) **+983**, (89 **+991**, (160) **+1151**, (108) **1259**, (42) **+1301**, (158) **+1459**, (22) **+1481**, (128) **+1609**, (60) **+1669**, (40) **+1709**, (14) **+1723**, (36) **+1759**, (24) **+1783**, (40) **+1823**, (188) **+2011**, (102) **+2113**, (90) **+2203**, (190) **+2393**, (18) **+2411**, (6) **+2417**, (50) **+2467**, (6) **+2473**, (30) **+2503**, (48) **+2551**, (162) **+2713**, (76) **+2789**, (14) **+2803**, (34) **+2837**, (6) **+2843**, (44) **+2887**, (82) **+2969**, (72) **+3041**, (20) **+3061**, (76) **+3137**

$\underline{n_{g\ 7}\text{-}}$ **-12979**, (72) **-12907**, (144) **-12763**, (156) **-12611**, (70) **-12541**, (24) **-12517**, (80) **12437**, (174) **-12263**, (172) -12091, (84) (107 · 113) **-12007**, (38) **-11969**, (16) **-11953**, (<u>290</u>) -11663, (172) (107 · 109) **-11491**. (68) **-11423**, (30) **-11393**, (<u>222</u>) **-11171**, (<u>262</u>) **-10909**, (<u>222</u>) **-10687**

$\underline{n_{g\ 7}\text{+}}$ **+10613**, (116) **+10729**, (24) **+10753**, (106) **+10859**, (24) **+10883**, (6) **+10889**, ...

.... (2) **+10991**, (<u>520</u>) **+11411**, (38) +11449, (<u>364</u>) (107 · 107) **+11813**, (8) **+11821**, (10) **+11831**, (96) **+11927**, (32) **+11959**, (78) **+12037**, (4) **+12041**, (66) **+12107**, (12) **+12119**, (38) **+12157**, (106) **+12263**, (26) **+12289**, (58) **+12347**, (54) **+12401**, (20) **+12421**. (118) **+12539**, (50) **+12589**, (22) **+12611**, (92) **+12703**, (36) **+12739**, (102) **+12841**, (12) **+12853**, (36) **+12889**, (118) **+13007**

Anmerkungen:

1. In **Tab. 58** werden die in **Tab. 57** nachgewiesenen Gesetze vollauf bestätigt - so befindet sich im direkten Umfeld der geraden Zahl $n_{g,7}$ eine geringe Primzahl-Dichte. Die Zahl $n_{g,7}$ ist asymmetrisch in einem Zwilling mod 258* situiert und alle Primzahlen im Intervall Δ= ±3300 sind in primen Abständen von $n_{g,7}$ situiert.

2. In den Intervallen: Δ= -13081 bis -10609 sowie +10609 bis +13081 sind drei Primzahlen in den Abständen: -107 · 113, -107 · 109 und +107 · 107 situiert.

3. Im Intervall Δ= +10609 bis 13081 ist ein Zwilling mod 2 situiert, der durch eine punktierte Linie markiert wurde.

In **Tab. 60** und **Tab. 61** wird der Nachweis erbracht, dass Gesetz 1 und Gesetz 2 auch für alle geraden Zahlen vom Typ $n_g = 2^n$ Geltung haben. In diesen geraden Zahlen treten keine Primzahlen p:= 3, 5, 7, ... auf. Gesetz 1 kann an den folgenden vier Beispielen unter Beweis gestellt werden:

Beispiel 1:	n_g = 2^16 - 2	= $2 \cdot 7 \cdot 31 \cdot 151$	= 65534
Beispiel 2:	n_g = 2^16 - 2 ·2	= $2^2 \cdot 3 \cdot 43 \cdot 127$	= 65532
Beispiel 3:	n_g = 2^16 +2	= $2 \cdot 3^2 \cdot 11 \cdot 331$	= 65538
Beispiel 4:	n_g = 2^16 +2 ·2	= $2^2 \cdot 5 \cdot 29 \cdot 113$	= 65540

Die addierten bzw. subtrahierten Primzahl-Faktoren: 2, 2^2, ... treten in der gleichen Potenz in der Faktor-Zerlegung der (jeweiligen) geraden Zahlen auf, so wie es Gesetz 1 bestimmt.

Für den Nachweis der Gültigkeit von Gesetz 2 mögen die folgenden sechs Beispiele dienen:

Beispiel 1:	$n_{u,1}$ = 2^16 -3 = $13 \cdot 71^2$	= 65533	Beispiel 2:	$n_{u,2}$ = 2^16 -5 = $19 \cdot 3449$	= 65531
Beispiel 3:	$n_{u,3}$ = 2^16 -15	= **65521**	Beispiel 4:	$n_{u,4}$ = 2^16 +1 =	= **65537**
Beispiel 5:	$n_{u,5}$ = 2^16 +3	= **65539**	Beispiel 6:	$n_{u,6}$ = 2^16 +5 = $3 \cdot 7 \cdot 3121$	= 65541

Gesetz 2 besagt, dass bei der Addition bzw. Subtraktion von Primzahlen bzw. Primzahl-Faktoren zu einer geraden Zahl die in der Primzahl-Fakultät der sich dann ergebenden ungeraden Zahl nicht mehr auftreten können. Diese addierten bzw. subtrahierten Primzahlen oder Primzahl-Faktoren mutieren vielmehr zu andersartigen Primzahl-Fakultäten oder zu einer Primzahl, was in den sechs Beispielen eine Bestätigung findet. Unter diesem Aspekt sind die Ergebnisse von **Tab. 60** und **Tab. 61** für den Leser leicht nachvollziehbar und können durch weitere Beispiele bestätigt werden.

Tab. 60 Die Primzahlen und Primzahl-Dichten im Umfeld der geraden Zahl $n_{g,8}$:

$n_{g,8}$ -/+	-2717,	(230)	-2487,	(10)	**-2477,**	(18)	**-2459,**	(104)	-2355,	(168)	-2187,
	$(11 \cdot 13 \cdot 19)$		$(3 \cdot 829)$						$(3 \cdot 5 \cdot 157)$		(3^7)
	(64)	-2123,	(78)	-2045,	(66)	**-1979,**	(26)	-1953,	(18)	-1935,	(66)
		$(11 \cdot 193)$		$(5 \cdot 409)$				$(3^2 \cdot 7 \cdot 31)$		$(3^2 \cdot 5 \cdot 43)$	
	-1869,	(12)	-1857,	(58)	-1799,	(114)	-1685,	(6)	-1679,	(104)	-1575,
	$(3 \cdot 7 \cdot 89)$		$(3 \cdot 619)$		$(7 \cdot 257)$		$(5 \cdot 337)$		$(23 \cdot 73)$		$(3^2 \cdot 5^2 \cdot 7)$
	(40)	-1535,	(48)	**-1487,**	(8)	-1479,	(60)	-1419,	(10)	**-1409,...**	(2) ......
		$(5 \cdot 307)$				$(3 \cdot 17 \cdot 29)$		$(3 \cdot 11 \cdot 43)$			
	.......-1407,	(102)	-1305,	(112)	**-1193,**	(396)	**-797,**	(84)	-713,	(38)	-675,
	$(3 \cdot 7 \cdot 67)$		$(3^2 \cdot 5 \cdot 29)$						$(23 \cdot 31)$		$(3^3 \cdot 5^2)$
	(318)	-357,	(120)	-275,	(38)	-237,	(6)	**-233,**	(60)	**-173,**	(14)
		$(3 \cdot 7 \cdot 17)$		$(5^2 \cdot 11)$		$(3 \cdot 79)$					
	-159,	(210)*	+51,	(30)	+81,	(82)	+165,	(108)	+273,	(112)	+385,
	$(3 \cdot 53)$		$(3 \cdot 17)$		(3^4)		$(3 \cdot 5 \cdot 11)$		$(3 \cdot 7 \cdot 13)$		$(5 \cdot 7 \cdot 11)$
	(36)	**+421,**	(42)	**+453,**	(110)	+573,	(52)	+625,	(132)	**+757,**	(60)
						(3 191)		(5^4)			

Tab. 60 - Fortsetzung

+817, (30) +847, (326) +1173, (114) +1287, (496) **+1783**, (92) +1875,
(19·43) (7·11²) (3·17·23) (3²·11·13) (3·5⁴)
(100) +1975, (272) +2247, (36) +2283, (54) +2337, (90) +2427, (4)
(5²·79) (3·7·107) (3·761) (3·19·41) (3·809)
+2431, (2) +2433, (24) +2457, (40) +2497, (398) +2895, (150) +3045,
(11·13·17) (3·811) (3³·7·13) (11·227) (3·5·193) (3·5·7·29)
(72) +3117, (60) +3177, (16) +3193, (8) +3201, (4) +3205, (80) +3285
(3·1039) (3²·353) (31·103) (3·11·97) (5·641) (3²·5·73)

$n_{g\ 8}$ - **-13037**, (168) -12869, (44) -12825, (28) -12797, (38) -12759, (112) **-12647**,
(17·757) (3³·5²·19) (67·191) (3·4253)
(74) -12573, (54) -12519, (72) -12447, (28) -12419, (174) -12245, (80)
(3²·11·127) (3²·13·107) (3³·461) (11·1129) (5·31·79)
-12165, (112) -12053, (204) -11849, (6) -11843, (144) **-11699**, (30) -11669,

(96) -11573, (408) -11165, (6) **-11159**, (44) -11115, (48) -11067, (48)
(71·163) (5·7·11·29) (3²·5·13·19) (3·7·17·31)

-11019, (66) -10953, (34) -10919, (12) -10907, (42) -10865, (62) -10803,
(61·179) (3²·1217) (3·3673) (13·839) (5·41·53)(3·13·277)

(76) -10727, (92) -10635
(17·631) (3·5·709)

$n_{g\ 8}$ + +10815, (250) +11065, (72) +11137, (44) +11181, (156) +11337, (4) +11341,

(20) +11361, (84) +11445, (30) +11475, (10) +11485, (308) +11793, (298)

+12091, (26) +12117, (168) +12285, (88) **12373**, (12) +12385, (32) +12417,

(34) **+12451**, (102) **+12553**, (18) +12571, (86) +12657, (18) +12675, (150)

+12825, (106) +12931, (62) +12993, (60)

Anmerkungen:

1. Im Intervall Δ= ±3300 treten in der Umgebung der geraden Zahl $n_{g\ 8}$ 67 Primzahlen in variierenden Abständen auf, von denen 13 in Primzahl-Abständen und 54 in Primzahl-Faktor-Abständen auftreten. Die Primzahlen, die in Primzahl-Abständen (von $n_{g\ 8}$) auftreten, wurden fett gedruckt markiert.

2. Die gerade Zahl $n_{g\ 8}$ = 2^128 ist asymmetrisch im Zwilling mod 210* situiert, also in einer Zone mit geringer Primzahl-Dichte. In größeren Abständen von $n_{g\ 8}$ treten (variierend) weitere Zonen mit einer geringen Primzahl-Dichte auf und dazwischen sind (variierend) Zonen mit höheren Primzahl-Dichten situiert, in denen auch ein Zwilling mod 2 auftritt, der durch eine punktierte Linie markiert wurde.

3. In den Intervallen: Δ:= -13081 bis -10609 und Δ = + 10609 bis +13081 sind insgesamt nur 56 Primzahlen situiert, von denen 19 in Primzahl-Abständen und 27 in Primzahl-Faktor- Abständen auftreten, die (jeweils) angegeben wurden.

4. Auch in den (weiter entfernten) Intervallen treten variierende Primzahl-Dichten "wellenartig" auf. Die Zwillinge mod (>200) wurden durch Unterstreichung markiert.

5 Gesetz 2 der Primzahl-Generierung wird in **Tab. 60** vollauf bestätigt.

Tab. 61 Die Primzahlen und Primzahl-Dichten im Umfeld der geraden Zahl $n_{g,9}$:

$n_{g,9}$ -/+	-3199,	(274)	-2925,	(210)	-2715,	(176)	**-2539**,	(20)	-2359,	(4)	-2335,	
	$(7 \cdot 457)$		$(3^2 \cdot 5^2 \cdot 13)$		$(3 \cdot 5 \cdot 181)$				$(3 \cdot 8539$		$(3 \cdot 5 \cdot 157)$	
	(20)	-2335,	(6)	-2329,	(120)	-2209,	(60)	-2089,	(60)	**-2089**,	(46)	
		$(5 \cdot 467)$		$(17 \cdot 137)$		(47^2)		$(7 \cdot 307)$				
	-2043,	(482)	-1561,	(30)	**-1531**,	(132)	**-1399**,	(30)	-1369,	(4)	-1365,	
	$(3^2 \cdot 227)$		$(7 \cdot 223)$						(37^2)		$(3 \cdot 5 \cdot 7 \cdot 13)$	
	(180)	-1185,	(72)	-1113,	(308)	-805,	(64)	-741,	(6)	-735,	(122)	
		$(3 \cdot 5 \cdot 79)$		$(3 \cdot 7 \cdot 539$		$(5 \cdot 7 \cdot 23)$		$(3 \cdot 13 \cdot 19)$		$(3 \cdot 5 \cdot 7^2)$		
	-613,	(210)	-403,	(88)	-315,	(290)	-25,	(42)*	**+17**,	(58)	+75,	(86)
			$(13 \cdot 43)$		$(3^2 \cdot 5 \cdot 7)$		(5^2)				$(3 \cdot 5^2)$	
	+161,	(150)	**+311**,	(150)	+425,	(114)	+579,	(154)	+591,	(12)	+669,	
	$(7 \cdot 23)$				$(5^2 \cdot 17)$		$(3 \cdot 193)$		$(3 \cdot 191)$		$(3 \cdot 223)$	
	(78)	**+701**,	(34)	+735,	(12)	+747,	(188)	+935,	(12)	**+947**,	(40)	
				$(3 \cdot 5 \cdot 7^2)$		$(3^2 \cdot 83)$		$(5 \cdot 11 \cdot 17)$				
	+987,	(158)	+1145,	(36)	**+1181**,	(40)	+1221,	(84)	+1305,	(102)	+1407,	
	$(3 \cdot 7 \cdot 47)$		$(5 \cdot 229)$				$(3\ 11 \cdot 37)$		$(3^2 \cdot 5 \cdot 29)$		$(3 \cdot 7 \cdot 67)$	
	(120)	+1527,	(170)	**+1697**,	(72)	+1769.	(30)	+1799,	(78)	**+1877**,	(100)	
		$(3 \cdot 509)$				$(29 \cdot 61)$		$(7 \cdot 257)$				
	+1977,	(54)	+2031,	(50)	**+2081**,	(106)	+2187,	(50)	**+2237**,	(84)	+2321,	
	$(3 \cdot 659)$		$(3 \cdot 677)$				(3^7)				$(11 \cdot 211)$	
	(130)	+2451,	(50)	+2501,	(54)	+2555,	(84)	+2639,	(52)	+2691,	(414)	
		$(3 \cdot 19 \cdot 43)$		$(41 \cdot 61)$		$(5 \cdot 7 \cdot 73)$		$(7 \cdot 13 \cdot 29)$		$(3^2 \cdot 13 \cdot 23)$		
	+3105,	(154)	+3159,	(122)	+3281							
	$(3^3 \cdot 5 \cdot 23)$		$(3^5 \cdot 13)$		$(17 \cdot 193)$							

Anmerkungen:

1. Im Intervall $\Delta = \pm 3300$ treten im Umfeld der geraden Zahl $n_{g,9} = 2$^129 insgesamt 64 Primzahlen auf, von denen 13 Primzahlen in Primzahl-Abständen situiert sind (diese primen Abstände wurden fett gedruckt hervorgehoben) und 51 in Primzahl-Faktor-Abständen (diese Abstände wurden nicht fett gedruckt und die jeweiligen Primzahl-Faktoren dieser Abstände wurden in Klammern vermerkt).

2. Die gerade Zahl $n_{g,9} =$ 2^129 ist in einer Zone mit einer relativ hohen Primzahl-Dichte situiert und befindet sich (leicht asymmetrisch) in einem Zwilling mod 42* . Im Vergleich hierzu ist die gerade Zahl $n_{g,8} =$ 2^128 (ebenfalls asymmetrisch) in einem Zwilling mod 210* situiert, also in einer Zone mit einer (wesentlich) geringeren Primzahl-Dichte,

3. Auf die Untersuchung der Primzahl-Verteilungen in den Abständen: $\Delta = -13081$ bis -10609 sowie $\Delta = +10609$ bis 13081 wurde in **Tab. 61** verzichtet.

4. Gesetz 2 der Primzahl-Generierung wird in Tab. 61 vollauf bestätigt.

---

17. Die Suche nach dem Gesetz, dem die zyklischen Wiederholungen von Primzahl-Dichten in wachsenden Zahlen-Intervallen zugrunde liegt und des Gesetzes der Genese von Primzahl-Zwillingen mod 2 und von Zwillingen mod 2n, n>1:

Die der Mathematik heute zur Verfügung stehenden schnellen PC-Rechner und Rechenprogramme, wie z.B. MuPAD-Pro, versetzen uns heute in die Lage, nach dem Gesetz der zyklischen Wiederholung ähnlicher Primzahl-Dichten in wachsenden Zahlen-Intervallen zu fanden.
Für das Verständnis dieses Gesetzes wurden die Primzahl-Dichten für das gleiche Zahlen-Intervall von $\Delta = \pm 10000$ in der Zahlenfolge $n \rightarrow \infty$ in definierter Weise für eine Reihe gerader und ungerader Zahlen analysiert, die aus gleichen Primzahl-Faktoren konstituiert sind. Für diese Untersuchungen ist ein Rechen-Aufwand mit dem MuPAD-Programm von mehreren Tagen erforderlich.

In **Tab. 62** wurden die Primzahl-Dichten in der Umgebung der geraden Zahlen $n_g = 2^n$ für die Exponenten: n:= 65, 72, 531, 2000 und 2500 (vereinfachend) angegeben und kommentiert.

Tab. 62 Die Primzahlen im Intervall $\Delta = \pm 10.000$ ($\Sigma\Delta = 20000$) im Umfeld von $n_g = 2^n$:

n_g = 2^n:	2^65	2^72	2^531	2^2000	2^2500	Anmerkung:
Dezimalstellen:=	20	22	160	603	753	
Primzahlen (*P*):	461	426	53	17	10	
P/ΣΔ:=	0,02308	0,02130	0,00265	0,00085	0,00050	***P***-Dichte
ΣΔ/ *P*:=	43,48	47,06	373,85	1051,5	1306,66	ΣΔ/***P***
Zwilling mod:=	(220)	(108)	(690)	(3058)	(1924)	n_g inmitten

Die Primzahl - Abstände von n_g und ihre Prim - Faktor-Strukturen sowie die aufeinander folgenden Zwillinge:

	2^65	2^72	2^531	2^2000	2^2500
Δ_1	-9999	-9999	-9799	-8969	-8837
	$3^2 \cdot 11 \cdot 101$	$3^2 \cdot 11 \cdot 101$	$41 \cdot 239$	8969	8837
	(8)	(54)	(64)	(524)	(1304)
Δ_2	-9991	-9945	-9735	-8445	-7533
	$97 \cdot 101$	$3^2 \cdot 5 \cdot 13 \cdot 17$	$5 \cdot 1951$	$3 \cdot 5 \cdot 563$	7533
	(126)	(36)	(446)	(1006)	(8)
Δ_3	-9865	-9909	-9289	-7439	-7325
	$5 \cdot 19 \cdot 73$	$3^3 \cdot 367$	$7 \cdot 1327$	$43 \cdot 173$	$5^2 \cdot 293$
	(52)	(34)	(158)	(350)	(56)
Δ_4	-9813	-9875	-9111	-7089	-7269
	$3 \cdot 3271$	$5^3 \cdot 79$	$3 \cdot 3037$	$3 \cdot 17 \cdot 139$	$3 \cdot 2423$
	(12)	(38)	(546)	(240)	(4704)
Δ_5	-9801	-9837	-8565	-6849	-2565
	$3^4 \cdot 11$	$3^2 \cdot 1093$	$3 \cdot 5 \cdot 571$	$3^2 \cdot 761$	$3^3 \cdot 5 \cdot 19$
	(30)	(12)	(476)	(232)	(468)
Δ_6	-9771	-9825	-8089	-6617	-2097
	$3 \cdot 3257$	$3 \cdot 5^2 \cdot 131.$	8089	$13 \cdot 509$	$3^2 \cdot 233$
	(6)	(12)	(258)	(660)	(474)
Δ_7	-9765	-9813	-7831	-5957	-1623
	$3^2 \cdot 5 \cdot 7 \cdot 31$	$3 \cdot 3271$	$41 \cdot 191$	$7 \cdot 23 \cdot 37$	$3 \cdot 541$
	(26)	(94)	(490)	(2010)	(1924)
Δ_8	-9739	-9719	-7341	-3947	+301
	83739	9719	$3 \cdot 2447$	3947	$7 \cdot 43$
	(70)	(26)	(132)	(552)	(1122)
Δ_9	-9669	-9693	-7209	-3395	+1423
	$3 \cdot 11 \cdot 293$	$3^3 \cdot 359$	$3^4 \cdot 89$	$5 \cdot 7 \cdot 97$	1423
	(14)	(24)	(500)	(1178)	(1500)
Δ_{10}	-9655	-9669	-6709	-2217	+2923
	$5 \cdot 1931$	$3 \cdot 11 \cdot 293$	6709	2217	$37 \cdot 79$
	(4)	(30)	(144)	(3058)	(7320)
Δ_{11}	-9651	-9639	-6565	+841	+10243 > +100000
	$3 \cdot 1931$	$3^4 \cdot 7 \cdot 17$	$5 \cdot 13 \cdot 101$	29^2	10243
	(156)	(94)	(534)	(84)	(432)
Δ_{12}	-9651	-9545	-6031	+925	+10675 > +100000

	2^72	
Δ_{32}	-8489	Anmerkung:
	$13 \cdot 653$	Die "Genese" eines Zwillings mod 2 im Umfeld
	(2)	der geraden Zahl n_g = 2^72. Im Intervall $\Sigma\Delta = 20000$
Δ_{33}	-8487	treten insgesamt 13 Zwillinge mod 2 auf.
	$3^2 \cdot 23 \cdot 41$	

	2^65	
Δ_{59}	-7551	Anmerkung:
	$3^2 \cdot 839$	Die "Genese" eines Zwillings mod 2 im Umfeld der geraden Zahl n_g = 2^65.
	(2)	Im Intervall $\Sigma\Delta = 20000$ treten insgesamt 12 Zwillinge mod 2 auf.
Δ_{59}	-7549	
	7549	

Anmerkungen und Schlussfolgerungen:

1. Die mittlere Primzahl-Dichte fällt im analysierten Intervall (Δ) mit wachsendem Exponenten (n) asymptotisch und kann den Wert Null erreichen, sofern im Intervall Δ= ±10000 keine Primzahlen situiert sind.

2. Innerhalb des analysierten Intervalls Σ Δ = 20000 treten die Primzahlen in den Umgebungen der geraden Zahlen n_g:= 2^65, 2^72, 2^531, 2^2000 und 2^2500 in variierenden Abständen (Δ) auf. Die in **Tab. 62** angegebenen mittleren Primzahl-Abstände (Σ Δ/*P*) weisen mit steigender Potenz (n) der geraden Zahlen ($n_g = 2^n$) die Gesetzmäßigkeit eines stetigen Wachstums auf.

3. Die Abstände der Primzahlen von der (jeweiligen) geraden Zahl $n_g = 2^n$ sind überwiegend durch Primzahl-Faktoren definiert, die in **Tab. 62** aufgeführt wurden, oder aber durch (einzelne) Primzahlen. In den Umgebungen der analysierten geraden Zahlen treten folgende Zahlen von Primzahl-Abständen auf:

- In der Umgebung der Zahl n_g = 2^65 treten 71 Primzahlen in Primzahl-Abständen auf.
- In der Umgebung der Zahl n_g = 2^72 treten 65 Primzahlen in Primzahl-Abständen auf.
- In der Umgebung der Zahl n_g = 2^531 treten 6 Primzahlen in Primzahl-Abständen auf.
- In der Umgebung der Zahl n_g = 2^2000 treten 3 Primzahlen in Primzahl-Abständen auf.
- In der Umgebung der Zahl n_g = 2^2500 treten 3 Primzahlen in Primzahl-Abständen auf.

4. Aus **Tab. 62** kann die "Genese" von Zwillingen mod 2 im Umfeld der geraden Zahlen: n_g = 2^65 und n_g = 2^72 wahrgenommen werden:

<u>Zwilling mod 2 im Umfeld der geraden Zahl n_g = 2^65 = 36893488147419103232 (EZ2):</u>

- 2^65 - (7551 = $3^2 \cdot 839$) - 7551 = **36893488147419095681** <u>Anmerkung:</u> Wie ersichtlich erfolgt hier
- 2^65 - (**7549** = 7549) - **7549** = **36893488147419095683** eine Konvertierung der Zwillingsklasse!

<u>Zwilling mod 2 im Umfeld der geraden Zahl n_g = 2^72 = 4722366482869645213696 (EZ6):</u>

- 2^72 - (8489 = 13 · 653) - 8489 = **4722366482869645205207** <u>Anmerkung:</u> Wie ersichtlich erfolgt
- 2^72 - (8487 = $3^2 \cdot 23 \cdot 41$) - 8487 = **4722366482869645205209** hier keine Klassen-Konvertierung.

Im Gegensatz zu dieser "Genese" von Zwillingen mod 2 kann aus **Tab. 63** die folgende Genese eines Zwillings mod 2 entnommen werden, bei der ein "Klassenerhalt" von Zwillingen mod 2 erfolgt.

<u>Zwilling mod 2 im Umfeld der geraden Zahl n_g = (2 · 3 · 5 · .. · 53) = 32589158477190044771900044730:</u>

- (2 · 3 · 5 · ... · 53) - **8821** = **32589158477190035909** <u>Anmerkung:</u> Wie ersichtlich erfolgt hier keine
- (2 · 3 · 5 · ... · 53) - **8819** = **32589158477190035911** Konvertierung der Klassen-Zugehörigkeit.

<u>Anmerkung:</u>
Der Zwilling mod 2 (**8819, 8821**) ist ein Element der Folge n_g = 30 + n x30, n:= 293
Der Zwilling mod 2 (**32589158477190035909, 32589158477190035911**) ist (hier) ein Element der gleichen Folge: n_g = 30 + n x 30, n = 1086305282573001196.

Für das Verständnis der Unterschiede, die bei der "Genese" von Zwillingen mod 2 auftreten können. wird (noch) der folgende Zwilling mod 2 im Umfeld der geraden Zahl n_g = 2 · 3 · 5 · .. · 53 betrachtet::

- (2 · 3 · 5 · .. · 53) - **9851** = **32589158477190034877**
- (2 · 3 · 5 · .. · 53) - (9853 = 59 · 167) = **32589158477190034879**

<u>Anmerkung:</u>
Hier erfolgt ein "Wechsel" des (generierten) Zwillings mod 2 zu einem Element der Klasse: n_g = 18 + n x 30, n = 1096305282573001162. Dieser Wechsel ist deshalb möglich, weil die Zahlen: (**9851**, 9853) keinen Zwilling mod 2 bilden!

Tab. 63 Die Primzahlen im Intervall $\Delta = \pm 10000$ ($\Sigma\,\Delta = 20000$) im Umfeld der geraden Zahlen, in die faktoriell die Primzahlen p:= 2, 3, 5, ..., 53 eingehen:

n_g:=	Πp_i	$\Pi p_i \cdot 53$	53!	53!^5	53!^10
Dezimalstellen:	20	22	90	349	697
Primzahlen (*P*):	439	380	138	19	10
$P/\Sigma\,\Delta$:	0,02195	0,01900	0,00690	0,00095	0,00050
$\Sigma\,\Delta/P$:	45,5581	52,6316	144,927	1052,63	2000,00
Zwilling mod:	(132)	(68 = -1, +67)	(130)	(1502)	(1414)

Die Primzahl-Abstände (Δ_i) von n_g und ihre Prim-Faktor-Strukturen sowie die aufeinander folgenden Zwillinge:

Δ_1	-9991	-9983	-9767	-8123	-5393
	97 · 101	67 · 149	9767	8123	5393
	(18)	(54)	(28)	(1414)	(1120)
Δ_2	-9973	-9929	-9739	-6709	-4273
	9973	9929	9739	6709	4273
	(6)	(42)	(42)	(480)	(2520)
Δ_3	-9967	-9887	-9697	-6229	-1753
	9967	9887	9697	6229	1753
	(36)	(28)	(84)	(786)	(1320)
Δ_4	-9931	-9859	-9613	-5443	-433
	9931	9859	9613	5443	433
	(78)	(30)	(30)	(782)	(122)
Δ_5	-9853	-9829	-9563	-4661	-311
	59 · 167	9829	73 · 131	59 · 79	311
	(2)	(26)	(354)	(1894)	(1414)
Δ_6	-9851	.-9803	-9209	-2767	+1103
	9851	9803	9209	2767	1103
	(48)	(12)	(166)	(60)	(494)
Δ_7	-9803	-9791	-9043	-2389	+1597
	9803	9791	9043	2389	1597
	(36)	(10)	(2)	(318)	(4776)
Δ_8	-9767	-9781	-9041	-2389	+6373
	9767	9781	9041	2389	8373
	(46)	(60)	(118)	(810)	(2596)
Δ_9	-9721	-9721	-8923	-1597	+8969
	9721	9721	8923	1597	8969
	(2)	(2)	(74)	(156)	(218)
Δ_{10}	-9719	-9719	-8849	-1423	+9187
	9719	9719	8849	1423	89 · 103

Anmerkungen und Schlussfolgerungen:

1. Die Primzahl-Dichte strebt mit wachsender Stellenzahl der geraden Zahlen, die faktoriell aus den Primzahlen: $p_1 = 2$ bis $p_n = 53$ gebildet werden, im betrachteten Intervall $\Sigma\,\Delta = 20000$ asymptotisch nach Null.

2. Alle Primzahlen, die im Intervall $\Sigma\,\Delta = 200000$ um die in **Tab. 63** ausgewiesenen geraden Zahlen vorkommen, treten stets in Primzahl-Abständen zu den geraden Zahlen (n_g) oder aber in Primzahl-Faktor-Abständen auf, die aus Primzahlen $p > p_n = 53$ kreiert sind. Insoweit wird das Gesetz 2 erfüllt. Dabei können (wie aus **Tab. 63** ersichtlich ist) in den Abständen $\Delta \leq 59 \cdot 59 =$ **34811** stets nur Primzahl-Abstände auftreten.

3. Im Umfeld der geraden Zahlen: $n_g = \Pi p_i$ und $n_g = \Pi p_i \cdot 53$treten im Intervall $\Sigma\,\Delta = 200000$ jeweils 11 Zwillinge mod 2 auf. Die Bildungsgesetze dieser Zwillinge wurden in den Anmerkungen der **Tab. 62** bereits ausführlich kommentiert.

Tab. 64 Die Primzahlen im Intervall. Δ= ±10000 (Σ Δ= 20000) im Umfeld der geraden Zahlen, in die faktoriell die Primzahlen: p:= 2, 3, 5, ..., 101 eingehen:

n_g:=	Πp_i	$\Pi p_i \cdot 101$	101!	101!^3	101!^5
Dezimalstellen:	39	41	160	480	800
Primzahlen (*P*):	226	196	59	18	13
P/Σ Δ:=	0,0113	0,0098	0,0080	0,0009	0,00065
Σ Δ/*P*:=	88,496	102,041	338,983	1111,111	1538,462
Zwilling mod:=	(364)	(354)	(618)	(2596)	(2548)

Die Primzahl-Abstände (Δ_i) und die aufeinander folgenden Zwillinge mod 2n, n ≥1:

Δ_1	-9871	-9941	-9857	-8971	-9923
Zwilling mod:	(80)	(84)	(720)	(702)	(2346)
Δ_2	-9791	-9857	-9137	-8269	-7577
Zwilling mod:	(58)	(6)	(1044)	(552)	(418)
Δ_3	-9733	-9851	-8093	-7717	-7159
Zwilling mod:	(44)	(60)	(516)	(170)	(210)
Δ_4	-9689	-9791	-7577	-7547	-6949
Zwilling mod:	(280)	(72)	(324)	(504)	(158)
Δ_5	-9419	-9719	-7253	-7043	-6791
Zwilling mod:	(100)	(378)	(174)	(154)	(2334)
Δ_6	-9319	-9341	-7079	-6899	-4457
Zwilling mod:	(92)	(84)	(80)	(3636)	(246)
Δ_7	-9227	-9257	-7019	-3253	-4211
Zwilling mod:	(124)	(84)	(162)	(1374)	(484)
Δ_8	-9103	-9173	-6857	-1879	-3727
Zwilling mod:	(62)	(124)	(28)	(348)	(1290)
Δ_9	-9041	-9049	-6829	-1531	-2437
Zwilling mod:	(78)	(350)	(530)	(368)	(1976)
Δ_{10}	-8963	-8699	-6299	-1163	-461
Zwilling mod:	(144)	(22)	(346)	(2596)	(2548)
Δ_{11}	-8819	-8677	-5953	+1433	+2087
Zwilling mod:	(56)	(50)	(1020)	(944)	(866)
Δ_{12}	-8737	-8627	-4933	+2377	+2953
Zwilling mod:	(28)	(54)	(310)	(342)	(504)
Δ_{13}	-8269	-8573	-4483	+2719	+3457
Zwilling mod:	(146)	(210)	(860)	(1918)	------- (Intervall-Ende)
Δ_{14}	-8123	-8363	-3623	+4637	
Zwilling mod:	(244)	(52)	(94)	(2382)	
Δ_{15}	-7879	-8311	-3529	+7019	
Zwilling mod:	(2)	(24)	(168)	(854)	
Δ_{16}	-7877	-8287	-3361	+7873	
Zwilling mod:	(204)	(50)	(48)	(796)	
Δ_{17}	-7673	-8237	-3313	+8669	
Zwilling mod:	(144)	(58)	(546)	(492)	
Δ_{18}	-7529	-8179	-2767	+9161	
Zwilling mod:	(42)	(68)	(36)	------- (Intervall-Ende)	

Anmerkungen und Schlussfolgerungen:

1. In **Tab. 63** wird das Gesetz der asymptotischen Abnahme der Primzahl-Dichten für die geraden Zahlen in den die Primfaktoren $p_1 = 2$ bis $p_n = 101$ in steigenden Potenzen Auftreten vollauf bestätigt.

2. Innerhalb des analysierten Intervalls Σ Δ= 200000 treten Zwillinge mod 2n, n≥1 in stark fluktuierenden Größen auf - dies ist ein Sachverhalt, der sich bereits in kleineren Zahlenbereichen manifestiert.

3. Alle in **Tab. 63** auftretenden Abstände von den geraden Zahlen (n_g) werden aus Primzahlen gebildet, da Primfaktor-Abstände erst in Bereichen von Δ> 103 · 103 = 10609 auftreten können.

Das in **Tab. 62**, **Tab. 63** und **Tab. 64** nachgewiesene Gesetz der asymptotischen Abnahme der Primzahl-Dichten in der Umgebung gerader Zahlen, in denen die diese Zahlen bildenden Primzahlen in wachsenden Potenzen auftreten,, wird in **Tab. 65**, **Tab. 66** und in **Tab. 67** für die ungeraden Zahlen: $n_u = 3^n$, 5^n und 7^n, die in steigenden Potenzen auftreten, bestätigt.
Die in diesen Tabellen angeführten Ergebnisse können vom Leser unter Einsatz des MuPAD-Programms nachvollzogen werden.

Tab. 65 Die Primzahlen im Intervall Δ= ±10.000 (Σ Δ= 20.000) im Umfeld der ungeraden Zahlen n_u = 3^n:

n_u = 3^n::	3^100	3^500	3^1000	3^1500
Dezimalstellen:	48	234	480	716
Primzahlen (*P*):	182	33	21	13
P/Σ Δ:	0,00910	0,00165	0,00105	0,00065
Σ Δ/*P*:	109,890	606,061	952,381	1538,462
Zwilling mod:	(158)	(640)	(2052)	(844)

Anmerkungen und Schlussfolgerungen;

1. Auf die Angabe der Abstände (Δ_i), in denen die Primzahlen (*P*) von den jeweiligen ungeraden Zahlen n_u: = 3^100, 3^500, 3^1000 und 3^1500 auftreten, wurde in Tab. 65 verzichtet. Gleichfalls wurde auf die Angabe der Prim-Faktoren verzichtet, aus denen die (jeweiligen) Abständen (Δ_i) bestehen.
Im intervall (Σ Δ) treten vereinzelt Zwillinge mod 2 auf. Die hier auftretenden Primzahlen (*P*) kommen in stark variierenden Abständen vor. In den unterstrichenen Zwillingen sind die Zahlen (n_u) situiert.

2. In **Tab. 65** wird das Gesetz der asymptotischen Abnahme der Primzahl-Dichten (vollauf) bestätigt.

Tab. 66 Die Primzahlen im Intervall Δ= ±10.000 (Σ Δ= 20.000) im Umfeld der ungeraden Zahlen n_u = 5^n:

n_u = 5^n:	5^75	5^750	5^850
Dezimalstellen:	52	446	595
Primzahlen (*P*):	137	19	16
P/Σ Δ:	0,00685	0,00095	0,00080
Σ Δ/*P*:	145,985	1052,632	1250,000
Zwilling mod:	(272)	(3198)	(3054)

Anmerkungen und Schlussfolgerungen:

1. Auf die Angabe der Abstände (Δ_i), in denen die Primzahlen (*P*) von den jeweiligen ungeraden Zahlen: n_u:= 5^75, 5^750 und 5^850 auftreten, wurde in **Tab. 66** verzichtet. Gleichfalls wurde auf die Angabe der Prim-Faktoren verzichtet, aus denen die (jeweiligen) Abstandszahlen (Δ_i) bestehen.
Im Intervall (Σ Δ) treten vereinzelt Zwillinge mod 2 auf.

2. In Tab. 66 wird das Gesetz der asymototischen Abnahme der Primzahl-Dichten (vollauf) bestätigt.

Tab. 67 Die Primzahlen im Intervall Δ= ±10.000 (Σ Δ= 20.000) im Umfeld der ungeraden Zahlen n_u = 7^n:

n_u = 7^n:	7^60	7^400	7^500	7^550
Dezimalstellen:	51	339	426	634
Primzahlen (*P*):	163	25	19	18
P/Σ Δ:	0,00815	0,00125	0,0,00095	0,00090
Σ Δ/*P*:	122,699	800,000	1052,632	1111,111
Zwilling mod:	(198)	(4764)	(3908)	(2780)

Anmerkungen und Schlussfolgerungen:

1. In Tab. 67 wird das Gesetz der asymptotischen Abnahme der Primzahl-Dichten (vollauf) bestätigt.

2. Im Intervall (Σ Δ) treten vereinzelt Zwillinge mod 2 auf.

Die zweite Aufgabe, die in diesem Kapitel zu einer näheren Untersuchung ansteht, ist den Schwankungen der Dichten der Zwillinge mod 2 und mod 2n, n>1 gewidmet, wobei aus den Daten, die in den Tabellen: **Tab. 61** bis **Tab. 67** gewonnen wurden, die Einsicht hervorgeht, dass diese Zwillinge in größeren Intervallen: $\Sigma\Delta$ analysiert werden müssen, um eine Gesetzmäßigkeit ihrer asymptotischen Abnahme in hohen Zahlenbereichen ($n \to \infty$) zu erkennen.

Eine derartige Gesetzmäßigkeit konnte von den großen Mathematikern, die sich mit den Problemen der Primzahlen befasst hatten, nicht gefunden werden, da die Auffindung dieser Gesetzmäßigkeiten den Einsatz schneller Rechner und geeigneter Test-Programme erfordert, wie auch die Kenntnis der Bildungsgesetze der Zwillinge mod 2n, n ≥1. Diese Bildungsgesetze wurden in dieser Monographie in **Kapitel 7** und in **Kapitel 8** deduziert und in **Tab. 46** zusammengefasst.

Zunächst wird die asymptotische Abnahme der Zahlen der Zwillinge mod 2 in steigenden Zahlen-Intervallen untersucht, wobei ein konstantes Intervall von $\Delta = \pm 300.000$ ($\Sigma\Delta = 600.000$) angenommen wird, das in immer höhere Zahlenbereiche verschoben wird.

In **Tab. 68** wird der Verlauf der Dichten der Zwillinge mod 2 für die im Intervall $\Sigma\Delta = 600.000$ für die Zahlen $n_g = \Pi p_i = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 23 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47\ \ 53$ und nachfolgend für die Zahlen: $n_g = 53!,\ 53! \cdot 53,\ 53! \cdot 53!,\ 53! \cdot 53! \cdot 53$ und 53!^6 untersucht.

Den Lesern, die die in **Tab. 68** angeführten Ergebnisse überprüfen möchten, wird nachfolgend das verwendete MuPAD-Testprogramm vorgestellt:

Folge 1: for i from +(Πp_i/30 -10000) to +(Πp_i/30 +10000) do: if (isprime (11+30*i) and isprime (13+30*i)) then print (("11", 11+30*i, " = ", 11+30*i, " and 13", 13+30*i)) end_if: end_for:

Folge 2: for i from +(Πp_i/30 -10000) to +(Πp_i/30 +100000) do: if (isprime (17+30*i) and isprime (19+30*i)) then print (("17", 17+30*i, " = " 17+30*i, " and 19", 19+30*i)) end-if: end_for:

Folge 3: for i from +(Πp_i/30 -10000) to +(Πp_i/30 + 100000) do: if (isprime (29+30*i) and isprime (31+30*i) then print (("29", 29+30*i, " = " 29+30*i, " and 31", (31+30*i)) end_if: end_for:

Tab. 68 Zahl der Zwillinge mod 2 in : *Folge 1*, *Folge 2* und *Folge 3* und in der Summe auftreten:

n_g:=	Dezimalstellen:=	*Folge 1*	*Folge 2*	*Folge 2*	Summe:=
$n_g = \Pi p_i$	20	131	114	137	382
$n_g = \Pi p_i \cdot 53$	22	119	113	115	347
$n_g = 53!$	70	7	8	12	27
$n_g = 53! \cdot 53$	72	17	18	15	50
$n_g = 53! \cdot 53!$	140	0	1	2	3
$n_g = 53! \cdot 53! \cdot 53$	141	1	3	5	9
n_g = 53!^6	418	0	0	0	0

<u>Anmerkungen und Schlussfolgerungen:</u>

1. Die Zahl der Zwillinge mod 2 strebt (mit steigender Zahl der Dezimalstellen) der Zahlen n_g (variierend) im Intervall $\Sigma\Delta = 60.000$ dem Grenzwert Null zu - dies bedeutet, dass Zwillinge mod 2 (dann nur) innerhalb größerer Intervalle $\Sigma\Delta > 60.000$ auftreten können.

In **Tab. 69** wurden die Zahlen der Zwillinge mod 2 angegeben, die im Intervall $\Sigma\Delta$= 600.000 in den Umgebungen der geraden Zahlen n_g = Πp_i= 2 ·3 ·5 ·7 ·... ·101, n_g = 101! und n_g = 101! ·101! und 101!^3 auftreten.

Tab. 69 Die Zahl der Zwillinge mod 2 in *Folge 1*, *Folge 2*. Folge 3 und in der Summe auftreten:

n_g:=	Dezimalstellen:	*Folge 1*	*Folge 2*	*Folge 3*	Summe:=
$n_g = \Pi p_i$	38	62	76	80	218
n_g = 101!	80	3	2	4	9
n_g = 101! ·101!	160	2	2	0	4
n_g = 101!^3	480	0	0	0	0

Anmerkungen und Schlussfolgerungen:

1, Die Zahl der Zwillinge mod 2 strebt (mit steigender Zahl der Dezimalstellen) der Zahlen n_g im Intervall $\Sigma\Delta$= 600.0000 asymptotisch dem Grenzwert Null zu - dies bedeutet, dass Zwillinge mod 2 in noch höheren Zahlenbereichen nur innerhalb größerer Intervalle $\Sigma\Delta$> 600.0000 auftreten können.

In Tab. 70 wurden die Zahlen der Zwillinge abgeleitet, die im Intervall $\Sigma\Delta$= 600.000 in den Umgebungen der Zahlen $n_g = 2^{65}$, 2^{72}, 2^{500}, 2^{750}, 2^1000, 2^1195 und 2^1200 auftreten, sowie in den Umgebungen der Zahlen $n_u = 3^{100}$, 3^{300} und 3^{600} situiert sind.

Tab. 70 Die Zahl der Zwillinge mod 2, die in *Folge 1*, *Folge 2*, *Folge 3* und in der Summe auftreten:

n_g:=	Dezimalstellen;	*Folge 1*	*Folge 2*	*Folge 3*	Summe:=
n_g = 2^72	22	87	96	104	287
n_g = 2^250	76	8	8	5	21
n_g = 2^500	151	1	3	0	4
b_g = 2^750	226	1	2	1	4
n_g = 2^1000	302	2	0	0	2
n_g = 2^1195	360	0	1	0	1
n_g = 2^1200	362	0	0	0	0
n_g = 3^100*30	49	21	13	13	47
n_g = 3^300*30	145	4	1	5	10
n_g = 3^600*30	288	1 *)	0	0	1

Anmerkungen und Schlussfolgerungen:

1. Die Zahl der Zwillinge mod 2 strebt (mit steigender Zahl der Dezimalstellen) auch im Umfeld der geraden Zahlen n_g = 2^n und der ungeraden Zahlen 3^n im Intervall $\Sigma\Delta$= 600.000 asymptotisch dem Grenzwert Null zu - dies bedeutet, dass Zwillinge mod 2 in höheren Zahlenbereichen nur innerhalb größerer Intervalle auftreten.

2. Der einzige Zwilling mod 2, der im Intervall $\Sigma\Delta = 600.000$ im Umfeld der ungeraden Zahl 3^600 in der *Folge 1* vorkommt, ist ein Element der Folge $n_g = 12 + n \times 30$, Die Zahl n = (3^600+ 2070) x 30). hat 288 Dezimalstellen.

In den nachfolgenden Tabellen wird der Nachweis erbracht, dass die Zahl der Zwillinge mod 4, mod 10, mod 100 (verallgemeinert der Zwillinge mod 2n, n > 1) in zunehmend steigenden Zahlenbereichen ($n \to \infty$) in definierten Intervallen ebenfalls asymptotisch dem Wert Null zustreben.
In **Tab. 71** wurden die Zahlen der Zwillinge mod 4 bestimmt, die im Intervall $\Sigma\Delta = 600.000$ in steigenden Zahlenbereichen auftreten. In **Tab. 72** wurden die Zahlen der Zwillinge mod 10 und in **Tab. 73** die Zahlen der Zwillinge mod 100 im gleichen Intervall ($\Sigma\Delta = 600.000$) bestimmt.

Tab. 71 Die Zahl der Zwillinge mod 4, die in *Folge 1*, *Folge 2* und *Folge 3* und in der Summe auftreten:

n_g =	Dezimalstellen:	*Folge 1*	*Folge 2*	*Folge 3*	Summe:=
$n_g = \Pi(2 \cdot 3 \cdot .. \cdot 53)$	20	115	114	130	359
$n_g = 53!$	70	14	9	14	37
$n_g = 53! \cdot 53$	72	9	10	9	28
n_g = 53!^2	140	5	1	5	11
n_g = 53!^2 ·53	141	2	2	5	9
n_g = 53!^3	210	2	1	1	4
n_g = 53!^4	280	1	1	1	3
n_g = 3^100*10	49	23	19	17	49
n_g = 3^300*10	145	4	1	5	10
n_g = 3^600*10	288	0	0	0	0

Anmerkungen und Schlussfolgerungen:

1. Die Asymptote-Gesetzmäßigkeit wird in Tab. 71 für die Zwillinge mod 4 bestätigt, da die Zahl der Zwillinge mod 4 im Intervall $\Sigma\Delta = 600.000$ in immer größeren Zahlenbereichen ($n \to \infty$) nach Null strebt.
2. In den Intervallen $\Sigma\Delta = 600.000$ treten in wachsenden Zahlenbereichen ($n \to \infty$) auch immer geringere Primzahl-Dichten von Zwillingen mod 2n, n > 1 auf.

Tab. 72 Die Zahl der Zwillinge mod 10, die *in Folge 1, Folge 2, Folge 3 und Folge 4* auftreten:

n_g:=	Dezimalstellen:	*Folge 1*	*Folge 2*	*Folge 3*	*Folge 4*	Summe:=
$n_g = 53!$	70	11	11	7	17	46
n_g = 53!^2	140	0	3	1	1	5
n_g = 53!^4	280	1	1	2	1	5

Anmerkungen und Schlussfolgerungen:

1. In **Tab. 72** wird die Asymptote-Gesetzmäßigkeit bestätigt, verbunden mit dem Hinweis, dass die Zahl der Zwillinge mod 10 im Intervall $\Sigma\Delta = 600.000$ im Zahlenbereich ($n \to \infty$) sich variierend dem Grenzwert Null nähern kann.

Tab. 73 Die Zahl der Zwillinge mod 100, die in *Folge 1*, *Folge 2*, *Folge 3* und *Folge 4* auftreten:

n_g:=	Dezimalstellen:	*Folge 1*	*Folge 2*	*Folge 3*	*Folge 4*	Summe:=
n_g = = $\Pi(2 \cdot .. \cdot 53)/10$	19	142	98	125	155	520
n_g = 53!/10	70	6	8	10	10	34
n_g = 101!/10	160	2	1	1	1	5
n_g = 101!^2/10	320	0	0	0	1	1
n_g = 101!^3/10	480	0	1	0	0	1

Anmerkungen und Schlussfolgerungen:

1. In **Tab. 73** ist die Geltung der Asymptote-Gesetzmäßigkeit für die Zwillinge mod 100 nachgewiesen. Diese Gesetzmäßigkeit wird (wie ersichtlich) von allen Zwillingen mod 2n, n ≥1 erfüllt.

Das Asymptote-Gesetz der Zwillinge mod 2n, n ≥1, das auf der Basis der Bildungs-Gesetze dieser Zwillinge gefunden werden konnte, erweitert das Gesetz der variierenden Abnahme der Primzahl-Dichten in immer größeren Zahlen-Bereichen (n $\to \infty$), das in diesem Kapitel ausführlich untersucht wurde.

18. Die Bestimmung von Primzahlen der Typen SM_p, SF_p und CM_p, CF_p:

Die in **Kapitel 11** angestellten Untersuchungen, die dem auftreten von Primzahlen der Typen SM_p und SF_p gewidmet waren, werden in diesem Kapitel (wesentlich) vertieft.
Mit den heute verfügbaren Rechenprogrammen von MuPAD kann (leicht) der Nachweis erbracht werden, dass es wesentlich mehr Primzahlen vom Typ: SM_p und SF_p gibt, als die bis heute bekannten Primzahlen M_p (von denen bisher 42 gefunden wurden) und F_p (von denen nur 5 bekannt sind).

Aus der in dieser Monographie eingeführten Klassenteilung der geraden Zahlen $n_g = 2^n$ in die vier zyklisch aufeinander folgenden Klassen: $n_g = 2^{1+4n}$ (EZ2), $n_g = 2^{2+4n}$ (EZ4), $n_g = 2^{3+4n}$ (EZ8) sowie $n_g = 2^{4+4n}$ (EZ6) konnten die tieferen Gründe erkannt werden, warum die Primzahlen M_p nur in den Endziffern: EZ1 und EZ7 auftreten können und die Primzahlen F_p stets nur in der Endziffer EZ7 - siehe **Tab. 28**. Des weiteren wurde bereits unter Beweis gestellt, dass die Primzahlen M_p und F_p mit Ausnahme des Zwillimgs mod 2 (2^2 -1, 2^2 +1 = 3, 5) keine weiteren Zwillinge mod 2 bilden können. Die Primzahlen vom Typ SM_p und SF_p können dagegen in allen vier Endziffern: EZ1, EZ3, EZ7, EZ9 auftreten, aber niemals Zwillinge mod 2 bilden, dagegen jedoch oft symmetrisch in Zwillingen mod 4 situiert sein.. Des weiteren besteht die Möglichkeit der Bildung von Primzahlen SM_p und SF_p mit definierten (gewünschten) Endziffern. Für das Verständnis dieser Eigenschaften der Primzahlen SM_p und SF_p ist eine Analyse der Klassen erforderlich, in denen die nicht primen ungeraden Zahlen (n_u) auftreten können. Dieser Aufgabe ist die nachfolgende **Tab. 74** gewidmet.

Tab. 74 Die Klassen in denen die Potenzen ungerader Zahlen (n_u) auftreten:

n_u :=	1 + 4n	EZ.=	2 + 4n	EZ:=	3 + 4n	EZ:=	4 + 4n	EZ:=	n:=
n_u = 3	3	EZ3	9	EZ9	27	EZ7	81	EZ1	0
	243	EZ3	729	EZ9	2187	EZ7	6561	EZ1	1
	19683	EZ3	59049	EZ7	177147	EZ7	531441	EZ1	2

Anmerkung: Die ungeraden Zahlen n_u = 3^n treten in vier Klassen mit den Endziffern: EZ3, EZ9, EZ7, EZ1 auf.

Tab. 74 Fortsetzung

n_u :=	1 + 4n	EZ:=	2 + 4n	EZ:=	3 + 4n	EZ:=	4 + 4n	EZ:=	n:=
$n_u = 5$	5	EZ5	25	EZ5	125	EZ5	625	EZ5	0
	3125	EZ5	15625	EZ5	78125	EZ5	390625	EZ5	1

Anmerkung: Die ungeraden Zahlen $n_u = 5^n$ treten allesamt in der Endziffer EZ5 auf.

$n_u = 7$	7	EZ7	49	EZ9	343	EZ3	2401	EZ1	0
	16807		117649		823543		5764801		1

Anmerkung: Die ungeraden Zahlen $n_u = 7^n$ treten in vier Klassen mit den Endziffern: EZ7, EZ9, EZ3, EZ1 auf.

$n_u = 11$	11	EZ1	121	EZ1	1331	EZ1	14641	EZ1	0
	161051		1771561		19487171		214358881		1

Anmerkung: Die ungeraden Zahlen $n_u = 11^n$ treten allesamt in der Endziffer EZ1 auf.

$n_u = 13$	13	EZ1	169	EZ9	2197	EZ7	28561	EZ1	0
	371293		4826809		62748517		815730721		1

Anmerkung: Die ungeraden Zahlen $n_u = 13^n$ treten in vier Klassen mit den Endziffern:EZ3, EZ9, EZ7, EZ1 auf.

$n_u = 17$	17		289		4913		83521		0
	1419857		24137569		410338673		6975757441		1

Anmerkung: Die ungeraden Zahlen $n_u = 17^n$ haben eine gleiche Klassenteilung in vier Klassen mit der Folge der Endziffern: EZ7, EZ9, EZ3, EZ1 wie die ungeraden Zahlen $n_u = 7^n$.

$n_u = 19$	19		361		6859		130321		0
	2476099		47045881		893871739		16983563041		1

Anmerkung: Die ungeraden Zahlen $n_u = 19^n$ treten in zwei Klassen mit den Endziffern: EZ9, EZ1 auf.

Anmerkungen und Schlussfolgerungen:

1. Aus **Tab. 74** folgt, dass alle Potenzen der Primzahlen $p \geq 3$ mit den Endziffern: 1, 3. 7, 9 eine gleiche Teilung in Klassen mit den folgenden zyklisch aufeinander folgenden Endziffern vermitteln:

EZ1:	p:= 11, 31, 41, 71, 101, ...
EZ3, EZ9, EZ7, EZ1	p:= 3, 13, 23, 53. 73, 103, ...
EZ5	p = 5 und alle $n_u = 5\times(p_u := 3, 7, 11, ...)$
EZ7, EZ9, EZ3, EZ1	p:= 7, 17, 37, 47, 67, 97, ...
EZ9, EZ1	p:= 19, 29, 59, 79, 109, ...

2. Die hier nachgewiesene Klassenteilung der ungeraden Zahlen der Potenzen: p^n kommt eine wichtige Bedeutung für die Generierung von Primzahlen der Typen: SM_p und SF_p.

3. Zunächst ist es (leicht) verständlich, dass sich aus ungeraden Zahlen: n_u mit der Endziffer EZ7 niemals Primzahlen vom Typ SM_p ergeben können, da ungerade Zahlen mit der EZ7 nach Abzug der Zahl 2 zu ungeraden nicht prime Zahlen mit der Endziffer EZ5 mutieren. Aus ungeraden nicht primen Zahlen mit der Endziffer EZ7 können hingegen zu Primzahlen vom Typ SF_p mutieren.

4. Alle ungeraden Zahlen (n_u), mit der Endziffer EZ3 können niemals zu Primzahlen vom Typ SF_p mutieren, da sich durch die Addition der Zahl $n_g = 2$ stets nicht prime ungerade Zahlen mit der Endziffer EZ5 gebildet werden.

5. Ungeraden Zahlen (n_u) mit den Endziffern: EZ1, EZ5 und EZ9 können hingegen (fallweise) durch die Addition bzw. Subtraktion der Zahl $n_g = 2$ zu Primzahlen der Typen: SF_p und SM_p mutieren und dabei fallweise. Zwillinge mod 4 ergeben.

6. Der hier festgestellte Sachverhalt, dass nur ungerade nicht ptime Zahlen n_u mit den Endziffern: EZ5, EZ9 und EZ1 inmitten von Zwillingen mod 4 situiert sein können, findet seine Bestätigung in den Bildungsgesetzen der *Folge 1*, *Folge 2* und der *Folge 3* der Zwillinge mod 4 - siehe **Tab. 26**.

Unter Einsatz der nachfolgenden MuPAD-Test - Methode wurden in **Tab. 75** zahlreiche Primzahlen vom Typ: SM_p und SF_p generiert:

Test für Primzahlen vom Typ SM_p: for i from $-n_{g\,1}$ to $+n_{g\,2}$ do: if isprime (p^i -2) then print ("i", p^i, i, " = " p^i - 2) end_if: end_for:

Test für Primzahlen vom Typ SF_p: for i from $-n_{g\,1}$ to $+n_{g\,2}$ do: if isprime (p^i+2) then print ("i", p^i, " = " p^i +2) end_if: end_for:

In **Tab. 75** wurden auch die Potenzen ungerader nicht primer Zahlen ($n_u = \Pi p_i$, p_i := 3, 5, ...) in die o.g. Testformeln eingesetzt, die die in **Tab. 74** deduzierten Randbedingungen erfüllen.

Für ungerade Zahlen, die in Zwillingen mod 4 situiert sein können, wurden schließlich in **Tab. 76** nach der folgenden MuPAD-Test - Methode zahlreiche Primzahlen: SM_p und SF_p bestimmt, die innerhalb von Zwillingen mod 4 - vom Typ: (SM_p, SF_p) situiert sind:

Test für Zwillinge mod 4: for i from $-n_{g\,1}$ to $+n_{g\,2}$ do: if (isprime (n_u^i -2) and isprim (n_u^i +2)) then print (("i", n_u^i, i, " = ", n_u^i ", n_u^i +2)) end_if: end_for:

Tab. 75 Die Bestimmung von Primzahlen vom Typ SM_p und SF_p:

n_u:=	:EZ:	SM_p =	Dezimalstellen:=	n_u :=	EZ:	SFp:=	Dezimalstellen
3^2 - 2	7	<u>7</u>	1	3^1 +2	5	5	1
3^4 - 2	9	<u>79</u>	2	3^2 +2	1	<u>11</u>	2
3^5 - 2	1	241	3	3^3 +2	9	29	2
3^6 -2	7	727	3	3^4 +2	3	<u>83</u>	2
3^9 -2	1	19681	5	3^8 +2	3	6563	4
3^22 - 2	7	31381059607	11	3^10 +2	1	59051	5
3^37 - 2	1		18	3^14 +2	1	4782971	7
3^41 - 2	1		20	3^15 +2	9	14348909	8
3^90 - 2	7		43	3^24 +2	3	282429536483	12
3^102 - 2	7		49	3^36 +2	3		18

Tab. 75 Fortsetzung

n_u :=	EZ:	SM_p	Dezimalstellen	n_u :=	EZ:	SF_p	Dezimalstellen
3^105 - 2	1		51	3^63 +2	9		31
3^317 - 2	1		152	3^98 +2	1		47
3^520 - 2	9		249	3^110 +2	1		53
3^541 - 2	1		259	3^123 +2	9		59
3^561 - 2	1		268	3^126 +2	1		61
3^648 - 2	9		310	3^139 +2	9		67
3^780 - 2	9		373	3^235 +2	9		113
3^786 - 2	7		380	3^243 +2	9		116
3^957 - 2	1		458	3^315 +2	9		151
3^1353 - 2	1		646	3^363 +2	9		174
3^2224 - 2	9		1062	3^391 +2	9		176
				3^494 +2	1		236
				3^1131 +2	9		540
				3^1220 +2	3		583

Anmerkungen:
Im betrachteten Intervall treten wesentlich mehr Primzahlen SF_p auf. Die Primzahlen $SM_p = 7$, $SF_p = 11$ und $SM_p = 79$, $SF_p = 83$ bilden Zwillinge mod 4. in *Folge 2.* und *Folge 3.*

---

n_u :=	EZ:=	SM_p	Dezimalstellen	n_u :=	EZ:	SF_p	Dezimalstellen:
5^1 -2	3	3	1	5^1 +2	7	7	1
5^2 -2	3	23	2	5^3 +2	7	127	3
5^14 - 2	3	6103515623	10	5^17 +2	7	762939453127	12
5^26 - 2	3		19	5^143 +2	7		99
5^50 - 2	3		35	5^261 +2	7		103
5^126 - 2	3		88	5^551 +2	7		386
5^144 - 2	3		100				
5^260 - 2	3		182				
5^624 - 2	3		437				
5^1424 -2	3		996				

Anmerkungen:

1. Im Exponenten-Intervall: 1 bis 2000 treten 10 Primzahlen vom Typ SM_p auf und nur 6 Primzahlen vom Typ SF_p.
2. In diesem Intervall ist ein Zwilling mod 4 ($SM_p = 3$, $SF_p = 7$) situiert.

Anmerkungen und Schlussfolgerungen:

1. Alle ungeraden Zahlen n_u = 5^n treten in der Endziffer EZ5, so dass alle Primzahlen SM_p in der Endziffer EZ3 auftreten und alle Primzahlen SF_p in der Endziffer EZ7.

2. Alle Zwillinge mod 4, die in der Folge 5^n -2 und 5^n +2 auftreten sind Elemente der *Folge 1.*

Tab. 75 Fortsetzung

n_u :=	EZ:	SM_p	Dezimalstellen:	n_u :=	EZ:	SM_p	Dezimalstellen:
7^1 - 2	7	7	1	7^2 -2	9	49	2
7^4 - 2	9	2399	4	7^7 -2	1	823541	6
7^8 - 2	9	5764799	7	7^12 -2	9	13841287199	11
7^15 - 2	1	4747561509941	13	7^28 -2	9		24
7^31 - 2	1		27	7^84 -2	1		71
7^98 - 2	7		83	7^128 -2	9		97
7^238 -2	7		202	7^302 -2	7		256
7^859 -2	1		726	7^1508 -2	9		1275
7^1586 -2	7		1420				

Anmerkungen:

1. Im Exponenten-Intervall von 1 bis 20000 treten 17 Primzahlen vom Typ SM_p mit den Endziffern: EZ1, EZ3, EZ7, EZ9 auf. .

2. Primzahlen vom Typ SF_p = 7^n + 2 können hingegen nicht auftreten, da alle ungeraden Zahlen hier entweder in der Endziffer: EZ5 auftreten oder durch 3 teilbar sind.

Tab. 75 Fortsetzung

n_u :=	EZ:	SM_p	Dezimalstellen	n_u :=	EZ:	SF_p	Dezimalstellen:
11^4 - 2	9	14639	5	11^1 +2	3	13	2
11^6 -2	9	1771559	7	11^9 +2	3	2357947693	10
				11^287 +2	3		220

Anmerkungen:

1. Die Primzahlen SM_p und SF_p wurden hier nur im Exponenten-Intervall: 1 bis 1000 analysiert.

2. Es treten (hier) nur zwei Primzahlen SM_p mit der Endziffer EZ9 auf und drei Primzahlen SF_p mit der Endziffer EZ3.

3. Nachdem alle ungeraden Zahlen 11^n in der Endziffer EZ1 auftreten, müssen (offensichtlich) alle Primzahlen SM_p = 3^n -2 in der Endziffer EZ9 und alle Primzahlen SF_p = 11^n +2 in der Endziffer EZ3 auftreten.

4. In höheren Zahlenbereichen können Zwillinge mod 4 vom Typ (SM_p, SF_p) auftreten, die Elemente der *Folge* 3 mit den Endziffern: EZ9, EZ3 sind. Ein MuPAD-Test ergab, dass im Exponenten-Intervall von 11^1 bis 11^2000 keine Zwillinge mod 4 situiert sind.

5. Durch MuPAD-Teste kann man jedoch feststellen, das die Zahlen 11^n in Zwillingen mod 6 und mod 8 in diesem Exponenten-Intervall situiert sein können. Beispiele sind: 11^1 -4, 11^1 +2 (Zwilling mod 6) und 11^1 - 4, 11^1 +2 (Zwilling mod 6). und 11^1 -6, 11^1+2 (ein Zwilling mod 8).

Tab. 75 Fortsetzung

n_u :=	EZ:	SM_p	Dezimalstellen:	n_u :=	EZ:	SM_p	Dezimalstellen:
13^1 -2	1	11	2	13^2 -2	7	167	3
13^4 -2	9	28559	5	13^5 -2	1	371291	6
13^12 -2	9	23298085122479	9	13^78 -2	7		87
13^80 -2	9		90	13^90 -2	7		101
13^117 -2	1		131	13^120 -2	1		134
13^813 -2	1		758				

Anmerkungen:

1. Im Exponentenbereich von n =1 bis 1000 sind 11 Primzahlen vom Typ SM_p = 13^n -2 aituiert.

2. Primzahlen vom Typ SF_p = 13^n +2 können nicht auftreten, da diese ungeraden Zahlen entweder die Endziffer EZ5 aufweisen, oder aber ihre Quersummen durch 3 teilbar sind.

Tab. 75 Fortsetzung

n_u :=	EZ:	SM_p	Dezimalstellen:	n_u :=	EZ:	SF_p	Dezimalstellen.
17^6 -2	7	24137567	8	17^1 +2	9	19	2
17^24 -2	9		30	17^105 +2	9		130
17^30 .2	7		37	17^369 +2	9		376
17^106 -2	7		131				
17^184 -2	9		227				
17^232 -2	9		286				
17^460 -2	9		567				

Anmerkungen:

1. Im Exponentenbereich n = 1 bis 1000 sind 7 Primzahlen SM_p auf und 3 Primzahlen SF_p

2. Die Zahlen n_o = 17^n können fallweise in Zwillingen mod 6 bzw. mod 8 situiert sein - z.B. bilden die Zahlen 17^1 -4 und 17^1 +2 den Zwilling mod 6 (13, 19)

Die Schlussfolgerungen aus **Tab. 75**:

1- Aus **Tab. 75** kann zunächst die (wichtige) Schlussfolgerung abgeleitet werden, dass im Bereich der natürlichen Zahlen ($n \to \infty$) Primzahlen der Typen SM_p und SF_p (die in dieser Monographie als (Semi - Mersennesche und Semi - Fermatsche Zahlen bezeichnet wurden) in einer dominant höheren Zahl auftreten, als die Primzahlen M_p und F_p (von denen bisher nur 42 und 5 bestimmt wurden). Die tiefere Ursache für diesen (einsichtigen) Sachverhalt ist dadurch gegeben, dass es wesentlich mehr ungerade Zahlen (n_u) und ungerade Primzahlen ($p \geq 3$) gibt als die in der Zahlenmenge ($n \to \infty$) nur verstreut auftretenden geraden Zahlen $n_g = 2^n$ gibt.

2. In **Tab. 76** wird demonstriert, dass Primzahlen vom Typ: SM_p und SF_p im Umfeld beliebig großer ungerader Zahlen $n_u = \Pi p_i (p_i \geq 3)$ und im Umfeld der Potenzen: (Πp_i)^n dieser Zahlen auftreten.

3. Für die Suche nach (noch größeren) Primzahlen als die bekannte Primzahl $M_{p=42}$ eignet sich deshalb eine Test-Methode, die auf der Suche nach immer größeren Primzahlen der Typen: SM_p und SFp aufbaut und auf der Methode des Rest-Moduls Null aufbaut.

Tab. 76 Die Auffindung von Primzahlen SM_p und SF_p im Umfeld der Potenzen ungerader Zahlen: $n_u = 69 = 3 \cdot 23$ und $n_u = 231 = 3 \cdot 7 \cdot 11$

$n_u = 69$	EZ:	SM_p	Dezimalstellen:
69^1 -2	7	67	2
69^2 -2	9	4759	4
69^13 -2	7		24
69^47 -2	7		87
69^96 -2	9		177
69^111 -2	7		205
69^127 -2	7		234
69^159 -2	7		293
69^161 -2	7		297
69^249 -2	7		458

$n_u = 69$	EZ:	SF_p	Dezimalstellen:
69^1 +2	1	71	2
69^3 +2	1	328511	6
69^4 +2	3	22667123	8
69^13 +2	1		24
69^43 +2	1		80

Anmerkungen:

1. Im Exponenten-Intervall n = 1 bis 500 treten 10 Primzahlen. SM_p und 5 Primzahlen SF_p auf.
2. Die unterstrichenen Primzahlen: 69^1 -2, 69^1 +2 und 69^13 -2, 69^13 + 2 bilden Zwillinge mod 4, die Elemente der *Folge 2* sind.

$n_u = 231$	EZ:	SM_p	Dezimalstellen:
231^1 -2	9	229	3
231^2 -2	9	53359	5
231^6 -2	9		15
231^9 -2	9		22
231^29 -2	9		69
231^31 -2	9		74
231^44 -2	9		104
231^47 -2	9		113
231^85 -2	9		201
231^113 -2	9		268
231^199 -2	9		471
231^261 -2	9		617

$n_u = 231$	EZ:	SF_p	Dezimalstellen:
231^1 +2	3	233	3
231^3 +2	3	12326393	8
231^5 +2	3		12
231^6 +2	3		15
231^12 +2	3		29
231^20 +2	3		48
231^27 +2	3		64
231^31 +2	3		74
231^47 +2	3		113

Anmerkungen:
Im Exponenten-Intervall n = 1 bis 300 treten 12 Primzahlen SM_p und 9 Primzahlen SF_p auf, von denen 4 Zwillinge mod 4 bilden, die allesamt Elemente der *Folge 3* sind.

Die Schlussfolgerungen aus **Tab. 76**:

1. In den "Exponenten-Folgen" beliebiger ungerader Zahlen: n_u ^n treten stets Primzahlen der Typen SM_p und SF_p auf.

2. Fallweise bilden diese Primzahlen SM_p und SF_p Zwillinge mod 4 - diese wurden in **Tab. 76** durch Unterstreichung markiert.

Aus dem Studium der Anlagen: **Anlage 4** bis **Anlage 12**, das dem Leser empfohlen wird, ergeben sich weitere wichtige Erkenntnisse, die für die Bestimmung von Zwillingen mod 2n, n ≥1 wichtig sind, wie auch für die Bestimmung von Primzahlen mit definierten Endziffern: EZ1, EZ3, EZ7, EZ9-

In **Anlage 4** wurden die Primzahlen SM_p und SF_p, die fallweise Zwillinge mod 4 bilden, mit dem MuPAD - Test- Programm für definierte Zahlen-Intervalle generiert und ausgewertet.
Aus **Anlage 4** ergeben sich wichtige Erkenntnisse und Bestätigungen der bereits erkannten Gesetze, die nachfolgend zitiert werden:

Erkenntnis 1: Die Primzahlen SM_p und SF_p treten in den drei Folgen: *Folge 1*, *Folge 2*, *Folge 3* in unendlicher Zahl auf und bilden unendlich viele Zwillinge mod 4.

Erkenntnis 2: Alle Zwillinge mod 4 werden aus Primzahlen (SM_p, SF_p) gebildet, die stets Elemente der der Obergruppe OG I, Rest 1 oder OG II, Rest 3 sind.

Erkenntnis 3: In wachsenden Zahlen-Bereichen ($n \to \infty$) erfüllen die Primzahlen SM_p und SF_p sowie die aus ihnen gebildeten Zwillinge mod 4 das Asymptote-Gesetz.

Erkenntnis 4: Die Primzahlen SM_p und SF_p treten in sechs Ordnungs-Folgen auf und die aus ihnen gebildeten Zwillinge mod 4 in drei weiteren Ordnungs-Folgen.

Erkenntnis 5: Die neun Ordnungs-Folgen, in denen die Primzahlen SM_p, SF_p und die aus ihnen gebildeten Zwillinge mod 4 auftreten, können die Basis für eine sehr effektive Daten-Verschlüsselung mit einer breiten Anwendung bilden, da diese (neun) Ordnungs-Folgen (ähnlich wie die Dezimal-Zahlen: n:= 1 bis 100, n = 101 bis 200 usw.) in Abschnitten "genutzt" werden können.

In **Anlage 5** wurden die Primzahlen CM_p und SF_p, die fallweise Zwillinge mod 2 bilden, mit dem MuPAD - Test - Programm für definierte Zahlen-Intervalle generiert und ausgewertet.
Aus Anlage 5 ergeben sich (analoge) wichtige Erkenntnisse, die bereits für die Primzahlen SM_p, SF_p und die aus diesen Primzahlen gebildeten Zwillinge mod 4 aufgezählt wurden.

In **Anlage 6** wurden die Primzahlen CM_p und CF_p und die aus ihnen gebildeten Zwillinge mod 6 mit dem MuPAD - Test - Programm in verschiedenen Zahlen-Intervallen generiert und in analogen Ordnungs-Folgen präsentiert.

In **Anlage 7** wurden die Primzahlen SM_p und SF_p und die aus ihnen fallweise kreierten Zwillinge mod 8 mit dem MuPAD - Test - Programm für verschiedene Zahlen-Intervalle generiert.

In **Anlage 8** wurden die in den vier Folgen: *Folge 1*, *Folge 2*, *Folge 3* und *Folge 4* auftretenden Primzahlen CM_p und CF_p und die aus ihnen fallweise kreierten Zwillinge mod 10 mit dem MuPAD - Test - Programm für verschiedene Zahlen-Bereiche generiert.

In **Anlage 9** wurden die in den vier Folgen: *Folge 1*, *Folge 2*, *Folge 3* und *Folge 4* auftretenden Primzahlen CM_p und CF_p und die aus ihnen fallweise kreierten Zwillinge mod 20 mit dem MuPAD - Test - Programm für verschiedene Zahlen-Bereiche generiert.

In **Anlage 10** wurden die Zwillinge mod 2 und mod 6 bestimmt, die in definierten Intervallen der Potenz-Folgen: 6^n, 14^n und 18^n situiert sind. Bei der Bestimmung dieser Zwillinge wurden die erweiterten Beweis-Gleichungen des Hauptsatzes der Primzahl-Theorie nutzbar angewendet.

In **Anlage 11** wurden zahlreiche Primzahl-Folgen in definierten Intervallen in den Umgebungen diverser Zahlen (n_g) und (n_u) und zahlreiche Zwillings-Folgen diverser Modularität mit dem MuPAD-Testprogramm abgeleitet und die hier auftretenden Gesetzmäßigkeiten kommentiert.

In **Anlage 12** wurden drei Methoden für die Bestimmung von Primzahlen mit definierten Endziffern: EZ1, EZ3, EZ7, EZ9 und für die Bestimmung von Zwillingen mod 2n, n ≥1 in den Umgebungen der geraden Zahlen $n_g = 2\text{^}n$ mit dem MuPAD - Testprogramm vorgestellt. und kommentiert:

Methode 1: - die "vertikale Methode",
Methode 2: - die "horizontale Methode"
Methode 3: - die "horizontale Methode mit definierten Endziffern der Primzahlen"

Diese Methoden wurden in der **Anlage 12** eingehend kommentiert und ihre Vorteile an zahlreichen Beispielen herausgestellt. Es konnte dabei der Nachweis erbracht werden, dass sich für die Bestimmung sehr großer Primzahlen die Methode 3 am besten eignet. Beim Einsatz der Methode 3 genügt der Einsatz von einigen Hundert Rechnern, um die größte (noch unbekannte) Primzahl und dies noch mit einer vorgegebenen Endziffer zu bestimmen.

In der **Anlage 13** wurden die tieferen (noch unbekannten) Ursachen für das Asymptote-Gesetz und für die Schwankungen der Primzahl-Dichten als eine zwingende Folge von Gesetz 1 und Gesetz 2 der Faktor-Zerlegung von Zahlen $n \in N$ herausgestellt.

19. Die Suche nach optimalen Methoden für die Generierung sehr großer Primzahlen mit vorgegebenen Endziffern:

In **Anlage 12** wurden drei Methoden für die Bestimmung von Primzahlen mit den (vorgegebenen) Endziffern: EZ1, EZ3, EZ7, EZ9 und von Zwillingen mod 2n, n ≥1 eingeführt und kommentiert. Die in Anlage 12 analysierte Methode 3 (die "horizontale Methode mit definierten Endziffern der Primzahlen) hat sich dabei im Vergleich zu der Methode 1 (der "vertikalen Methode" der Suche nach besonders großen Primzahlen) als strategisch optimalere Methode erwiesen.

Hieraus ergibt sich die Schlussfolgerung, dass die dem GIMPS-Projekt zugrunde liegende Rechen - Methode auf die "horizontale Methode mit definierten Endziffern der Primzahlen" angepasst werden sollte. Alle näheren Informationen, die diesen Sachverhalt belegen, können der **Anlage 12** entnommen werden. In der **Anlage 12** wurde auch die 2918-stellige Primzahl p = 2^9691+6461 mit der vorgegebenen Endziffer EZ9 mit einem (entsprechend programmierten) MuPAD - Test gefunden.

In den Anlagen: **Anlage 4**, **Anlage 5**, **Anlage 6**, **Anlage 7**, **Anlage 8**, **Anlage 9**, **Anlage 10** und **Anlage 11** wurden des weiteren die Grundlagen der zahlreichen Ordnungs-Folgen vorgestellt, in denen die Zwillinge mod 2n, n ≥1 in den "Weiten" der natürlichen Zahlen $(n \to \infty)$ auftreten.

Diese Ordnungs-Folgen können in Abschnitte von Nummern-Folgen gegliedert werden und damit für eine (bisher unbekannte und sichere) Daten-Verschlüsselung genutzt werden.

20. Zusammenfassung und Ausblick:

In der vorgestellten Monographie wurden zahlreiche bisher in der Primzahl-Theorie noch unbekannte Ordnungsstrukturen und Gesetzmäßigkeiten unter Beweis gestellt, die in der unendlichen Folge der Primzahlen **$p := 2, 3, 5, 7, 11, \ldots, p_n, \ldots, (p \to p_\infty)$** immanent gelten.

Hierzu zählen:

1. Die Gliederung aller Primzahlen in zwei Obergruppen OG I und OG II, in denen gleich viele Primzahlen mit den Endziffern: EZ1, EZ3, EZ7 und EZ9, Rest 1 und Rest 3 auftreten, und 8 unendliche Dirichlet - Progressionen bilden.

2. Die Erweiterung der von Euklid stammenden Beweis-Gleichung des Hauptsatzes der Primzahl-Theorie zu vier Beweis-Gleichungen - siehe die nachfolgende **Tab. 1**

Tab. 1 Die vier Beweis-Gleichungen des Hauptsatzes der Primzahl-Theorie

<u>Gleichung 1:</u> $p = n_g +1, +3, +5, \ldots$ <u>Gleichung 2:</u> $p = n_g - 1, -3, -5, \ldots$

<u>Gleichung 3:</u> $p = n_u + 2, +4, +6, \ldots$ <u>Gleichung 4:</u> $p = n_u - 2, -4, -6, \ldots$

3. Die Gliederung der Zwillinge mod 2n, $n \geq 1$ in charakteristischen und invarianten unendlichen Folgen: *Folge 1, Folge 2, Folge 3* und fallweise in *Folge* 4 - siehe die nachfolgende **Tab. 2**

Tab. 2 Die Folge-Gleichungen der Zwillinge mod 2n, $n \geq 1$

Zwillinge mod:=	*Folge 1*	*Folge 2*	*Folge 3*	*Folge 4*
Zwillinge mod 2:	12 + 30n	18 + 30n	30 + 30n	
Zwillinge mod 4	5 + 10n	9 + 10n	11 + 10n	
Zwillinge mod 6	10 + 10n	14 + 10n	16 +10n	
Zwillinge mod 8	7 + 10n	13 + 10n	15 + 10n	
Zwillinge mod 10	6 + 10n	8 + 10n	12 + 10n	14 + 10n
-------------------	-----------	------------	-----------	------------
Zwillinge mod 100	51 + 10n	53 + 10n	57 + 10n	59 + 10n

Nachdem die in **Tab. 2** angeführten *Folgen* Dirichlet - Progressionen bilden, liegt der Beweis vor, dass alle Zwillinge mod 2n, $n \geq 1$ in einer unendlichen Zahl auftreten und dabei Ordnungsfolgen bilden, die für eine hochgradige Daten-Verschlüsselung nutzbar sind.

4. Es wurde der Nachweis erbracht, dass die geraden Zahlen $n_g = 2^n$ in vier unendlichen Folgen gemäß Tab. 3 auftreten, in denen die Endziffern sich zyklisch wiederholen.

Tab. 3 Die Folgen in denen die Zahlen $n_g = 2^n$ $(n \to \infty)$ auftreten:

Folge 1	*Folge 2*	*Folge 3*	*Folge 4*
2^{1+4n}, EZ2	2^{2+4n}, EZ4	2^{3+4n}, EZ8	2^{4+4n}, EZ6

5. An zahlreichen Beispielen wurden die Gesetze der Faktor-Zerlegungen der Zahlen vom Typ:

$$n_u = (n_g = \Pi p_i(p_i = 2 \text{ bis } p_n)) \pm n_u \quad \text{und} \quad n_u = (n_u = \Pi p_i,\ p_i > 2) \pm n_g$$

analysiert (von Gesetz 1 und Gesetz 2) und die Kriterien für die Bildung von Primzahlen abgeleitet, die in den Umgebungen der Zahlen n_g bzw. n_u (fallweise) auftreten können.

6. Für die Zahlen $n_g = 2^n \pm n_u$ (für konstante Werte der ungeraden Zahl n_u) wird bekanntlich bisher nur das "vertikale Testverfahren" für die Bestimmung von Primzahlen mit definierter Endziffer bei der Suche nach Mersenne'schen Primzahlen eingesetzt Diese Methode findet im GIMPs - Projekt Anwendung, die dafür Millionen von PC-Rechnern weltweit beschäftigt und auf dem Restmodul - Kalkül - mod 0 basiert.

Das in dieser Monographie eingeführte und erprobte "horizontale Testverfahren" für die Bestimmung großer Primzahlen mit definierten Endziffern im Intervall: $2^n \pm i$, benötigt jedoch nur einige Hundert PC-Rechner, denen definierte "Abstands-Zahlen: $\pm n_u$ aus dem Intervall $i := 0$ bis m vorgegeben werden.
Die "horizontale Testmethode" erlaubt es daher sehr große Primzahlen beim Einsatz parallel arbeitender Rechner zu bestimmen. Diese Testmethode erweist sich gegenüber der GIMPs - Methode daher als ein wesentlich schnelleres Verfahren.

Ein wichtiges Ergebnis, das im Rahmen dieser Monographie abgeleitet werden konnte, betrifft die Gesetzmäßigkeit, von der die Schwankungen der Primzahl-Dichten in großen Zahlen- Intervallen abhängig sind. Die hier gewonnenen Erkenntnisse geben Anlass zu der Vermutung, dass die von Gauß und Riemann eingeführten Gleichungen:

$$\pi(n) \simeq n / \ln(n) \simeq \mathrm{li}\,(n) \qquad \text{... die von Gauß stammenden Gleichungen}$$

$$R(n) = 1 + \sum_{k=1}^{k=\infty}\left(1/ k\, \zeta(k+1) \cdot (\ln(n)^k) / k! \right) \qquad \text{... die von Riemann stammenden Gleichungen}$$

$$\zeta(z) = 1 + 1/2^z + 1/3^z + 1/4^z + \ldots$$

die Zahl der Primzahlen $\pi(n)$ nur annähernd bestimmen. Aus diesen Gleichungen können auch nicht die in großen Intervallen: Δn auftretenden Zuwächse $\Delta\pi(\Delta n)$ der Primzahlen bestimmt werden

Aus den hier angestellten Untersuchungen geht hervor, dass es sehr große Zahlen-Intervalle Δn gibt, in denen gar keine Primzahl-Zuwächse erfolgen, in denen also die Beziehung: $\Delta\pi(\Delta n) = 0$ gilt.
Sofern man dann die Gauß-Gleichung in der Form: $\pi(n + \Delta n) = \pi(n) \simeq (n + \Delta n) / \ln (n + \Delta n)$ notiert, gelangt man zu höheren Werte für $\pi(n + \Delta n)$, die sich aus dem Quotienten: $(n + \Delta n) / \ln (n + \Delta n)$ ergeben.

Wenn man z.B. von der Fakultät $\Pi p_i = 2 \cdot 3 \cdot \ldots \cdot (p_n = M_{p,42})$ ausgeht, in der alle Primzahlen bis zu der größten bekannten Mersenne-Primzahl $M_{p,42}$, auftreten, die 7.816.230 Dezimalstellen hat, gelangt man zu einem primzahlfreien Intervall von rd. $\Delta n \geq 20.000.000$. Aus der Gleichung von Gauß ergibt sich dann ein eindeutig zu hoher Wert für die Zahl der Primzahlen im Bereich: $\pi(n + \Delta n)$.

Seitens des Verfassers besteht deshalb auch die Vermutung, dass weitergehende Untersuchungen die Riemann'sche Vermutung ebenfalls nur als eine gute Annäherung beweisen können, die annähernd die variierenden Primzahl-Dichten in den Umgebungen der Zahlen $n = 2^{\alpha,2} \cdot 3^{\alpha,3} \cdot 5^{\alpha,5} \ldots p^{\alpha,p}$ erfasst, jedoch nicht exakt bestimmen kann, da die Summenzahl $\pi(n)$ eine asymptotisch fallende Treppen-Funktion darstellt, die auch von der Riemann'schen Zeta-Funktion nicht nachgebildet werden kann.

Nachdem (inzwischen) sowohl die Goldbach'sche Vermutung, als auch die Zwillings-Vermutung unter Beweis gestellt werden konnten, wie auch die Vermutung, dass es unendlich viele Mersenne- und Fermat - Primzahlen gibt, konnte über den Wahrheitswert der Riemann'schen Vermutung hingegen noch keine abschließende Erkenntnis gewonnen werden.

Aus den hier vorgestellten Untersuchungen resultiert die Empfehlung das GIMPs-Projekt derart zu erweitern, dass auf der Basis der "Restklassen - modulo Null Methode" auch sehr große Primzahlen mit den Endziffern: EZ3 und EZ9 im Umfeld extrem hoher gerader Zahlen n_g = 2^n ($n \to \infty$) bestimmt werden können. Des weiteren sollte diese Restklassen - Methode auch auf sehr große ungerade Zahlen p^n ($n \to \infty$), p:= 3, 5, 7, ... p_n erweitert werden.

Aus der Erweiterung des in Kapitel 1 zitierten Hauptsatzes der Primzahl-Theorie resultieren nämlich unendlich viele extrem große Primzahlen vom Typ:

$$p_{n\to\infty} = p^n\,(n \to \infty) + 2 \quad \text{und} \quad p_{n\to\infty} = p^n\,(n \to \infty) - 2$$

Aus dem im Rahmen dieser Monographie erbrachten Beweis, dass alle Zwillinge mod 2n, n $\geq$2 in einer unendlichen Zahl innerhalb eindeutig definierter Dirichlet-Progressionen auftreten, ergeben sich des weiteren wichtige (neue) Aspekte für eine gesicherte Daten-Verschlüsselung unter Einsatz der sehr empfehlenswerten MuPAD - Testmethode.

Auf diese Thematik konnte im Rahmen dieser Monographie jedoch nicht erschöpfend eingegangen werden. Unternehmen und Teams, die an hochgradig gesicherter Daten-Verschlüsselung besonders interessiert sind, können deshalb mit dem Verfasser einen direkten Kontakt aufnehmen.

Bochum, im April 2006
Der Verfasser: Dr. Georg Viktor Cwienk

21. Anlagen - Verzeichnis

1. Anlage 1
Die Ordnungsfolge der ungeraden Primzahlen von $p = 3$ bis $p = 99.991$ und die Folge der ersten 1224 (markiert hervorgehobenen) Zwillinge mod 2 - 19 Seiten

2. Anlage 2
Die geordneten Folgen der Zwillinge mod 2, mod 4, mod 6, mod 8, mod 10 und mod 100, die im Zahlen-Intervall: 0 bis 10.000 auftreten - 13 Seiten

3. Anlage 3
Die geordneten Folgen der Zwillinge mod 2, mod 4, mod 6, mod 8, mod 10 und mod 100, die im (verschobenen) Zahlen-Intervall: 1.000.000 bis 1.010.000 auftreten - 7 Seiten

4. Anlage 4
Die Bestimmung der Primzahlen in der Umgebung der Zahl $n_g = 2 \cdot 3 \cdot 5 \cdot 7 \cdot ... \cdot 251$ im Zahlen-Intervall: $n_g - 3000$ bis $n_g + 3000$ mit dem MuPAD - Testverfahren - 12 Seiten

5. Anlage 5
Die Generierung von Primzahlen vom Typ CM_p und CF_p mit dem MuPAD - Testprogramm und die Bestimmung von Zwillingen mod 2 in *Folge 1, Folge 2 , Folge 3* - 21 Seiten

6. Anlage 6
Die Generierung von Primzahlen vom Typ CM_p, CF_p und der aus ihnen kreierten Zwillinge mod 6 mit dem MuPAD - Testprogramm - 11 Seiten

7. Anlage 7
Die Bestimmung der Primzahlen SM_p und SF_p und der Zwillinge mod 8 in den Intervallen i:= 0 bis 30 und i:= 10^350 bis 10^350+2000 - 10 Seiten

8. Anlage 8
Die Bestimmung der Primzahlen CM_p und CF_p sowie der Zwillinge mod 10, die in unterschiedlichen Intervallen i:= 0 bis 30 und i:= 10^150 bis 10^150 +2000 situiert sind mit dem MuPAD - Testprogramm - 13 Seiten

9. Anlage 9
Die Generierung von Primzahlen vom Typ SM_p, SF_p und von Zwillingen mod 20, die in den vier Folgen: *Folge 1, Folge 2, Folge 3, Folge 4* auftreten mit dem MuPAD - Test - 14 Seiten

10. Anlage 10
Die Bestimmung der Zwillinge mod 2 und mod 6 in den Potenz - Folgen: 6^n, 14^n und 18^n mit dem MuPAD - Testprogramm - 7 Seiten

11. Anlage 11
Die erweiterten Beweisgleichungen des Hauptsatzes der Primzahl-Theorie führen zu drei Methoden für die Bestimmung von Primzahlen und Zwillingen mod 2n, $n \geq 1$ - 20 Seiten

12. Anlage 12 - Die Bestimmung von Primzahlen mit definierten Endziffern: EZ1, ET3, EZ7, EZ9 und von Zwillingen mod 2n, $n \geq 1$ in den Umgebungen der geraden Zahlen $n_g = 2^n$ mit dem MuPAD - Testprogramm - 31 Seiten

13. Anlage 13 - Die Ursachen für das Asymptote-Gesetz und die Schwankungen der Primzahl - Dichten, die durch <u>Gesetz 1</u> und <u>Gesetz 2</u> der Faktor-Zerlegung von Zahlen $n \in N$ bedingt sind - 7 Seiten

22. Verzeichnis der verwendeten PC - Testprogramme

1. PC-Test zur Bestimmung der Zwillinge mod 2n, $n \geq 1$ in wählbaren Zahlen-Intervallen, *)

2. PC-Test zur Bestimmung der geordneten Folgen von Primzahlen der Klassen 1 bis 8, *) in wählbaren Zahlen-Intervallen

3. PC- Primzahl-Testung für Zahlen bis zu 16 Dezimalstellen

4. PC - Primzahl - Tester für Zahlen bis zu 29 Dezimalstellen

5. PC - MuPAD, Pro - Programm für die Faktor - Zerlegung von Zahlen

6. PC -MuPAD, Pro - Programm für die Bestimmung von Primzahlen

*) Diese PC- Programme können beim Verlag bestellt werden

23. Literaturhinweise

/1 / Richard Courant, Herbert Robinson — *Was ist Mathematik - Zweite Auflage (Springer Verlag Berlin, Heidelberg, New York - 1967)*

/2 / Heinrich Tietze — *Gelöste und ungelöste mathematische Probleme aus alter und neuer Zeit - Band 1 (d.t.v wissenschaft, 1982)*

/3 / P.S. Alexandrow, A. Markuschewitsch, A.J. Chintschin — *Enzyklopädie der Elementarmathematik - Band 1 (VEB-Verlag, Berlin 1972)*

/4 / Behnke, Bachmann, Fladt, Süss — *Grundzüge der Mathematik (Göttingen, Vandenhoek, Ruprecht, 1966)*

/5 / E. *Grosswald* — *Topics from the Theory of Numbers (New York, Mac-Millan, 1966)*

/6 / D.N. Lehmer — *List of Prime Numbers from 1 to 10.006.721 (Washington, Publication No. 163, 1914)*

/ 7 / H. Zassenhaus — *Über die Existenz von Primzahlen in arithmetischen Progressionen (Comment Helvetici 22, 1949)*

/8 / K. Prachar — *Primzahlverteilungen, Grundlehren 91 (Berlin, 1957, Springer Verlag)*

/9 / D. Zagier — *Die ersten 50 Millionen Primzahlen (Elemente der Mathematik, Beiheft Nr. 15, 1977)*

/10 / L.A. Steen — *Order from Chaos (Science News, 1975)*

/11 / Ph. J. Davis, R. Hersch — *Erfahrung Mathematik (Birkhäuser Verlag Basel, Boston, Berlin - 1994)*

/12 / Fr. Padberg — *Elementare Zahlentheorie (Herder Verlag Freiburg, 1972)*

/13 / E. Zeidler (Herausgeber) — *Teubner - Taschenbuch der Mathematik (B. G. Teubner Verlag, stuttgart. leipzig, 1996)*

/14 / Marcus du Sautoy — *Musiik der Primzahlen (C. H. Beck-Verlag, 2004)*

/15 / Hans Riesel — *Prime numbers and computer methods for factorization (Birkhhäuser, Boston, 1994)*

/16 / G. V. Cwienk — *Der Beweis der Goldbach'schen Vermutung - Die verborgenen Eigenschaften von Primzahlzwillingen mod 2n, n ≥1*

Die erweiterten Dirichlet - Progressionen:

OG I:

Klasse 1, EZ1, Rest 1:= 41 + 20n ... 41, 61, 101, ...
Klasse 2, EZ3, Rest 1:= 13 + 20n ... 13, 53, 73, ...
Klasse 3, EZ7, Rest 1:= 17 + 20n ... 17, 37, 97, ...
Klasse 4, EZ9, Rest 1:= 29 + 20n ... 29, 89, 109, ...

OG II:

Klasse 5, EZ1, Rest 3:= 11 + 20n ... 11, 31, 71, ...
Klasse 6, EZ3, Rest 3:= 3 + 20n ... 3, 23, 43, ...
Klasse 7, EZ7, Rest 3:= 7 + 20n ... 7, 47, 67, ...
Klasse 8, EZ9, Rest 3:= 19 + 20n ... 19, 59, 79, ...

Die Bildungs-Gesetze der *Folgen* in denen Zwillinge mod 2n, n ≥1 auftreten

Zwillinge mod 2n, n ≥1:	*Folge 1*	*Folge 2*	*Folge 3*	*Folge 4*
Zwillinge mod 2:	12 + 30n	18 + 30n	30 + 30n	---------
Zwillinge mod 4:	5 + 10n	9 + 10n	11 + 10n	.---------
Zwillinge mod 6:	14 + 10n	16 + 10n	10 + 10n	---------
Zwillinge mod 8:	7 + 10n	13 + 10n	15 + 10n	---------
Zwillinge mod 10:	6 + 10n	8 + 10n	12 + 10n	14 + 10n
Zwillinge mod 12:	7 + 10n	13 + 10n	15 + 10n	---------
Zwillinge mod 14:	10 + 10n	14 + 10n	16 + 10n	---------
Zwillinge mod 16:	9 + 10n	11 + 10n	15 + 10n	---------
Zwillinge mod 18	10 + 10n	12 + 10n	18 + 10n	----------
Zwillinge mod 20:	11 + 10n	13 + 10n	17 + 10n	19 + 10n
Zwillinge mod 100:	51 + 10n	53 + 10n	57 + 10n	59 + 10n

Die erweiterten Gleichungen für den Beweis des Hauptsatzes der Primzahl-Theorie:

1. Gleichung: $p = n_g + 1, + 3, +5, \ldots, + (n_u = p_{n+1} \cdot p_{n+2} \cdot \ldots \cdot p_{n+n})$
2. Gleichung: $p = n_g - 1, - 3, -5, \ldots, - (n_u = p_{n+1} \cdot p_{n+2} \cdot \ldots \cdot p_{n+n})$
3. Gleichung: $p = n_u + 2, +4, +6, \ldots, + (n_g = 2 \cdot p_{n+1} \cdot p_{n\,2} \cdot \ldots \cdot p_{n+n})$
4. Gleichung: $p = n_u - 2, - 4, - 6,, \ldots, - (n_g = 2 \cdot p_{n+1} \cdot p_{n+2} \cdot \ldots \cdot p_{n+n})$

Primzahlen die in den addierten bzw. subtrahierten geraden bzw. ungeraden Zahlen als Faktoren auftreten, dürfen in den vorgegebenen geraden bzw. ungeraden Zahlen nicht auftreten. Unter dieser Prämisse ergeben sich fallweise aus den vier Gleichungen prime Zahlen oder aber Zahlen, in denen Primzahl-Faktoren mit größeren Primzahlen als die Primzahlen, die in den Zahlen: n_g bzw. n_u auftreten.

Anlage 1

Primzahlen

3	5	7	11	13	17	19	23
29	31	37	41	43	47	53	59
61	67	71	73	79	83	89	97
101	103	107	109	113	127	131	137
139	149	151	157	163	167	173	179
181	191	193	197	199	211	223	227
229	233	239	241	251	257	263	269
271	277	281	283	293	307	311	313
317	331	337	347	349	353	359	367
373	379	383	389	397	401	409	419
421	431	433	439	443	449	457	461
463	467	479	487	491	499	503	509
521	523	541	547	557	563	569	571
577	587	593	599	601	607	613	617
619	631	641	643	647	653	659	661
673	677	683	691	701	709	719	727
733	739	743	751	757	761	769	773
787	797	809	811	821	823	827	829
839	853	857	859	863	877	881	883
887	907	911	919	929	937	941	947
953	967	971	977	983	991	997	1009
1013	1019	1021	1031	1033	1039	1049	1051
1061	1063	1069	1087	1091	1093	1097	1103
1109	1117	1123	1129	1151	1153	1163	1171
1181	1187	1193	1201	1213	1217	1223	1229
1231	1237	1249	1259	1277	1279	1283	1289
1291	1297	1301	1303	1307	1319	1321	1327
1361	1367	1373	1381	1399	1409	1423	1427
1429	1433	1439	1447	1451	1453	1459	1471
1481	1483	1487	1489	1493	1499	1511	1523
1531	1543	1549	1553	1559	1567	1571	1579
1583	1597	1601	1607	1609	1613	1619	1621
1627	1637	1657	1663	1667	1669	1693	1697
1699	1709	1721	1723	1733	1741	1747	1753
1759	1777	1783	1787	1789	1801	1811	1823
1831	1847	1861	1867	1871	1873	1877	1879
1889	1901	1907	1913	1931	1933	1949	1951
1973	1979	1987	1993	1997	1999	2003	2011
2017	2027	2029	2039	2053	2063	2069	2081
2083	2087	2089	2099	2111	2113	2129	2131
2137	2141	2143	2153	2161	2179	2203	2207
2213	2221	2237	2239	2243	2251	2267	2269
2273	2281	2287	2293	2297	2309	2311	2333
2339	2341	2347	2351	2357	2371	2377	2381
2383	2389	2393	2399	2411	2417	2423	2437
2441	2447	2459	2467	2473	2477	2503	2521
2531	2539	2543	2549	2551	2557	2579	2591
2593	2609	2617	2621	2633	2647	2657	2659
2663	2671	2677	2683	2687	2689	2693	2699
2707	2711	2713	2719	2729	2731	2741	2749
2753	2767	2777	2789	2791	2797	2801	2803
2819	2833	2837	2843	2851	2857	2861	2879
2887	2897	2903	2909	2917	2927	2939	2953
2957	2963	2969	2971	2999	3001	3011	3019
3023	3037	3041	3049	3061	3067	3079	3083
3089	3109	3119	3121	3137	3163	3167	3169
3181	3187	3191	3203	3209	3217	3221	3229
3251	3253	3257	3259	3271	3299	3301	3307
3313	3319	3323	3329	3331	3343	3347	3359
3361	3371	3373	3389	3391	3407	3413	3433
3449	3457	3461	3463	3467	3469	3491	3499
3511	3517	3527	3529	3533	3539	3541	3547

Anlage 1

.....**3557**.....3559.....3571.....3581.....**3583**.....3593.....3607.....3613
.....3617.....3623.....3631.....3637.....3643.....3659.....**3671**.....**3673**
.....3677.....3691.....3697.....3701.....3709.....3719.....3727.....3733
.....3739.....3761.....**3767**.....**3769**.....3779.....3793.....3797.....3803
.....**3821**.....**3823**.....3833.....3847.....**3851**.....**3853**.....3863.....3877
.....3881.....3889.....3907.....3911.....**3917**.....**3919**.....3923.....**3929**
.....**3931**.....3943.....3947.....3967.....3989.....**4001**.....**4003**.....4007
.....4013.....**4019**.....**4021**.....4027.....**4049**.....**4051**.....4057.....4073
.....4079.....**4091**.....**4093**.....4099.....4111.....4127.....4129.....4133
.....4139.....4153.....**4157**.....**4159**.....4177.....4201.....4211.....**4217**
.....**4219**.....**4229**.....**4231**.....**4241**.....**4243**.....4253.....**4259**.....**4261**
.....**4271**.....**4273**.....4283.....4289.....4297.....4327.....**4337**.....**4339**
.....4349.....4357.....4363.....4373.....4391.....4397.....4409.....4421
.....4423.....4441.....4447.....4451.....4457.....4463.....**4481**.....**4483**
.....4493.....4507.....4513.....**4517**.....**4519**.....4523.....**4547**.....**4549**
.....4561.....4567.....4583.....4591.....4597.....4603.....4621.....**4637**
.....**4639**.....4643.....**4649**.....**4651**.....4657.....4663.....4673.....4679
.....4691.....4703.....**4721**.....**4723**.....4729.....4733.....4751.....4759
.....4783.....**4787**.....**4789**.....4793.....**4799**.....**4801**.....4813.....4817
.....4831.....4861.....4871.....4877.....4889.....4903.....4909.....4919
.....**4931**.....**4933**.....4937.....4943.....4951.....4957.....**4967**.....**4969**
.....4973.....4987.....4993.....4999.....5003.....**5009**.....**5011**.....**5021**
.....**5023**.....5039.....5051.....5059.....5077.....5081.....5087.....**5099**
.....**5101**.....5107.....5113.....5119.....5147.....5153.....5167.....5171
.....5179.....5189.....5197.....5209.....5227.....**5231**.....**5233**.....5237
.....5261.....5273.....**5279**.....**5281**.....5297.....5303.....5309.....5323
.....5333.....5347.....5351.....5381.....5387.....5393.....5399.....5407
.....5413.....**5417**.....**5419**.....5431.....5437.....**5441**.....**5443**.....5449
.....5471.....**5477**.....**5479**.....5483.....**5501**.....**5503**.....5507.....**5519**
.....**5521**.....5527.....5531.....5557.....5563.....5569.....5573.....5581
.....5591.....5623.....**5639**.....**5641**.....5647.....**5651**.....**5653**.....**5657**
.....**5659**.....5669.....5683.....5689.....5693.....5701.....5711.....5717
.....5737.....**5741**.....**5743**.....5749.....5779.....5783.....5791.....5801
.....5807.....5813.....5821.....5827.....5839.....5843.....**5849**.....**5851**
.....5857.....5861.....**5867**.....**5869**.....**5879**.....**5881**.....5897.....5903
.....5923.....5927.....5939.....5953.....5981.....5987.....6007.....6011
.....6029.....6037.....6043.....6047.....6053.....6067.....6073.....6079
.....**6089**.....**6091**.....6101.....6113.....6121.....**6131**.....**6133**.....6143
.....6151.....6163.....6173.....**6197**.....**6199**.....6203.....6211.....6217
.....6221.....6229.....6247.....6257.....6263.....**6269**.....**6271**.....6277
.....6287.....**6299**.....**6301**.....6311.....6317.....6323.....6329.....6337
.....6343.....6353.....**6359**.....**6361**.....6367.....6373.....6379.....6389
.....6397.....6421.....6427.....**6449**.....**6451**.....6469.....6473.....6481
.....6491.....6521.....6529.....6547.....**6551**.....**6553**.....6563.....6569
.....6571.....6577.....6581.....6599.....6607.....6619.....6637.....6653
.....**6659**.....**6661**.....6673.....6679.....**6689**.....**6691**.....**6701**.....**6703**
.....6709.....6719.....6733.....6737.....**6761**.....**6763**.....**6779**.....**6781**
.....**6791**.....**6793**.....6803.....6823.....**6827**.....**6829**.....6833.....6841
.....6857.....6863.....**6869**.....**6871**.....6883.....6899.....6907.....6911
.....6917.....**6947**.....**6949**.....**6959**.....**6961**.....6967.....6971.....6977
.....6983.....6991.....6997.....7001.....7013.....7019.....7027.....7039
.....7043.....7057.....7069.....7079.....7103.....7109.....7121.....**7127**
.....**7129**.....7151.....7159.....7177.....7187.....7193.....7207.....**7211**
.....**7213**.....7219.....7229.....7237.....7243.....7247.....7253.....7283
.....7297.....**7307**.....**7309**.....7321.....**7331**.....**7333**.....**7349**.....**7351**
.....7369.....7393.....7411.....7417.....7433.....7451.....**7457**.....**7459**
.....7477.....7481.....**7487**.....**7489**.....7499.....7507.....7517.....7523
.....7529.....7537.....7541.....**7547**.....**7549**.....**7559**.....**7561**.....7573
.....7577.....7583.....**7589**.....**7591**.....7603.....7607.....7621.....7639
.....7643.....7649.....7669.....7673.....7681.....7687.....7691.....7699
.....7703.....7717.....7723.....7727.....7741.....7753.....**7757**.....**7759**
.....7789.....7793.....7817.....7823.....7829.....7841.....7853.....7867
.....7873.....**7877**.....**7879**.....7883.....7901.....7907.....7919.....7927
.....7933.....7937.....**7949**.....**7951**.....7963.....7993.....**8009**.....**8011**

Anlage 1

.....8017.....8039.....8053.....8059.....8069.....8081.....8087.....8089
.....8093.....8101.....8111.....8117.....8123.....8147.....8161.....8167
.....8171.....8179.....8191.....8209.....8219.....8221.....8231.....8233
.....8237.....8243.....8263.....8269.....8273.....8287.....8291.....8293
.....8297.....8311.....8317.....8329.....8353.....8363.....8369.....8377
.....8387.....8389.....8419.....8423.....8429.....8431.....8443.....8447
.....8461.....8467.....8501.....8513.....8521.....8527.....8537.....8539
.....8543.....8563.....8573.....8581.....8597.....8599.....8609.....8623
.....8627.....8629.....8641.....8647.....8663.....8669.....8677.....8681
.....8689.....8693.....8699.....8707.....8713.....8719.....8731.....8737
.....8741.....8747.....8753.....8761.....8779.....8783.....8803.....8807
.....8819.....8821.....8831.....8837.....8839.....8849.....8861.....8863
.....8867.....8887.....8893.....8923.....8929.....8933.....8941.....8951
.....8963.....8969.....8971.....8999.....9001.....9007.....9011.....9013
.....9029.....9041.....9043.....9049.....9059.....9067.....9091.....9103
.....9109.....9127.....9133.....9137.....9151.....9157.....9161.....9173
.....9181.....9187.....9199.....9203.....9209.....9221.....9227.....9239
.....9241.....9257.....9277.....9281.....9283.....9293.....9311.....9319
.....9323.....9337.....9341.....9343.....9349.....9371.....9377.....9391
.....9397.....9403.....9413.....9419.....9421.....9431.....9433.....9437
.....9439.....9461.....9463.....9467.....9473.....9479.....9491.....9497
.....9511.....9521.....9533.....9539.....9547.....9551.....9587.....9601
.....9613.....9619.....9623.....9629.....9631.....9643.....9649.....9661
.....9677.....9679.....9689.....9697.....9719.....9721.....9733.....9739
.....9743.....9749.....9767.....9769.....9781.....9787.....9791.....9803
.....9811.....9817.....9829.....9833.....9839.....9851.....9857.....9859
.....9871.....9883.....9887.....9901.....9907.....9923.....9929.....9931
.....9941.....9949.....9967.....9973....10007....10009....10037....10039
....10061....10067....10069....10079....10091....10093....10099....10103
....10111....10133....10139....10141....10151....10159....10163....10169
....10177....10181....10193....10211....10223....10243....10247....10253
....10259....10267....10271....10273....10289....10301....10303....10313
....10321....10331....10333....10337....10343....10357....10369....10391
....10399....10427....10429....10433....10453....10457....10459....10463
....10477....10487....10499....10501....10513....10529....10531....10559
....10567....10589....10597....10601....10607....10613....10627....10631
....10639....10651....10657....10663....10667....10687....10691....10709
....10711....10723....10729....10733....10739....10753....10771....10781
....10789....10799....10831....10837....10847....10853....10859....10861
....10867....10883....10889....10891....10903....10909....10937....10939
....10949....10957....10973....10979....10987....10993....11003....11027
....11047....11057....11059....11069....11071....11083....11087....11093
....11113....11117....11119....11131....11149....11159....11161....11171
....11173....11177....11197....11213....11239....11243....11251....11257
....11261....11273....11279....11287....11299....11311....11317....11321
....11329....11351....11353....11369....11383....11393....11399....11411
....11423....11437....11443....11447....11467....11471....11483....11489
....11491....11497....11503....11519....11527....11549....11551....11579
....11587....11593....11597....11617....11621....11633....11657....11677
....11681....11689....11699....11701....11717....11719....11731....11743
....11777....11779....11783....11789....11801....11807....11813....11821
....11827....11831....11833....11839....11863....11867....11887....11897
....11903....11909....11923....11927....11933....11939....11941....11953
....11959....11969....11971....11981....11987....12007....12011....12037
....12041....12043....12049....12071....12073....12097....12101....12107
....12109....12113....12119....12143....12149....12157....12161....12163
....12197....12203....12211....12227....12239....12241....12251....12253
....12263....12269....12277....12281....12289....12301....12323....12329
12343....12347....12373....12377....12379....12391....12401....12409
....12413....12421....12433....12437....12451....12457....12473....12479
....12487....12491....12497....12503....12511....12517....12527....12539
....12541....12547....12553....12569....12577....12583....12589....12601
....12611....12613....12619....12637....12641....12647....12653....12659
....12671....12689....12697....12703....12713....12721....12739....12743

Anlage 1

....12757....12763....12781....12791....12799....12809....**12821**....**12823**
....12829....12841....12853....12889....12893....12899....12907....12911
....**12917**....**12919**....12923....12941....12953....12959....12967....12973
....12979....12983....**13001**....**13003**....**13007**....**13009**....13033....13037
....13043....13049....13063....13093....13099....13103....13109....13121
....13127....13147....13151....13159....13163....13171....13177....13183
....13187....**13217**....**13219**....13229....13241....13249....13259....13267
....13291....13297....13309....13313....13327....13331....**13337**....**13339**
....13367....13381....**13397**....**13399**....13411....13417....13421....13441
....13451....13457....13463....13469....13477....13487....13499....13513
....13523....13537....13553....13567....13577....13591....13597....13613
....13619....13627....13633....13649....13669....**13679**....**13681**....13687
....**13691**....**13693**....13697....**13709**....**13711**....**13721**....**13723**....13729
....13751....**13757**....**13759**....13763....13781....13789....13799....13807
....**13829**....**13831**....13841....13859....13873....13877....13879....13883
....**13901**....**13903**....13907....13913....13921....**13931**....**13933**....13963
....13967....**13997**....**13999**....**14009**....**14011**....**14029**....**14033**....14051
....14057....14071....**14081**....**14083**....14087....14107....14143....14149
....14153....14159....14173....14177....14197....14207....14221....14243
....**14249**....**14251**....14281....14293....14303....**14321**....**14323**....14327
....14341....14347....14369....**14387**....**14389**....14401....14407....14411
....14419....14423....14431....14437....**14447**....**14449**....14461....14479
....14489....14503....14519....14533....14537....14543....14549....14551
....14557....**14561**....**14563**....**14591**....**14593**....14621....**14627**....**14629**
....14633....14639....14653....14657....14669....14683....14699....14713
....14717....14723....14731....14737....14741....14747....14753....14759
....14767....14771....14779....14783....14797....14813....14821....14827
....14831....14843....14851....**14867**....**14869**....14879....14887....14891
....14897....14923....14929....14939....14947....14951....14957....14969
....14983....15013....15017....15031....15053....15061....15073....15077
....15083....15091....15101....15107....15121....15131....**15137**....**15139**
....15149....15161....15173....15187....15193....15199....15217....15227
....15233....15241....15259....15263....**15269**....**15271**....15277....**15287**
....**15289**....15299....15307....15313....15319....**15329**....**15331**....15349
....**15359**....**15361**....15373....15377....15383....15391....15401....15413
....15427....15439....15443....15451....15461....15467....15473....15493
....15497....15511....15527....15541....15551....15559....15569....**15581**
....**15583**....15601....15607....15619....15629....**15641**....**15643**....**15647**
....**15649**....15661....15667....15671....15679....15683....15727....**15731**
....**15733**....**15737**....**15739**....15749....15761....15767....15773....15787
....15791....15797....15803....15809....15817....15823....15859....15877
....15881....**15887**....**15889**....15901....15907....15913....15919....15923
....15937....15959....**15971**....**15973**....15991....16001....16007....16033
....16057....**16061**....**16063**....**16067**....**16069**....16073....16087....16091
....16097....16103....16111....16127....**16139**....**16141**....16183....**16187**
....**16189**....16193....16217....16223....**16229**....**16231**....16249....16253
....16267....16273....16301....16319....16333....16339....16349....**16361**
....**16363**....16369....16381....16411....16417....16421....16427....16433
....16447....**16451**....**16453**....16477....16481....16487....16493....16519
....16529....16547....16553....16561....16567....16573....16603....16607
....16619....**16631**....**16633**....**16649**....**16651**....16657....16661....16673
....**16691**....**16693**....16699....16703....16729....16741....16747....16759
....16763....16787....16811....16823....**16829**....**16831**....16843....16871
....16879....16883....16889....**16901**....**16903**....16921....16927....16931
....16937....16943....16963....**16979**....**16981**....16987....16993....17011
....17021....**17027**....**17029**....17033....17041....17047....17053....17077
....17093....17099....17107....17117....17123....17137....17159....17167
....17183....**17189**....**17191**....17203....**17207**....**17209**....17231....17239
....17257....**17291**....**17293**....17299....17317....17321....17327....17333
....17341....17351....17359....17377....17383....**17387**....**17389**....17393
....17401....**17417**....**17419**....17431....17443....17449....17467....17471
....17477....17483....**17489**....**17491**....17497....17509....17519....17539
....17551....17569....17573....**17579**....**17581**....**17597**....**17599**....17609
....17623....17627....**17657**....**17659**....17669....**17681**....**17683**....17707

....17713....17729....17737....17747....17749....17761....17783....17789
....17791....17807....17827....17837....17839....17851....17863....17881
....17891....17903....17909....17911....17921....17923....17929....17939
....17957....17959....17971....17977....17981....17987....17989....18013
....18041....18043....18047....18049....18059....18061....18077....18089
....18097....18119....18121....18127....18131....18133....18143....18149
....18169....18181....18191....18199....18211....18217....18223....18229
....18233....18251....18253....18257....18269....18287....18289....18301
....18307....18311....18313....18329....18341....18353....18367....18371
....18379....18397....18401....18413....18427....18433....18439....18443
....18451....18457....18461....18481....18493....18503....18517....18521
....18523....18539....18541....18553....18583....18587....18593....18617
....18637....18661....18671....18679....18691....18701....18713....18719
....18731....18743....18749....18757....18773....18787....18793....18797
....18803....18839....18859....18869....18899....18911....18913....18917
....18919....18947....18959....18973....18979....19001....19009....19013
....19031....19037....19051....19069....19073....19079....19081....19087
....19121....19139....19141....19157....19163....19181....19183....19207
....19211....19213....19219....19231....19237....19249....19259....19267
....19273....19289....19301....19309....19319....19333....19373....19379
....19381....19387....19391....19403....19417....19421....19423....19427
....19429....19433....19441....19447....19457....19463....19469....19471
....19477....19483....19489....19501....19507....19531....19541....19543
....19553....19559....19571....19577....19583....19597....19603....19609
....19661....19681....19687....19697....19699....19709....19717....19727
....19739....19751....19753....19759....19763....19777....19793....19801
....19813....19819....19841....19843....19853....19861....19867....19889
....19891....19913....19919....19927....19937....19949....19961....19963
....19973....19979....19991....19993....19997....20011....20021....20023
....20029....20047....20051....20063....20071....20089....20101....20107
....20113....20117....20123....20129....20143....20147....20149....20161
....20173....20177....20183....20201....20219....20231....20233....20249
....20261....20269....20287....20297....20323....20327....20333....20341
....20347....20353....20357....20359....20369....20389....20393....20399
....20407....20411....20431....20441....20443....20477....20479....20483
....20507....20509....20521....20533....20543....20549....20551....20563
....20593....20599....20611....20627....20639....20641....20663....20681
....20693....20707....20717....20719....20731....20743....20747....20749
....20753....20759....20771....20773....20789....20807....20809....20849
....20857....20873....20879....20887....20897....20899....20903....20921
....20929....20939....20947....20959....20963....20981....20983....21001
....21011....21013....21017....21019....21023....21031....21059....21061
....21067....21089....21101....21107....21121....21139....21143....21149
....21157....21163....21169....21179....21187....21191....21193....21211
....21221....21227....21247....21269....21277....21283....21313....21317
....21319....21323....21341....21347....21377....21379....21383....21391
....21397....21401....21407....21419....21433....21467....21481....21487
....21491....21493....21499....21503....21517....21521....21523....21529
....21557....21559....21563....21569....21577....21587....21589....21599
....21601....21611....21613....21617....21647....21649....21661....21673
....21683....21701....21713....21727....21737....21739....21751....21757
....21767....21773....21787....21799....21803....21817....21821....21839
....21841....21851....21859....21863....21871....21881....21893....21911
....21929....21937....21943....21961....21977....21991....21997....22003
....22013....22027....22031....22037....22039....22051....22063....22067
....22073....22079....22091....22093....22109....22111....22123....22129
....22133....22147....22153....22157....22159....22171....22189....22193
....22229....22247....22259....22271....22273....22277....22279....22283
....22291....22303....22307....22343....22349....22367....22369....22381
....22391....22397....22409....22433....22441....22447....22453....22469
....22481....22483....22501....22511....22531....22541....22543....22549
....22567....22571....22573....22613....22619....22621....22637....22639
....22643....22651....22669....22679....22691....22697....22699....22709
....22717....22721....22727....22739....22741....22751....22769....22777

Anlage 1

....22783....22787....22807....22811....22817....22853....22859....22861
....22871....22877....22901....22907....22921....22937....22943....**22961**
....**22963**....22973....22993....23003....23011....23017....23021....**23027**
....**23029**....**23039**....**23041**....23053....**23057**....**23059**....23063....23071
....23081....23087....23099....23117....23131....23143....23159....23167
....23173....23189....23197....**23201**....**23203**....23209....23227....23251
....23269....23279....**23291**....**23293**....23297....23311....23321....23327
....23333....23339....23357....**23369**....**23371**....23399....23417....23431
....23447....23459....23473....23497....23509....23531....**23537**....**23539**
....23549....23557....**23561**....**23563**....23567....23581....23593....23599
....23603....23609....23623....**23627**....**23629**....23633....23663....**23669**
....**23671**....23677....**23687**....**23689**....23719....**23741**....**23743**....23747
....23753....23761....23767....23773....23789....23801....23813....23819
....23827....**23831**....**23833**....23857....23869....23873....23879....23887
....23893....23899....**23909**....**23911**....23917....23929....23957....23971
....23977....23981....23993....24001....24007....24019....24023....24029
....24043....24049....24061....24071....24077....24083....24091....24097
....24103....**24107**....**24109**....24113....24121....24133....24137....24151
....24169....**24179**....**24181**....24197....24203....24223....24229....24239
....24247....24251....24281....24317....24329....24337....24359....**24371**
....**24373**....24379....24391....24407....24413....**24419**....**24421**....24439
....24443....24469....24473....24481....24499....24509....24517....24527
....24533....24547....24551....24571....24593....24611....24623....24631
....24659....24671....24677....24683....24691....24697....24709....24733
....24749....24763....24767....24781....24793....24799....24809....24821
....24841....24847....24851....24859....24877....24889....24907....**24917**
....**24919**....24923....24943....24953....24967....24971....**24977**....**24979**
....24989....25013....**25031**....**25033**....25037....25057....25073....25087
....25097....25111....25117....25121....25127....25147....25153....25163
....**25169**....**25171**....25183....25189....25219....25229....25237....25243
....25247....25253....25261....**25301**....**25303**....**25307**....**25309**....25321
....25339....25343....25349....25357....25367....25373....25391....**25409**
....**25411**....25423....25439....25447....25453....25457....25463....**25469**
....**25471**....25523....25537....25541....25561....**25577**....**25579**....25583
....25589....**25601**....**25603**....25609....25621....25633....25639....25643
....25657....25667....25673....25679....25693....25703....25717....25733
....25741....25747....25759....25763....25771....25793....**25799**....**25801**
....25819....25841....**25847**....**25849**....25867....25873....25889....25903
....25913....25919....**25931**....**25933**....25939....25943....25951....25969
....25981....**25997**....**25999**....26003....26017....26021....26029....26041
....26053....26083....26099....26107....**26111**....**26113**....26119....26141
....26153....26161....26171....26177....26183....26189....26203....26209
....26227....26237....**26249**....**26251**....**26261**....**26263**....26267....26293
....26297....26309....26317....26321....26339....26347....26357....26371
....26387....26393....26399....26407....26417....26423....26431....26437
....26449....26459....26479....26489....26497....26501....26513....26539
....26557....26561....26573....26591....26597....26627....26633....26641
....26647....26669....**26681**....**26683**....26687....26693....**26699**....**26701**
....**26711**....**26713**....26717....26723....**26729**....**26731**....26737....26759
....26777....26783....26801....26813....26821....26833....26839....26849
....**26861**....**26863**....**26879**....**26881**....**26891**....**26893**....26903....26921
....26927....26947....**26951**....**26953**....26959....26981....26987....26993
....27011....27017....27031....27043....**27059**....**27061**....27067....27073
....27077....27091....27103....**27107**....**27109**....27127....27143....27179
....27191....27197....27211....**27239**....**27241**....27253....27259....27271
....27277....**27281**....**27283**....27299....27329....27337....27361....27367
....27397....**27407**....**27409**....27427....27431....27437....27449....27457
....**27479**....**27481**....27487....27509....**27527**....**27529**....**27539**....**27541**
....27551....**27581**....**27583**....27611....27617....27631....27647....27653
....27673....**27689**....**27691**....27697....27701....27733....**27737**....**27739**
....27743....**27749**....**27751**....27763....27767....27773....27779....**27791**
....**27793**....27799....27803....27809....27817....27823....27827....27847
....27851....27883....27893....27901....**27917**....**27919**....**27941**....**27943**
....27947....27953....27961....27967....27983....27997....28001....28019

....28027....28031....28051....28057....28069....28081....28087....28097
....28099....28109....28111....28123....28151....28163....28181....28183
....28201....28211....28219....28229....28277....28279....28283....28289
....28297....28307....28309....28319....28349....28351....28387....28393
....28403....28409....28411....28429....28433....28439....28447....28463
....28477....28493....28499....28513....28517....28537....28541....28547
....28549....28559....28571....28573....28579....28591....28597....28603
....28607....28619....28621....28627....28631....28643....28649....28657
....28661....28663....28669....28687....28697....28703....28711....28723
....28729....28751....28753....28759....28771....28789....28793....28807
....28813....28817....28837....28843....28859....28867....28871....28879
....28901....28909....28921....28927....28933....28949....28961....28979
....29009....29017....29021....29023....29027....29033....29059....29063
....29077....29101....29123....29129....29131....29137....29147....29153
....29167....29173....29179....29191....29201....29207....29209....29221
....29231....29243....29251....29269....29287....29297....29303....29311
....29327....29333....29339....29347....29363....29383....29387....29389
....29399....29401....29411....29423....29429....29437....29443....29453
....29473....29483....29501....29527....29531....29537....29567....29569
....29573....29581....29587....29599....29611....29629....29633....29641
....29663....29669....29671....29683....29717....29723....29741....29753
....29759....29761....29789....29803....29819....29833....29837....29851
....29863....29867....29873....29879....29881....29917....29921....29927
....29947....29959....29983....29989....30011....30013....30029....30047
....30059....30071....30089....30091....30097....30103....30109....30113
....30119....30133....30137....30139....30161....30169....30181....30187
....30197....30203....30211....30223....30241....30253....30259....30269
....30271....30293....30307....30313....30319....30323....30341....30347
....30367....30389....30391....30403....30427....30431....30449....30467
....30469....30491....30493....30497....30509....30517....30529....30539
....30553....30557....30559....30577....30593....30631....30637....30643
....30649....30661....30671....30677....30689....30697....30703....30707
....30713....30727....30757....30763....30773....30781....30803....30809
....30817....30829....30839....30841....30851....30853....30859....30869
....30871....30881....30893....30911....30931....30937....30941....30949
....30971....30977....30983....31013....31019....31033....31039....31051
....31063....31069....31079....31081....31091....31121....31123....31139
....31147....31151....31153....31159....31177....31181....31183....31189
....31193....31219....31223....31231....31237....31247....31249....31253
....31259....31267....31271....31277....31307....31319....31321....31327
....31333....31337....31357....31379....31387....31391....31393....31397
....31469....31477....31481....31489....31511....31513....31517....31531
....31541....31543....31547....31567....31573....31583....31601....31607
....31627....31643....31649....31657....31663....31667....31687....31699
....31721....31723....31727....31729....31741....31751....31769....31771
....31793....31799....31817....31847....31849....31859....31873....31883
....31891....31907....31957....31963....31973....31981....31991....32003
....32009....32027....32029....32051....32057....32059....32063....32069
....32077....32083....32089....32099....32117....32119....32141....32143
....32159....32173....32183....32189....32191....32203....32213....32233
....32237....32251....32257....32261....32297....32299....32303....32309
....32321....32323....32327....32341....32353....32359....32363....32369
....32371....32377....32381....32401....32411....32413....32423....32429
....32441....32443....32467....32479....32491....32497....32503....32507
....32531....32533....32537....32561....32563....32569....32573....32579
....32587....32603....32609....32611....32621....32633....32647....32653
....32687....32693....32707....32713....32717....32719....32749....32771
....32779....32783....32789....32797....32801....32803....32831....32833
....32839....32843....32869....32887....32909....32911....32917....32933
....32939....32941....32957....32969....32971....32983....32987....32993
....32999....33013....33023....33029....33037....33049....33053....33071
....33073....33083....33091....33107....33113....33119....33149....33151
....33161....33179....33181....33191....33199....33203....33211....33223
....33247....33287....33289....33301....33311....33317....33329....33331

....33343....**33347**....**33349**....33353....33359....33377....33391....33403
....33409....33413....33427....33457....33461....33469....33479....33487
....33493....33503....33521....33529....33533....33547....33563....33569
....33577....33581....**33587**....**33589**....**33599**....**33601**....33613....**33617**
....**33619**....33623....33629....33637....33641....33647....33679....33703
....33713....33721....33739....**33749**....**33751**....33757....**33767**....**33769**
....33773....33791....33797....**33809**....**33811**....**33827**....**33829**....33851
....33857....33863....33871....33889....33893....33911....33923....33931
....33937....33941....33961....33967....33997....34019....**34031**....**34033**
....34039....34057....34061....34123....**34127**....**34129**....34141....34147
....**34157**....**34159**....34171....34183....**34211**....**34213**....34217....34231
....34253....**34259**....**34261**....34267....34273....34283....34297....**34301**
....**34303**....34313....34319....34327....34337....34351....34361....**34367**
....**34369**....34381....34403....34421....34429....34439....34457....**34469**
....**34471**....34483....34487....**34499**....**34501**....**34511**....**34513**....34519
....34537....34543....34549....34583....**34589**....**34591**....34603....34607
....34613....34631....**34649**....**34651**....34667....34673....34679....34687
....34693....34703....34721....34729....34739....34747....**34757**....**34759**
....34763....34781....34807....34819....**34841**....**34843**....**34847**....**34849**
....34871....34877....34883....34897....34913....34919....34939....34949
....**34961**....**34963**....34981....35023....35027....**35051**....**35053**....35059
....35069....**35081**....**35083**....35089....35099....35107....35111....35117
....35129....35141....35149....35153....35159....35171....35201....35221
....35227....35251....35257....35267....**35279**....**35281**....35291....35311
....35317....35323....35327....35339....35353....35363....35381....35393
....35401....35407....35419....35423....35437....**35447**....**35449**....35461
....35491....**35507**....**35509**....35521....35527....**35531**....**35533**....35537
....35543....35569....35573....**35591**....**35593**....35597....35603....35617
....35671....35677....**35729**....**35731**....35747....35753....35759....35771
....35797....**35801**....**35803**....35809....35831....**35837**....**35839**....35851
....35863....35869....35879....**35897**....**35899**....35911....35923....35933
....35951....35963....35969....35977....35983....35993....35999....36007
....**36011**....**36013**....36017....36037....36061....36067....36073....36083
....36097....**36107**....**36109**....36131....36137....36151....36161....36187
....36191....36209....36217....36229....36241....36251....36263....36269
....36277....36293....36299....36307....36313....36319....**36341**....**36343**
....36353....36373....36383....36389....36433....36451....36457....**36467**
....**36469**....36473....36479....36493....36497....36523....**36527**....**36529**
....36541....36551....36559....36563....36571....36583....36587....36599
....36607....36629....36637....36643....36653....36671....36677....36683
....36691....36697....36709....36713....36721....36739....36749....36761
....36767....**36779**....**36781**....36787....**36791**....**36793**....36809....36821
....36833....36847....36857....36871....36877....36887....**36899**....**36901**
....36913....36919....36923....**36929**....**36931**....36943....36947....36973
....36979....36997....37003....37013....**37019**....**37021**....37039....37049
....37057....37061....37087....37097....37117....37123....37139....37159
....37171....37181....37189....**37199**....**37201**....37217....37223....37243
....37253....37273....37277....**37307**....**37309**....37313....37321....**37337**
....**37339**....37357....**37361**....**37363**....37369....37379....37397....37409
....37423....37441....37447....37463....37483....37489....37493....37501
....37507....37511....37517....37529....37537....**37547**....**37549**....37561
....37567....**37571**....**37573**....37579....**37589**....**37591**....37607....37619
....37633....37643....37649....37657....37663....**37691**....**37693**....37699
....37717....37747....**37781**....**37783**....37799....**37811**....**37813**....37831
....37847....37853....37861....37871....37879....37889....37897....37907
....37951....37957....37963....37967....37987....**37991**....**37993**....37997
....38011....38039....38047....38053....38069....38083....38113....38119
....38149....38153....38167....38177....38183....38189....38197....38201
....38219....38231....**38237**....**38239**....38261....38273....38281....38287
....38299....38303....38317....38321....**38327**....**38329**....38333....38351
....38371....38377....38393....38431....**38447**....**38449**....38453....**38459**
....**38461**....38501....38543....38557....38561....**38567**....**38569**....38593
....38603....**38609**....**38611**....38629....38639....**38651**....**38653**....**38669**
....**38671**....38677....38693....38699....38707....**38711**....**38713**....38723

....38729....38737....38747....38749....38767....38783....38791....38803
....38821....38833....38839....38851....38861....38867....38873....38891
....38903....38917....38921....38923....38933....38953....38959....38971
....38977....38993....39019....39023....39041....39043....39047....39079
....39089....39097....39103....39107....39113....39119....39133....39139
....39157....39161....39163....39181....39191....39199....39209....39217
....39227....39229....39233....39239....39241....39251....39293....39301
....39313....39317....39323....39341....39343....39359....39367....39371
....39373....39383....39397....39409....39419....39439....39443....39451
....39461....39499....39503....39509....39511....39521....39541....39551
....39563....39569....39581....39607....39619....39623....39631....39659
....39667....39671....39679....39703....39709....39719....39727....39733
....39749....39761....39769....39779....39791....39799....39821....39827
....39829....39839....39841....39847....39857....39863....39869....39877
....39883....39887....39901....39929....39937....39953....39971....39979
....39983....39989....40009....40013....40031....40037....40039....40063
....40087....40093....40099....40111....40123....40127....40129....40151
....40153....40163....40169....40177....40189....40193....40213....40231
....40237....40241....40253....40277....40283....40289....40343....40351
....40357....40361....40387....40423....40427....40429....40433....40459
....40471....40483....40487....40493....40499....40507....40519....40529
....40531....40543....40559....40577....40583....40591....40597....40609
....40627....40637....40639....40693....40697....40699....40709....40739
....40751....40759....40763....40771....40787....40801....40813....40819
....40823....40829....40841....40847....40849....40853....40867....40879
....40883....40897....40903....40927....40933....40939....40949....40961
....40973....40993....41011....41017....41023....41039....41047....41051
....41057....41077....41081....41113....41117....41131....41141....41143
....41149....41161....41177....41179....41183....41189....41201....41203
....41213....41221....41227....41231....41233....41243....41257....41263
....41269....41281....41299....41333....41341....41351....41357....41381
....41387....41389....41399....41411....41413....41443....41453....41467
....41479....41491....41507....41513....41519....41521....41539....41543
....41549....41579....41593....41597....41603....41609....41611....41617
....41621....41627....41641....41647....41651....41659....41669....41681
....41687....41719....41729....41737....41759....41761....41771....41777
....41801....41809....41813....41843....41849....41851....41863....41879
....41887....41893....41897....41903....41911....41927....41941....41947
....41953....41957....41959....41969....41981....41983....41999....42013
....42017....42019....42023....42043....42061....42071....42073....42083
....42089....42101....42131....42139....42157....42169....42179....42181
....42187....42193....42197....42209....42221....42223....42227....42239
....42257....42281....42283....42293....42299....42307....42323....42331
....42337....42349....42359....42373....42379....42391....42397....42403
....42407....42409....42433....42437....42443....42451....42457....42461
....42463....42467....42473....42487....42491....42499....42509....42533
....42557....42569....42571....42577....42589....42611....42641....42643
....42649....42667....42677....42683....42689....42697....42701....42703
....42709....42719....42727....42737....42743....42751....42767....42773
....42787....42793....42797....42821....42829....42839....42841....42853
....42859....42863....42899....42901....42923....42929....42937....42943
....42953....42961....42967....42979....42989....43003....43013....43019
....43037....43049....43051....43063....43067....43093....43103....43117
....43133....43151....43159....43177....43189....43201....43207....43223
....43237....43261....43271....43283....43291....43313....43319....43321
....43331....43391....43397....43399....43403....43411....43427....43441
....43451....43457....43481....43487....43499....43517....43541....43543
....43573....43577....43579....43591....43597....43607....43609....43613
....43627....43633....43649....43651....43661....43669....43691....43711
....43717....43721....43753....43759....43777....43781....43783....43787
....43789....43793....43801....43853....43867....43889....43891....43913
....43933....43943....43951....43961....43963....43969....43973....43987
....43991....43997....44017....44021....44027....44029....44041....44053
....44059....44071....44087....44089....44101....44111....44119....44123

Anlage 1

....**44129**....**44131**....44159....44171....44179....44189....**44201**....**44203**
....44207....44221....44249....44257....44263....**44267**....**44269**....44273
....**44279**....**44281**....44293....44351....44357....44371....**44381**....**44383**
....44389....44417....44449....44453....44483....44491....44497....44501
....44507....44519....**44531**....**44533**....44537....44543....44549....44563
....44579....44587....44617....**44621**....**44623**....44633....44641....44647
....44651....44657....44683....44687....**44699**....**44701**....44711....44729
....44741....44753....**44771**....**44773**....44777....44789....44797....44809
....44819....44839....44843....44851....44867....44879....44887....44893
....44909....44917....44927....44939....44953....44959....44963....44971
....44983....44987....45007....45013....45053....45061....45077....45083
....**45119**....**45121**....45127....45131....**45137**....**45139**....45161....**45179**
....**45181**....45191....45197....45233....45247....45259....45263....45281
....45289....45293....45307....**45317**....**45319**....45329....45337....**45341**
....**45343**....45361....45377....45389....45403....45413....45427....45433
....45439....45481....45491....45497....45503....45523....45533....45541
....45553....45557....45569....**45587**....**45589**....45599....45613....45631
....45641....45659....45667....45673....45677....45691....45697....45707
....45737....45751....45757....45763....45767....45779....45817....**45821**
....**45823**....45827....45833....45841....45853....45863....45869....45887
....45893....45943....45949....45953....45959....45971....45979....45989
....46021....46027....**46049**....**46051**....46061....46073....**46091**....**46093**
....46099....46103....46133....46141....46147....46153....46171....**46181**
....**46183**....46187....46199....46219....46229....46237....46261....**46271**
....**46273**....46279....46301....**46307**....**46309**....46327....46337....**46349**
....**46351**....46381....46399....46411....**46439**....**46441**....46447....46451
....46457....46471....46477....46489....46499....46507....46511....46523
....46549....46559....46567....46573....**46589**....**46591**....46601....46619
....46633....46639....46643....46649....46663....**46679**....**46681**....46687
....46691....46703....46723....46727....46747....46751....46757....**46769**
....**46771**....46807....46811....**46817**....**46819**....**46829**....**46831**....46853
....46861....46867....46877....46889....46901....46919....46933....46957
....46993....46997....47017....47041....47051....**47057**....**47059**....47087
....47093....47111....47119....47123....47129....47137....47143....**47147**
....**47149**....47161....47189....47207....47221....47237....47251....47269
....47279....47287....47293....47297....47303....47309....47317....47339
....**47351**....**47353**....47363....47381....**47387**....**47389**....47407....**47417**
....**47419**....47431....47441....47459....47491....47497....47501....47507
....47513....47521....47527....47533....47543....47563....47569....47581
....47591....47599....47609....47623....47629....47639....47653....**47657**
....**47659**....47681....**47699**....**47701**....**47711**....**47713**....47717....47737
....**47741**....**47743**....**47777**....**47779**....47791....47797....**47807**....**47809**
....47819....47837....47843....47857....47869....47881....47903....47911
....47917....47933....47939....47947....47951....47963....47969....47977
....47981....48017....48023....48029....48049....48073....48079....48091
....48109....**48119**....**48121**....48131....48157....48163....48179....48187
....48193....48197....48221....48239....48247....48259....48271....48281
....48299....**48311**....**48313**....48337....48341....48353....48371....48383
....48397....**48407**....**48409**....48413....48437....48449....48463....48473
....**48479**....**48481**....48487....48491....48497....48523....48527....48533
....**48539**....**48541**....48563....48571....48589....48593....48611....48619
....48623....**48647**....**48649**....48661....48673....**48677**....**48679**....**48731**
....**48733**....48751....48757....48761....48767....**48779**....**48781**....48787
....48799....48809....48817....**48821**....**48823**....48847....**48857**....**48859**
....**48869**....**48871**....48883....48889....48907....48947....48953....48973
....**48989**....**48991**....49003....49009....49019....**49031**....**49033**....49037
....49043....49057....49069....49081....49103....49109....49117....**49121**
....**49123**....49139....49157....**49169**....**49171**....49177....49193....**49199**
....**49201**....49207....49211....49223....49253....49261....**49277**....**49279**
....49297....49307....**49331**....**49333**....49339....49363....**49367**....**49369**
....**49391**....**49393**....**49409**....**49411**....49417....49429....49433....49451
....49459....49463....49477....49481....49499....49523....**49529**....**49531**
....49537....**49547**....**49549**....49559....49597....49603....49613....49627
....49633....49639....49663....**49667**....**49669**....49681....49697....49711

Anlage 1

....49727....49739....49741....49747....49757....49783....49787....49789
....49801....49807....49811....49823....49831....49843....49853....49871
....49877....49891....49919....49921....49927....49937....49939....49943
....49957....49991....49993....49999....50021....50023....50033....50047
....50051....50053....50069....50077....50087....50093....50101....50111
....50119....50123....50129....50131....50147....50153....50159....50177
....50207....50221....50227....50231....50261....50263....50273....50287
....50291....50311....50321....50329....50333....50341....50359....50363
....50377....50383....50387....50411....50417....50423....50441....50459
....50461....50497....50503....50513....50527....50539....50543....50549
....50551....50581....50587....50591....50593....50599....50627....50647
....50651....50671....50683....50707....50723....50741....50753....50767
....50773....50777....50789....50821....50833....50839....50849....50857
....50867....50873....50891....50893....50909....50923....50929....50951
....50957....50969....50971....50989....50993....51001....51031....51043
....51047....51059....51061....51071....51109....51131....51133....51137
....51151....51157....51169....51193....51197....51199....51203....51217
....51229....51239....51241....51257....51263....51283....51287....51307
....51329....51341....51343....51347....51349....51361....51383....51407
....51413....51419....51421....51427....51431....51437....51439....51449
....51461....51473....51479....51481....51487....51503....51511....51517
....51521....51539....51551....51563....51577....51581....51593....51599
....51607....51613....51631....51637....51647....51659....51673....51679
....51683....51691....51713....51719....51721....51749....51767....51769
....51787....51797....51803....51817....51827....51829....51839....51853
....51859....51869....51871....51893....51899....51907....51913....51929
....51941....51949....51971....51973....51977....51991....52009....52021
....52027....52051....52057....52067....52069....52081....52103....52121
....52127....52147....52153....52163....52177....52181....52183....52189
....52201....52223....52237....52249....52253....52259....52267....52289
....52291....52301....52313....52321....52361....52363....52369....52379
....52387....52391....52433....52453....52457....52489....52501....52511
....52517....52529....52541....52543....52553....52561....52567....52571
....52579....52583....52609....52627....52631....52639....52667....52673
....52691....52697....52709....52711....52721....52727....52733....52747
....52757....52769....52783....52807....52813....52817....52837....52859
....52861....52879....52883....52889....52901....52903....52919....52937
....52951....52957....52963....52967....52973....52981....52999....53003
....53017....53047....53051....53069....53077....53087....53089....53093
....53101....53113....53117....53129....53147....53149....53161....53171
....53173....53189....53197....53201....53231....53233....53239....53267
....53269....53279....53281....53299....53309....53323....53327....53353
....53359....53377....53381....53401....53407....53411....53419....53437
....53441....53453....53479....53503....53507....53527....53549....53551
....53569....53591....53593....53597....53609....53611....53617....53623
....53629....53633....53639....53653....53657....53681....53693....53699
....53717....53719....53731....53759....53773....53777....53783....53791
....53813....53819....53831....53849....53857....53861....53881....53887
....53891....53897....53899....53917....53923....53927....53939....53951
....53959....53987....53993....54001....54011....54013....54037....54049
....54059....54083....54091....54101....54121....54133....54139....54151
....54163....54167....54181....54193....54217....54251....54269....54277
....54287....54293....54311....54319....54323....54331....54347....54361
....54367....54371....54377....54401....54403....54409....54413....54419
....54421....54437....54443....54449....54469....54493....54497....54499
....54503....54517....54521....54539....54541....54547....54559....54563
....54577....54581....54583....54601....54617....54623....54629....54631
....54647....54667....54673....54679....54709....54713....54721....54727
....54751....54767....54773....54779....54787....54799....54829....54833
....54851....54869....54877....54881....54907....54917....54919....54941
....54949....54959....54973....54979....54983....55001....55009....55021
....55049....55051....55057....55061....55073....55079....55103....55109
....55117....55127....55147....55163....55171....55201....55207....55213
....55217....55219....55229....55243....55249....55259....55291....55313

Anlage 1

....**55331**....**55333**....**55337**....**55339**....55343....55351....55373....55381
....55399....55411....**55439**....**55441**....55457....55469....55487....55501
....55511....55529....55541....55547....55579....55589....55603....55609
....**55619**....**55621**....**55631**....**55633**....55639....**55661**....**55663**....55667
....55673....55681....55691....55697....55711....55717....55721....55733
....55763....55787....55793....55799....55807....55813....**55817**....**55819**
....55823....55829....55837....55843....55849....55871....55889....55897
....**55901**....**55903**....55921....55927....**55931**....**55933**....55949....55967
....55987....55997....56003....56009....**56039**....**56041**....56053....56081
....56087....56093....**56099**....**56101**....56113....56123....56131....56149
....56167....56171....56179....56197....**56207**....**56209**....**56237**....**56239**
....56249....56263....**56267**....**56269**....56299....56311....56333....56359
....56369....56377....56383....56393....56401....56417....56431....56437
....56443....56453....56467....56473....**56477**....**56479**....56489....56501
....**56503**....56509....56519....56527....**56531**....**56533**....56543....56569
....56591....**56597**....**56599**....56611....56629....56633....56659....56663
....56671....56681....56687....56701....**56711**....**56713**....56731....56737
....56747....56767....56773....56779....56783....**56807**....**56809**....56813
....56821....56827....56843....56857....56873....**56891**....**56893**....56897
....**56909**....**56911**....**56921**....**56923**....56929....56941....56951....56957
....56963....56983....56989....56993....56999....57037....57041....57047
....57059....57073....57077....57089....57097....57107....57119....57131
....57139....57143....57149....57163....57173....57179....**57191**....**57193**
....57203....**57221**....**57223**....57241....57251....57259....**57269**....**57271**
....57283....57287....57301....**57329**....**57331**....**57347**....**57349**....57367
....57373....57383....57389....57397....57413....57427....57457....57467
....57487....57493....57503....**57527**....**57529**....**57557**....**57559**....57571
....57587....57593....57601....57637....57641....57649....57653....57667
....57679....57689....57697....57709....57713....57719....57727....57731
....57737....57751....57773....57781....57787....**57791**....**57793**....57803
....57809....57829....57839....57847....57853....57859....57881....**57899**
....**57901**....57917....57923....57943....57947....57973....57977....57991
....58013....58027....58031....58043....58049....58057....58061....58067
....58073....58099....**58109**....**58111**....58129....58147....**58151**....**58153**
....**58169**....**58171**....58189....58193....58199....58207....58211....58217
....**58229**....**58231**....58237....58243....58271....58309....58313....58321
....58337....58363....**58367**....**58369**....58379....**58391**....**58393**....58403
....58411....58417....58427....**58439**....**58441**....**58451**....**58453**....58477
....58481....58511....58537....58543....58549....58567....58573....58579
....**58601**....**58603**....58613....58631....58657....58661....58679....58687
....58693....58699....58711....58727....58733....58741....58757....58763
....58771....**58787**....**58789**....58831....58889....58897....58901....**58907**
....**58909**....58913....58921....58937....58943....58963....58967....58979
....58991....58997....**59009**....**59011**....**59021**....**59023**....59029....**59051**
....**59053**....59063....59069....59077....59083....59093....59107....59113
....59119....59123....59141....59149....59159....59167....59183....59197
....**59207**....**59209**....**59219**....**59221**....59233....59239....59243....59263
....59273....59281....59333....59341....59351....**59357**....**59359**....59369
....59377....59387....59393....59399....59407....**59417**....**59419**....**59441**
....**59443**....59447....59453....59467....**59471**....**59473**....59497....59509
....59513....59539....59557....59561....59567....59581....59611....59617
....59621....**59627**....**59629**....59651....59659....59663....**59669**....**59671**
....59693....59699....59707....59723....59729....59743....59747....59753
....59771....59779....59791....59797....59809....59833....59863....59879
....59887....59921....59929....59951....59957....59971....59981....59999
....60013....60017....60029....60037....60041....60077....60083....**60089**
....**60091**....**60101**....**60103**....60107....60127....60133....60139....60149
....60161....**60167**....**60169**....60209....60217....60223....60251....**60257**
....**60259**....60271....60289....60293....60317....60331....60337....60343
....60353....60373....60383....60397....60413....60427....60443....60449
....60457....60493....60497....60509....60521....60527....60539....60589
....60601....60607....60611....60617....60623....60631....60637....**60647**
....**60649**....**60659**....**60661**....60679....60689....60703....60719....60727
....60733....60737....60757....**60761**....**60763**....60773....60779....60793

Anlage 1

....60811....60821....60859....60869....60887....60889....60899....60901
....60913....60917....60919....60923....60937....60943....60953....60961
....61001....61007....61027....61031....61043....61051....61057....61091
....61099....61121....61129....61141....61151....61153....61169....61211
....61223....61231....61253....61261....61283....61291....61297....61331
....61333....61339....61343....61357....61363....61379....61381....61403
....61409....61417....61441....61463....61469....61471....61483....61487
....61493....61507....61511....61519....61543....61547....61553....61559
....61561....61583....61603....61609....61613....61627....61631....61637
....61643....61651....61657....61667....61673....61681....61687....61703
....61717....61723....61729....61751....61757....61781....61813....61819
....61837....61843....61861....61871....61879....61909....61927....61933
....61949....61961....61967....61979....61981....61987....61991....62003
....62011....62017....62039....62047....62053....62057....62071....62081
....62099....62119....62129....62131....62137....62141....62143....62171
....62189....62191....62201....62207....62213....62219....62233....62273
....62297....62299....62303....62311....62323....62327....62347....62351
....62383....62401....62417....62423....62459....62467....62473....62477
....62483....62497....62501....62507....62533....62539....62549....62563
....62581....62591....62597....62603....62617....62627....62633....62639
....62653....62659....62683....62687....62701....62723....62731....62743
....62753....62761....62773....62791....62801....62819....62827....62851
....62861....62869....62873....62897....62903....62921....62927....62929
....62939....62969....62971....62981....62983....62987....62989....63029
....63031....63059....63067....63073....63079....63097....63103....63113
....63127....63131....63149....63179....63197....63199....63211....63241
....63247....63277....63281....63299....63311....63313....63317....63331
....63337....63347....63353....63361....63367....63377....63389....63391
....63397....63409....63419....63421....63439....63443....63463....63467
....63473....63487....63493....63499....63521....63527....63533....63541
....63559....63577....63587....63589....63599....63601....63607....63611
....63617....63629....63647....63649....63659....63667....63671....63689
....63691....63697....63703....63709....63719....63727....63737....63743
....63761....63773....63781....63793....63799....63803....63809....63823
....63839....63841....63853....63857....63863....63901....63907....63913
....63929....63949....63977....63997....64007....64013....64019....64033
....64037....64063....64067....64081....64091....64109....64123....64151
....64153....64157....64171....64187....64189....64217....64223....64231
....64237....64271....64279....64283....64301....64303....64319....64327
....64333....64373....64381....64399....64403....64433....64439....64451
....64453....64483....64489....64499....64513....64553....64567....64577
....64579....64591....64601....64609....64613....64621....64627....64633
....64661....64663....64667....64679....64693....64709....64717....64747
....64763....64781....64783....64793....64811....64817....64849....64853
....64871....64877....64879....64891....64901....64919....64921....64927
....64937....64951....64969....64997....65003....65011....65027....65029
....65033....65053....65063....65071....65089....65099....65101....65111
....65119....65123....65129....65141....65147....65167....65171....65173
....65179....65183....65203....65213....65239....65257....65267....65269
....65287....65293....65309....65323....65327....65353....65357....65371
....65381....65393....65407....65413....65419....65423....65437....65447
....65449....65479....65497....65519....65521....65537....65539....65543
....65551....65557....65563....65579....65581....65587....65599....65609
....65617....65629....65633....65647....65651....65657....65677....65687
....65699....65701....65707....65713....65717....65719....65729....65731
....65761....65777....65789....65809....65827....65831....65837....65839
....65843....65851....65867....65881....65899....65921....65927....65929
....65951....65957....65963....65981....65983....65993....66029....66037
....66041....66047....66067....66071....66083....66089....66103....66107
....66109....66137....66161....66169....66173....66179....66191....66221
....66239....66271....66293....66301....66337....66343....66347....66359
....66361....66373....66377....66383....66403....66413....66431....66449
....66457....66463....66467....66491....66499....66509....66523....66529
....66533....66541....66553....66569....66571....66587....66593....66601

Anlage 1

....66617....66629....66643....66653....66683....66697....66701....66713
....66721....66733....66739....66749....66751....66763....66791....66797
....66809....66821....66841....66851....66853....66863....66877....66883
....66889....66919....66923....66931....66943....66947....66949....66959
....66973....66977....67003....67021....67033....67043....67049....67057
....67061....67073....67079....67103....67121....67129....67139....67141
....67153....67157....67169....67181....67187....67189....67211....67213
....67217....67219....67231....67247....67261....67271....67273....67289
....67307....67339....67343....67349....67369....67391....67399....67409
....67411....67421....67427....67429....67433....67447....67453....67477
....67481....67489....67493....67499....67511....67523....67531....67537
....67547....67559....67567....67577....67579....67589....67601....67607
....67619....67631....67651....67679....67699....67709....67723....67733
....67741....67751....67757....67759....67763....67777....67783....67789
....67801....67807....67819....67829....67843....67853....67867....67883
....67891....67901....67927....67931....67933....67939....67943....67957
....67961....67967....67979....67987....67993....68023....68041....68053
....68059....68071....68087....68099....68111....68113....68141....68147
....68161....68171....68207....68209....68213....68219....68227....68239
....68261....68279....68281....68311....68329....68351....68371....68389
....68399....68437....68443....68447....68449....68473....68477....68483
....68489....68491....68501....68507....68521....68531....68539....68543
....68567....68581....68597....68611....68633....68639....68659....68669
....68683....68687....68699....68711....68713....68729....68737....68743
....68749....68767....68771....68777....68791....68813....68819....68821
....68863....68879....68881....68891....68897....68899....68903....68909
....68917....68927....68947....68963....68993....69001....69011....69019
....69029....69031....69061....69067....69073....69109....69119....69127
....69143....69149....69151....69163....69191....69193....69197....69203
....69221....69233....69239....69247....69257....69259....69263....69313
....69317....69337....69341....69371....69379....69383....69389....69401
....69403....69427....69431....69439....69457....69463....69467....69473
....69481....69491....69493....69497....69499....69539....69557....69593
....69623....69653....69661....69677....69691....69697....69709....69737
....69739....69761....69763....69767....69779....69809....69821....69827
....69829....69833....69847....69857....69859....69877....69899....69911
....69929....69931....69941....69959....69991....69997....70001....70003
....70009....70019....70039....70051....70061....70067....70079....70099
....70111....70117....70121....70123....70139....70141....70157....70163
....70177....70181....70183....70199....70201....70207....70223....70229
....70237....70241....70249....70271....70289....70297....70309....70313
....70321....70327....70351....70373....70379....70381....70393....70423
....70429....70439....70451....70457....70459....70481....70487....70489
....70501....70507....70529....70537....70549....70571....70573....70583
....70589....70607....70619....70621....70627....70639....70657....70663
....70667....70687....70709....70717....70729....70753....70769....70783
....70793....70823....70841....70843....70849....70853....70867....70877
....70879....70891....70901....70913....70919....70921....70937....70949
....70951....70957....70969....70979....70981....70991....70997....70999
....71011....71023....71039....71059....71069....71081....71089....71119
....71129....71143....71147....71153....71161....71167....71171....71191
....71209....71233....71237....71249....71257....71261....71263....71287
....71293....71317....71327....71329....71333....71339....71341....71347
....71353....71359....71363....71387....71389....71399....71411....71413
....71419....71429....71437....71443....71453....71471....71473....71479
....71483....71503....71527....71537....71549....71551....71563....71569
....71593....71597....71633....71647....71663....71671....71693....71699
....71707....71711....71713....71719....71741....71761....71777....71789
....71807....71809....71821....71837....71843....71849....71861....71867
....71879....71881....71887....71899....71909....71917....71933....71941
....71947....71963....71971....71983....71987....71993....71999....72019
....72031....72043....72047....72053....72073....72077....72089....72091
....72101....72103....72109....72139....72161....72167....72169....72173
....72211....72221....72223....72227....72229....72251....72253....72269

....72271....72277....72287....72307....72313....72337....72341....72353

....72367....72379....72383....72421....72431....72461....72467....72469

....72481....72493....72497....72503....72533....72547....72551....72559

....72577....72613....72617....72623....72643....72647....72649....72661

....72671....72673....72679....72689....72701....72707....72719....72727

....72733....72739....72763....72767....72797....72817....72823....72859

....72869....72871....72883....72889....72893....72901....72907....72911

....72923....72931....72937....72949....72953....72959....72973....72977

....72997....73009....73013....73019....73037....73039....73043....73061

....73063....73079....73091....73121....73127....73133....73141....73181

....73189....73237....73243....73259....73277....73291....73303....73309

....73327....73331....73351....73361....73363....73369....73379....73387

....73417....73421....73433....73453....73459....73471....73477....73483

....73517....73523....73529....73547....73553....73561....73571....73583

....73589....73597....73607....73609....73613....73637....73643....73651

....73673....73679....73681....73693....73699....73709....73721....73727

....73751....73757....73771....73783....73819....73823....73847....73849

....73859....73867....73877....73883....73897....73907....73939....73943

....73951....73961....73973....73999....74017....74021....74027....74047

....74051....74071....74077....74093....74099....74101....74131....74143

....74149....74159....74161....74167....74177....74189....74197....74201

....74203....74209....74219....74231....74257....74279....74287....74293

....74297....74311....74317....74323....74353....74357....74363....74377

....74381....74383....74411....74413....74419....74441....74449....74453

....74471....74489....74507....74509....74521....74527....74531....74551

....74561....74567....74573....74587....74597....74609....74611....74623

....74653....74687....74699....74707....74713....74717....74719....74729

....74731....74747....74759....74761....74771....74779....74797....74821

....74827....74831....74843....74857....74861....74869....74873....74887

....74891....74897....74903....74923....74929....74933....74941....74959

....75011....75013....75017....75029....75037....75041....75079....75083

....75109....75133....75149....75161....75167....75169....75181....75193

....75209....75211....75217....75223....75227....75239....75253....75269

....75277....75289....75307....75323....75329....75337....75347....75353

....75367....75377....75389....75391....75401....75403....75407....75431

....75437....75479....75503....75511....75521....75527....75533....75539

....75541....75553....75557....75571....75577....75583....75611....75617

....75619....75629....75641....75653....75659....75679....75683....75689

....75703....75707....75709....75721....75731....75743....75767....75773

....75781....75787....75793....75797....75821....75833....75853....75869

....75883....75913....75931....75937....75941....75967....75979....75983

....75989....75991....75997....76001....76003....76031....76039....76079

....76081....76091....76099....76103....76123....76129....76147....76157

....76159....76163....76207....76213....76231....76243....76249....76253

....76259....76261....76283....76289....76303....76333....76343....76367

....76369....76379....76387....76403....76421....76423....76441....76463

....76471....76481....76487....76493....76507....76511....76519....76537

....76541....76543....76561....76579....76597....76603....76607....76631

....76649....76651....76667....76673....76679....76697....76717....76733

....76753....76757....76771....76777....76781....76801....76819....76829

....76831....76837....76847....76871....76873....76883....76907....76913

....76919....76943....76949....76961....76963....76991....77003....77017

....77023....77029....77041....77047....77069....77081....77093....77101

....77137....77141....77153....77167....77171....77191....77201....77213

....77237....77239....77243....77249....77261....77263....77267....77269

....77279....77291....77317....77323....77339....77347....77351....77359

....77369....77377....77383....77417....77419....77431....77447....77471

....77477....77479....77489....77491....77509....77513....77521....77527

....77543....77549....77551....77557....77563....77569....77573....77587

....77591....77611....77617....77621....77641....77647....77659....77681

....77687....77689....77699....77711....77713....77719....77723....77731

....77743....77747....77761....77773....77783....77797....77801....77813

....77839....77849....77863....77867....77893....77899....77929....77933

....77951....77969....77977....77983....77999....78007....78017....78031

Anlage 1

....78041....78049....78059....78079....78101....78121....78137....78139
....78157....78163....78167....78173....78179....78191....78193....78203
....78229....78233....78241....78259....78277....78283....78301....78307
....78311....78317....78341....78347....78367....78401....78427....78437
....78439....78467....78479....78487....78497....78509....78511....78517
....78539....78541....78553....78569....78571....78577....78583....78593
....78607....78623....78643....78649....78653....78691....78697....78707
....78713....78721....78737....78779....78781....78787....78791....78797
....78803....78809....78823....78839....78853....78857....78877....78887
....78889....78893....78901....78919....78929....78941....78977....78979
....78989....79031....79039....79043....79063....79087....79103....79111
....79133....79139....79147....79151....79153....79159....79181....79187
....79193....79201....79229....79231....79241....79259....79273....79279
....79283....79301....79309....79319....79333....79337....79349....79357
....79367....79379....79393....79397....79399....79411....79423....79427
....79433....79451....79481....79493....79531....79537....79549....79559
....79561....79579....79589....79601....79609....79613....79621....79627
....79631....79633....79657....79669....79687....79691....79693....79697
....79699....79757....79769....79777....79801....79811....79813....79817
....79823....79829....79841....79843....79847....79861....79867....79873
....79889....79901....79903....79907....79939....79943....79967....79973
....79979....79987....79997....79999....80021....80039....80051....80071
....80077....80107....80111....80141....80147....80149....80153....80167
....80173....80177....80191....80207....80209....80221....80231....80233
....80239....80251....80263....80273....80279....80287....80309....80317
....80329....80341....80347....80363....80369....80387....80407....80429
....80447....80449....80471....80473....80489....80491....80513....80527
....80537....80557....80567....80599....80603....80611....80621....80627
....80629....80651....80657....80669....80671....80677....80681....80683
....80687....80701....80713....80737....80747....80749....80761....80777
....80779....80783....80789....80803....80809....80819....80831....80833
....80849....80863....80897....80909....80911....80917....80923....80929
....80933....80953....80963....80989....81001....81013....81017....81019
....81023....81031....81041....81043....81047....81049....81071....81077
....81083....81097....81101....81119....81131....81157....81163....81173
....81181....81197....81199....81203....81223....81233....81239....81281
....81283....81293....81299....81307....81331....81343....81349....81353
....81359....81371....81373....81401....81409....81421....81439....81457
....81463....81509....81517....81527....81533....81547....81551....81553
....81559....81563....81569....81611....81619....81629....81637....81647
....81649....81667....81671....81677....81689....81701....81703....81707
....81727....81737....81749....81761....81769....81773....81799....81817
....81839....81847....81853....81869....81883....81899....81901....81919
....81929....81931....81937....81943....81953....81967....81971....81973
....82003....82007....82009....82013....82021....82031....82037....82039
....82051....82067....82073....82129....82139....82141....82153....82163
....82171....82183....82189....82193....82207....82217....82219....82223
....82231....82237....82241....82261....82267....82279....82301....82307
....82339....82349....82351....82361....82373....82387....82393....82421
....82457....82463....82469....82471....82483....82487....82493....82499
....82507....82529....82531....82549....82559....82561....82567....82571
....82591....82601....82609....82613....82619....82633....82651....82657
....82699....82721....82723....82727....82729....82757....82759....82763
....82781....82787....82793....82799....82811....82813....82837....82847
....82883....82889....82891....82903....82913....82939....82963....82981
....82997....83003....83009....83023....83047....83059....83063....83071
....83077....83089....83093....83101....83117....83137....83177....83203
....83207....83219....83221....83227....83231....83233....83243....83257
....83267....83269....83273....83299....83311....83339....83341....83357
....83383....83389....83399....83401....83407....83417....83423....83431
....83437....83443....83449....83459....83471....83477....83497....83537
....83557....83561....83563....83579....83591....83597....83609....83617
....83621....83639....83641....83653....83663....83689....83701....83717
....83719....83737....83761....83773....83777....83791....83813....83833

....83843....83857....83869....83873....83891....83903....83911....83921
....83933....83939....83969....83983....83987....84011....84017....84047
....84053....84059....84061....84067....84089....84121....84127....84131
....84137....84143....84163....84179....84181....84191....84199....84211
....84221....84223....84229....84239....84247....84263....84299....84307
....84313....84317....84319....84347....84349....84377....84389....84391
....84401....84407....84421....84431....84437....84443....84449....84457
....84463....84467....84481....84499....84503....84509....84521....84523
....84533....84551....84559....84589....84629....84631....84649....84653
....84659....84673....84691....84697....84701....84713....84719....84731
....84737....84751....84761....84787....84793....84809....84811....84827
....84857....84859....84869....84871....84913....84919....84947....84961
....84967....84977....84979....84991....85009....85021....85027....85037
....85049....85061....85081....85087....85091....85093....85103....85109
....85121....85133....85147....85159....85193....85199....85201....85213
....85223....85229....85237....85243....85247....85259....85297....85303
....85313....85331....85333....85361....85363....85369....85381....85411
....85427....85429....85439....85447....85451....85453....85469....85487
....85513....85517....85523....85531....85549....85571....85577....85597
....85601....85607....85619....85621....85627....85639....85643....85661
....85667....85669....85691....85703....85711....85717....85733....85751
....85781....85793....85817....85819....85829....85831....85837....85843
....85847....85853....85889....85903....85909....85931....85933....85991
....85999....86011....86017....86027....86029....86069....86077....86083
....86111....86113....86117....86131....86137....86143....86161....86171
....86179....86183....86197....86201....86209....86239....86243....86249
....86257....86263....86269....86287....86291....86293....86297....86311
....86323....86341....86351....86353....86357....86369....86371....86381
....86389....86399....86413....86423....86441....86453....86461....86467
....86477....86491....86501....86509....86531....86533....86539....86561
....86573....86579....86587....86599....86627....86629....86677....86689
....86693....86711....86719....86729....86743....86753....86767....86771
....86783....86813....86837....86843....86851....86857....86861....86869
....86923....86927....86929....86939....86951....86959....86969....86981
....86993....87011....87013....87037....87041....87049....87071....87083
....87103....87107....87119....87121....87133....87149....87151....87179
....87181....87187....87211....87221....87223....87251....87253....87257
....87277....87281....87293....87299....87313....87317....87323....87337
....87359....87383....87403....87407....87421....87427....87433....87443
....87473....87481....87491....87509....87511....87517....87523....87539
....87541....87547....87553....87557....87559....87583....87587....87589
....87613....87623....87629....87631....87641....87643....87649....87671
....87679....87683....87691....87697....87701....87719....87721....87739
....87743....87751....87767....87793....87797....87803....87811....87833
....87853....87869....87877....87881....87887....87911....87917....87931
....87943....87959....87961....87973....87977....87991....88001....88003
....88007....88019....88037....88069....88079....88093....88117....88129
....88169....88177....88211....88223....88237....88241....88259....88261
....88289....88301....88321....88327....88337....88339....88379....88397
....88411....88423....88427....88463....88469....88471....88493....88499
....88513....88523....88547....88589....88591....88607....88609....88643
....88651....88657....88661....88663....88667....88681....88721....88729
....88741....88747....88771....88789....88793....88799....88801....88807
....88811....88813....88817....88819....88843....88853....88861....88867
....88873....88883....88897....88903....88919....88937....88951....88969
....88993....88997....89003....89009....89017....89021....89041....89051
....89057....89069....89071....89083....89087....89101....89107....89113
....89119....89123....89137....89153....89189....89203....89209....89213
....89227....89231....89237....89261....89269....89273....89293....89303
....89317....89329....89363....89371....89381....89387....89393....89399
....89413....89417....89431....89443....89449....89459....89477....89491
....89501....89513....89519....89521....89527....89533....89561....89563
....89567....89591....89597....89599....89603....89611....89627....89633
....89653....89657....89659....89669....89671....89681....89689....89753

....89759....89767....89779....89783....89797....89809....**89819**....**89821**
....89833....89839....89849....89867....89891....**89897**....**89899**....89909
....89917....89923....89939....89959....89963....89977....89983....89989
....90001....90007....90011....**90017**....**90019**....90023....90031....90053
....90059....90067....**90071**....**90073**....90089....90107....90121....90127
....90149....90163....90173....90187....90191....**90197**....**90199**....90203
....90217....90227....90239....90247....90263....90271....90281....90289
....90313....90353....90359....**90371**....**90373**....90379....90397....**90401**
....**90403**....90407....**90437**....**90439**....90469....90473....90481....90499
....90511....90523....**90527**....**90529**....90533....90547....90583....90599
....**90617**....**90619**....90631....90641....90647....90659....**90677**....**90679**
....90697....90703....90709....90731....90749....90787....90793....90803
....**90821**....**90823**....90833....90841....90847....90863....90887....90901
....90907....90911....90917....90931....90947....90971....90977....90989
....90997....91009....91019....91033....**91079**....**91081**....**91097**....**91099**
....91121....**91127**....**91129**....**91139**....**91141**....**91151**....**91153**....91159
....91163....91183....91193....91199....91229....91237....91243....91249
....91253....91283....91291....91297....91303....91309....91331....**91367**
....**91369**....91373....91381....91387....91393....91397....91411....91423
....91433....91453....**91457**....**91459**....91463....91493....91499....91513
....91529....91541....**91571**....**91573**....91577....91583....91591....91621
....91631....91639....91673....91691....91703....91711....91733....91753
....91757....91771....91781....91801....91807....**91811**....**91813**....91823
....91837....91841....91867....91873....91909....91921....91939....91943
....91951....91957....91961....**91967**....**91969**....91997....92003....92009
....92033....92041....92051....92077....92083....92107....92111....92119
....92143....92153....92173....**92177**....**92179**....92189....92203....**92219**
....**92221**....92227....92233....92237....92243....92251....92269....92297
....92311....92317....92333....92347....92353....92357....92363....92369
....92377....**92381**....**92383**....92387....**92399**....**92401**....92413....92419
....92431....**92459**....**92461**....92467....92479....92489....92503....92507
....92551....92557....**92567**....**92569**....92581....92593....92623....92627
....**92639**....**92641**....92647....92657....**92669**....**92671**....**92681**....**92683**
....92693....92699....92707....92717....92723....92737....92753....92761
....92767....92779....**92789**....**92791**....92801....92809....92821....92831
....92849....92857....**92861**....**92863**....92867....92893....92899....92921
....92927....92941....92951....**92957**....**92959**....92987....92993....93001
....93047....93053....93059....93077....93083....93089....93097....93103
....93113....**93131**....**93133**....93139....93151....93169....93179....93187
....93199....93229....**93239**....**93241**....**93251**....**93253**....93257....93263
....**93281**....**93283**....93287....93307....93319....93323....93329....93337
....93371....93377....93383....93407....93419....93427....93463....**93479**
....**93481**....93487....**93491**....**93493**....93497....93503....93523....93529
....93553....**93557**....**93559**....93563....93581....93601....93607....93629
....93637....93683....**93701**....**93703**....93719....93739....**93761**....**93763**
....93787....**93809**....**93811**....93827....93851....93871....**93887**....**93889**
....93893....93901....**93911**....**93913**....93923....93937....93941....93949
....93967....93971....93979....93983....93997....**94007**....**94009**....94033
....94049....94057....94063....94079....94099....**94109**....**94111**....94117
....94121....**94151**....**94153**....94169....94201....94207....94219....94229
....94253....94261....94273....94291....**94307**....**94309**....94321....94327
....94331....94343....**94349**....**94351**....94379....**94397**....**94399**....94421
....94427....94433....**94439**....**94441**....94447....94463....94477....94483
....94513....**94529**....**94531**....**94541**....**94543**....94547....**94559**....**94561**
....94573....94583....94597....94603....94613....94621....**94649**....**94651**
....94687....94693....94709....94723....94727....94747....94771....94777
....94781....94789....94793....94811....94819....94823....94837....94841
....**94847**....**94849**....94873....94889....94903....94907....94933....**94949**
....**94951**....94961....94993....94999....95003....95009....95021....95027
....95063....95071....95083....**95087**....**95089**....95093....95101....95107
....95111....95131....95143....95153....95177....**95189**....**95191**....95203
....95213....95219....**95231**....**95233**....95239....95257....95261....95267
....95273....95279....95287....95311....95317....95327....95339....95369
....95383....95393....95401....95413....95419....95429....**95441**....**95443**

Anlage 1

....95461....95467....95471....95479....95483....95507....95527....95531
....95539....95549....95561....95569....95581....95597....95603....95617
....95621....95629....95633....95651....95701....95707....95713....95717
....95723....95731....95737....95747....95773....95783....**95789**....**95791**
....**95801**....**95803**....95813....95819....95857....95869....95873....95881
....95891....95911....95917....95923....95929....95947....**95957**....**95959**
....95971....**95987**....**95989**....96001....96013....96017....96043....96053
....96059....96079....96097....96137....96149....96157....96167....**96179**
....**96181**....96199....96211....**96221**....**96223**....96233....96259....96263
....96269....96281....96289....96293....96323....**96329**....**96331**....96337
....96353....96377....96401....96419....96431....96443....96451....96457
....96461....96469....96479....96487....96493....96497....96517....96527
....96553....96557....96581....**96587**....**96589**....96601....96643....96661
....96667....96671....96697....96703....96731....**96737**....**96739**....96749
....96757....96763....96769....96779....96787....**96797**....**96799**....**96821**
....**96823**....96827....96847....96851....96857....96893....96907....96911
....96931....96953....96959....96973....96979....96989....96997....**97001**
....**97003**....97007....97021....97039....97073....97081....97103....97117
....97127....97151....**97157**....**97159**....**97169**....**97171**....97177....97187
....97213....97231....97241....97259....97283....**97301**....**97303**....97327
....**97367**....**97369**....97373....**97379**....**97381**....97387....97397....97423
....97429....97441....97453....97459....97463....**97499**....**97501**....97511
....97523....**97547**....**97549**....97553....97561....97571....**97577**....**97579**
....97583....**97607**....**97609**....97613....**97649**....**97651**....97673....97687
....97711....97729....97771....97777....**97787**....**97789**....97813....97829
....**97841**....**97843**....**97847**....**97849**....**97859**....**97861**....97871....97879
....97883....97919....97927....97931....97943....97961....97967....97973
....97987....**98009**....**98011**....98017....98041....98047....98057....98081
....98101....98123....98129....98143....98179....98207....98213....98221
....98227....98251....98257....98269....**98297**....**98299**....98317....**98321**
....**98323**....98327....98347....98369....98377....**98387**....**98389**....98407
....98411....98419....98429....98443....98453....98459....98467....98473
....98479....98491....98507....98519....98533....98543....**98561**....**98563**
....98573....98597....98621....98627....**98639**....**98641**....98663....98669
....98689....**98711**....**98713**....98717....**98729**....**98731**....98737....98773
....98779....98801....**98807**....**98809**....98837....98849....**98867**....**98869**
....98873....98887....98893....**98897**....**98899**....**98909**....**98911**....**98927**
....**98929**....98939....98947....98953....98963....98981....98993....98999
....99013....99017....99023....99041....99053....99079....99083....99089
....99103....99109....99119....**99131**....**99133**....**99137**....**99139**....99149
....99173....99181....99191....99223....99233....99241....99251....**99257**
....**99259**....99277....99289....99317....**99347**....**99349**....99367....99371
....99377....99391....99397....99401....99409....99431....99439....99469
....99487....99497....99523....**99527**....**99529**....99551....99559....99563
....99571....99577....99581....99607....99611....99623....99643....99661
....99667....99679....99689....**99707**....**99709**....99713....**99719**....**99721**
....99733....99761....99767....99787....99793....99809....99817....99823
....99829....99833....99839....99859....99871....99877....99881....99901
....99907....99923....99929....99961....99971....**99989**....**99991**

Von 3 bis 100000 sind 9591 Primzahlen zu finden sowie 1224 Zwillinge mod 2

Anmerkungen:

1. Die Ordnungsfolge der ungeraden Primzahlen von 3 bis 99991 erweist sich für das Verständnis des in dieser Monographie deduzierten Beweises der Goldbach'schen Vermutung als wichtige Basis, auf die der Leser zurückgreifen kann. Die hier deduzierten - in der Primzahlforschung noch unbekannten Gesetze - finden bei Betrachtung dieser Anlage eine volle Bestätigung und werden in Anlage 20 vertieft und bestätig

2. Die Primzahlzwillinge mod 2 wurden in Anlage 1 schattiert markiert. Daraus ist ersichtlich, dass sie auch in einer verketteten Sequenz vorkommen können, ohne dass zwischen ihnen (unbeteiligte) Primzahlen auftreten. Dieser Aspekt wurde in der Primzahlforschung bisher nicht näher analysiert. Beachtlich ist des weiteren, dass in diesen Zwillingen stets nur Primzahlen mit den Endziffern: 1&3, 1&9, 7&9 eingehen können und dass eine dieser Primzahlen zu den Klassen: 1,2,3,4 - die andere aber - zu den Klassen: 5,6,7,8 gehören muss.

Anlage 2

Die Zwillinge mod 2, mod 4, mod 6, mod 8, mod 10 und mod 100 im Intervall: 0 - 10000:

Primzahlzwillinge (ohne die Primzahl 2) im Intervall von 2 bis 10000

(3,
5)..+0...(5,7)..+4...(11,13)..+4...(17,19)..+10...(29,31)..+10...(41,43)..+
16...(59,
61)..+10...(71,73)..+28...(101,103)..+4...(107,109)..+28...(137,139)..+10..
.(
149,151)..+28...(179,181)..+10...(191,193)..+4...(197,199)..+28...(227,229)
..+10...(239,
241)..+28...(269,271)..+10...(281,283)..+28...(311,313)..+34...(347,349)..+
70...(419,
421)..+10...(431,433)..+28...(461,463)..+58...(521,523)..+46...(569,571)..+
28...(599,
601)..+16...(617,619)..+22...(641,643)..+16...(659,661)..+148...(809,811)..
+10...(
821,823)..+4...(827,829)..+28...(857,859)..+22...(881,883)..+136...(1019,10
21)..+10...(1031,1033)..+16...(1049,1051)..+10...(1061,1063)..+28...(1091,1
093)..+58...(1151,1153)..+76...(1229,1231)..+46...(1277,1279)..+10...(1289,
1291)..+10...(
1301,1303)..+16...(1319,1321)..+106...(1427,1429)..+22...(1451,1453)..+28..
.(
1481,1483)..+4...(1487,1489)..+118...(1607,1609)..+10...(1619,1621)..+46...
(1667,
1669)..+28...(1697,1699)..+22...(1721,1723)..+64...(1787,1789)..+82...(1871
,
1873)..+4...(1877,1879)..+52...(1931,1933)..+16...(1949,1951)..+46...(1997,
1999)..+28...(2027,
2029)..+52...(2081,2083)..+4...(2087,2089)..+22...(2111,2113)..+16...(2129,
2131)..+10...(
2141,2143)..+94...(2237,2239)..+28...(2267,2269)..+40...(2309,2311)..+28...
(2339,
2341)..+40...(2381,2383)..+166...(2549,2551)..+40...(2591,2593)..+64...(265
7,2659)..+28...(2687,2689)..+22...(2711,2713)..+16...(2729,2731)..+58...(27
89,2791)..+10...(2801,2803)..+166...(2969,2971)..+28...(2999,3001)..+118...
(3119,
3121)..+46...(3167,3169)..+82...(3251,3253)..+4...(3257,3259)..+40...(3299,
3301)..+28...(3329,3331)..+28...(3359,3361)..+10...(3371,3373)..+16...(3389
,3391)..+70...(
3461,3463)..+4...(3467,3469)..+58...(3527,3529)..+10...(3539,3541)..+16...(
3557,3559)..+22...(3581,3583)..+88...(3671,3673)..+94...(3767,3769)..+52...
(
3821,3823)..+28...(3851,3853)..+64...(3917,3919)..+10...(3929,3931)..+70...
(
4001,4003)..+16...(4019,4021)..+28...(4049,4051)..+40...(4091,4093)..+34...
(4127,
4129)..+28...(4157,4159)..+58...(4217,4219)..+10...(4229,4231)..+10...(4241
4243)..+16...(4259,
4261)..+10...(4271,4273)..+64...(4337,4339)..+82...(4421,4423)..+58...(4481
4483)..+34...(
4517,4519)..+28...(4547,4549)..+88...(4637,4639)..+10...(4649,4651)..+70...

721,4723)..+64...(4787,4789)..+10...(4799,4801)..+130...(4931,4933)..+34..
(4967,

4969)..+40...(**5009**,5011)..+10...(**5021**,5023)..+76...(5099,**5101**)..+130...(523
1,
5233)..+46...(5279,**5281**)..+136...(**5417**,5419)..+22...(**5441**,5443)..+34...(**547
7**,5479)..+22...(**5501**,5503)..+16...(5519,**5521**)..+118...(5639,**5641**)..+10...(5
651,
5653)..+4...(**5657**,5659)..+82...(**5741**,5743)..+106...(**5849**,5851)..+16...(5867
,
5869)..+10...(5879,**5881**)..+208...(**6089**,6091)..+40...(6131,**6133**)..+64...(**619
7**,6199)..+70...(**6269**,6271)..+28...(6299,**6301**)..+58...(6359,**6361**)..+88...(**64
49**,6451)..+100...(6551,**6553**)..+16...(**6569**,6571)..+88...(6659,**6661**)..+28...(
6689,6691)..+10...(**6701**,6703)..+58...(**6761**,6763)..+16...(6779,**6781**)..+10...
(6791,
6793)..+34...(6827,**6829**)..+40...(**6869**,6871)..+76...(6947,**6949**)..+10...(6959
,
6961)..+166...(7127,**7129**)..+82...(7211,**7213**)..+94...(7307,**7309**)..+22...(733
1,
7333)..+16...(**7349**,7351)..+106...(**7457**,7459)..+28...(7487,**7489**)..+58...(754
7,
7549)..+10...(7559,**7561**)..+28...(**7589**,7591)..+166...(**7757**,7759)..+118...(**78
77**,7879)..+70...(**7949**,7951)..+58...(**8009**,8011)..+76...(8087,**8089**)..+130...(
8219,
8221)..+10...(8231,**8233**)..+58...(8291,**8293**)..+94...(8387,**8389**)..+40...(**8429**
,8431)..+106...(
8537,8539)..+58...(**8597**,8599)..+28...(8627,**8629**)..+190...(8819,**8821**)..+16..
.(
8837,8839)..+22...(**8861**,8863)..+106...(**8969**,8971)..+28...(8999,**9001**)..+10..
.(9011,
9013)..+28...(**9041**,9043)..+196...(9239,**9241**)..+40...(**9281**,9283)..+58...(**934
1**,9343)..+76...(9419,**9421**)..+10...(9431,**9433**)..+4...(**9437**,9439)..+22...(**946
1**,9463)..+166...(**9629**,9631)..+46...(**9677**,9679)..+40...(9719,**9721**)..+46...(9
767,**9769**)..+88...(**9857**,9859)..+70...(**9929**,9931)

Insgesamt sind es 205 Zwillinge

Semizwillinge im Abstand 4 (ohne die Primzahl 2) im Intervall von 2 bis 10000

(3,7)..+0...(7,11)..+2...(
13,17)..+2...(19,23)..+14...(37,41)..+2...(43,47)..+20...(67,71)..+8...(79,
83)..+14...(
97,101)..+2...(103,107)..+2...(109,113)..+14...(127,131)..+32...(163,167)..
+26...(
193,197)..+26...(223,227)..+2...(229,233)..+44...(277,281)..+26...(307,311)
..+2...(
313,317)..+32...(349,353)..+26...(379,383)..+14...(397,401)..+38...(439,443
)..+14...(
457,461)..+2...(463,467)..+20...(487,491)..+8...(499,503)..+110...(613,617)
..+26...(643,647)..+26...(
673,677)..+62...(739,743)..+14...(757,761)..+8...(769,773)..+50...(823,827)
..+26...(
853,857)..+2...(859,863)..+14...(877,881)..+2...(883,887)..+20...(907,911).
.+26...(
937,941)..+26...(967,971)..+38...(1009,1013)..+74...(1087,1091)..+2...(1093
,
1097)..+116...(1213,1217)..+62...(1279,1283)..+14...(1297,1301)..+2...(1303
,1307)..+116...(1423,1427)..+2...(
1429,1433)..+14...(1447,1451)..+32...(1483,1487)..+2...(1489,1493)..+56...(
1549,1553)..+14...(1567,1571)..+8...(1579,1583)..+14...(1597,1601)..+8...(1
609,1613)..+50...(1663,1667)..+26...(1693,1697)..+86...(1783,1787)..+80...(
1867,1871)..+2...(
1873,1877)..+116...(1993,1997)..+2...(1999,2003)..+80...(2083,2087)..+50...
(
2137,2141)..+62...(2203,2207)..+32...(2239,2243)..+26...(2269,2273)..+20...
(
2293,2297)..+50...(2347,2351)..+26...(2377,2381)..+8...(2389,2393)..+44...(
2437,2441)..+32...(2473,2477)..+62...(2539,2543)..+74...(2617,2621)..+38...
(2659,2663)..+20...(2683,2687)..+2...(
2689,2693)..+14...(2707,2711)..+38...(2749,2753)..+44...(2797,2801)..+32...
(
2833,2837)..+20...(2857,2861)..+92...(2953,2957)..+62...(3019,3023)..+14...
(
3037,3041)..+38...(3079,3083)..+80...(3163,3167)..+20...(3187,3191)..+26...
(
3217,3221)..+32...(3253,3257)..+62...(3319,3323)..+20...(3343,3347)..+110..
.(
3457,3461)..+2...(3463,3467)..+62...(3529,3533)..+80...(3613,3617)..+56...(
3673,3677)..+20...(3697,3701)..+92...(3793,3797)..+50...(3847,3851)..+26...
(
3877,3881)..+26...(3907,3911)..+8...(3919,3923)..+20...(3943,3947)..+56...(
4003,4007)..+122...(
4129,4133)..+20...(4153,4157)..+290...(4447,4451)..+62...(4513,4517)..+2...
(4519,4523)..+116...(4639,4643)..+86...(
4729,4733)..+50...(4783,4787)..+2...(4789,4793)..+20...(4813,4817)..+116...
(
4933,4937)..+32...(4969,4973)..+26...(4999,5003)..+74...(5077,5081)..+86...
(5167,5171)..+56...(5227,5231)..+2...(
5233,5237)..+110...(5347,5351)..+62...(5413,5417)..+20...(5437,5441)..+38..
.(5479,5483)..+20...(5503,5507)..+20...(5527,5531)..+38...(
5569,5573)..+74...(5647,5651)..+2...(5653,5657)..+32...(5689,5693)..+44...(
5737,5741)..+38...(5779,5783)..+56...(5839,5843)..+14...(5857,5861)..+62...
(5923,5927)..+80...(6007,6011)..+32...(6043,6047)..+152...(6199,6203)..+14.
.(
6217,6221)..+248...(6469,6473)..+74...(6547,6551)..+26...(6577,6581)..+152.
.(
6733,6737)..+86...(6823,6827)..+2...(6829,6833)..+74...(6907,6911)..+56...(
6967,6971)..+26...(
6997,7001)..+38...(7039,7043)..+164...(7207,7211)..+32...(7243,7247)..+230.

..(
7477,7481)..+56...(**7537,7541**)..+32...(**7573,7577**)..+26...(7603,7607)..+32...
(7639,7643)..+26...(
7669,7673)..+14...(7687,7691)..+8...(7699,7703)..+20...(7723,7727)..+26...(
7753,7757)..+32...(**7789,7793**)..+80...(**7873,7877**)..+2...(7879,7883)..+50...(
7933,7937)..+152...(**8089,8093**)..+74...(8167,8171)..+62...(**8233,8237**)..+32..
.(
8269,8273)..+14...(8287,8291)..+2...(**8293,8297**)..+122...(8419,8423)..+20...
(8443,8447)..+92...(8539,8543)..+80...(8623,8627)..+50...(
8677,8681)..+8...(**8689,8693**)..+44...(**8737,8741**)..+38...(8779,8783)..+20...(
8803,8807)..+56...(8863,8867)..+62...(
8929,8933)..+74...(9007,9011)..+122...(**9133,9137**)..+20...(**9157,9161**)..+38..
.(9199,9203)..+74...(
9277,9281)..+38...(9319,9323)..+14...(**9337,9341**)..+92...(**9433,9437**)..+26...
(9463,9467)..+80...(9547,9551)..+68...(9619,9623)..+116...(9739,9743)..+44.
..(9787,9791)..+38...(**9829,9833**)..+50...(9883,9887)

Insgesamt sind es 203 Semizwillinge im Abstand 4

Semizwillinge im Abstand 6 (ohne die Primzahl 2) im Intervall von 2 bis 10000

(
5,11)..-4...(7,**13**)..-2...(11,**17**)..-4...(**13**,19)..-2...(**17**,23)..+0...(23,**29**)..
.+2...(31,
37)..+0...(**37**,43)..-2...(**41**,47)..+0...(47,**53**)..+0...(**53**,59)..+2...(**61**,67)...
+0...(67,
73)..+0...(**73**,79)..+4...(83,**89**)..+8...(**97**,103)..-2...(**101**,107)..-4...(103,1
09)..-2...(107,**113**)..+18...(131,**137**)..+14...(151,**157**)..+0...(**157**,163)..+4..
.(167,
173)..+0...(**173**,179)..+12...(191,**197**)..-4...(**193**,199)..+24...(223,**229**)..-2.
..(227,
233)..+0...(**233**,239)..+12...(251,**257**)..+0...(**257**,263)..+0...(263,**269**)..+2..
.(271,
277)..+0...(**277**,283)..+24...(307,**313**)..-2...(311,**317**)..+14...(331,**337**)..+10
...(347,
353)..+0...(**353**,359)..+8...(367,**373**)..+0...(**373**,379)..+4...(383,**389**)..+44..
.(
433,439)..+4...(443,**449**)..+8...(**457**,463)..-2...(**461**,467)..+36...(503,**509**)..
+32...(
541,547)..+10...(**557**,563)..+0...(563,**569**)..+2...(571,**577**)..+10...(587,**593**).
.+0...(
593,599)..+2...(**601**,607)..+0...(607,**613**)..+0...(**613**,619)..+22...(**641**,647)..
+0...(647,
653)..+0...(**653**,659)..+18...(**677**,683)..+44...(727,**733**)..+0...(**733**,739)..+12
...(751,
757)..+64...(**821**,827)..-4...(823,**829**)..+24...(**853**,859)..-2...(**857**,863)..+14
...(
877,883)..-2...(**881**,887)..+54...(**941**,947)..+0...(947,**953**)..+18...(971,**977**).
.+0...(
977,983)..+8...(991,**997**)..+16...(**1013**,1019)..+14...(**1033**,1039)..+24...(1063
,
1069)..+18...(1087,**1093**)..-2...(1091,**1097**)..+0...(**1097**,1103)..+0...(1103,**11**
09)..+8...(**1117**,1123)..+0...(1123,**1129**)..+52...(**1181**,1187)..+0...(1187,**1193**
)..+24...(
1217,1223)..+0...(1223,**1229**)..+2...(1231,**1237**)..+40...(**1277**,1283)..+0...(12
83,
1289)..+2...(1291,**1297**)..+0...(**1297**,1303)..-2...(**1301**,1307)..+14...(**1321**,13
27)..+34...(
1361,1367)..+0...(1367,**1373**)..+50...(1423,**1429**)..-2...(1427,**1433**)..+0...(**14**
33,1439)..+8...(1447,**1453**)..+0...(**1453**,1459)..+22...(**1481**,1487)..-4...(1483
,
1489)..-2...(1487,**1493**)..+0...(**1493**,1499)..+44...(1543,**1549**)..+4...(**1553**,15
59)..+42...(
1601,1607)..+0...(1607,**1613**)..+0...(**1613**,1619)..+2...(**1621**,1627)..+30...(**16**
57,1663)..+0...(1663,**1669**)..+24...(**1693**,1699)..+42...(**1741**,1747)..+0...(174
7,
1753)..+0...(**1753**,1759)..+18...(**1777**,1783)..+0...(1783,**1789**)..+72...(**1861**,1
867)..+0...(1867,
1873)..-2...(1871,**1877**)..-4...(**1873**,1879)..+22...(**1901**,1907)..+0...(1907,**19**
13)..+60...(**1973**,1979)..+8...(1987,**1993**)..+0...(**1993**,1999)..-2...(**1997**,2003
)..+8...(2011,
2017)..+46...(2063,**2069**)..+12...(**2081**,2087)..-4...(2083,**2089**)..+42...(2131,
2137)..+0...(**2137**,2143)..+64...(2207,**2213**)..+24...(**2237**,2243)..+24...(2267,
2273)..+8...(**2281**,2287)..+0...(2287,**2293**)..+40...(**2333**,2339)..+2...(**2341**,23
47)..+4...(2351,
2357)..+14...(2371,**2377**)..+0...(**2377**,2383)..+0...(2383,**2389**)..+4...(**2393**,23
99)..+12...(2411,
2417)..+0...(**2417**,2423)..+18...(**2441**,2447)..+20...(2467,**2473**)..+70...(2543,
2549)..+2...(2551,**2557**)..+100...(**2657**,2663)..+8...(2671,**2677**)..+0...(**2677**,2
683)..+0...(2683,

2689)..-2...(2687,**2693**)..+0...(**2693**,2699)..+8...(2707,**2713**)..+0...(**2713**,271
9)..+72...(2791,
2797)..+0...(**2797**,2803)..+34...(**2837**,2843)..+8...(2851,**2857**)..+40...(**2897**,2
903)..+0...(2903,
2909)..+48...(**2957**,2963)..+0...(2963,**2969**)..+92...(**3061**,3067)..+16...(3083,
3089)..+74...(3163,**3169**)..+12...(**3181**,3187)..+16...(3203,**3209**)..+42...(3251
,
3257)..-4...(**3253**,3259)..+42...(**3301**,3307)..+0...(3307,**3313**)..+0...(**3313**,33
19)..+4...(3323,
3329)..+78...(3407,**3413**)..+44...(**3457**,3463)..-2...(**3461**,3467)..-4...(3463,**3
469**)..+42...(3511,**3517**)..+10...(3527,**3533**)..+0...(**3533**,3539)..+2...(**3541**,35
47)..+60...(3607,
3613)..+4...(**3617**,3623)..+8...(3631,**3637**)..+0...(**3637**,3643)..+28...(3671,**36
77**)..+14...(3691,**3697**)..+30...(3727,**3733**)..+0...(**3733**,3739)..+22...(**3761**,37
67)..+30...(
3797,3803)..+44...(3847,**3853**)..+58...(3911,**3917**)..+0...(**3917**,3923)..+0...(3
923,
3929)..+72...(**4001**,4007)..+0...(4007,**4013**)..+0...(**4013**,4019)..+2...(**4021**,40
27)..+24...(4051,
4057)..+16...(**4073**,4079)..+14...(**4093**,4099)..+28...(4127,**4133**)..+0...(**4133**,
4139)..+14...(
4153,4159)..+52...(4211,**4217**)..+36...(**4253**,4259)..+24...(4283,**4289**)..+68...
(
4357,4363)..+28...(4391,**4397**)..+44...(**4441**,4447)..+4...(4451,**4457**)..+0...(**4
457**,4463)..+44...(4507,**4513**)..+0...(**4513**,4519)..-2...(**4517**,4523)..+38...(**45
61**,4567)..+24...(4591,**4597**)..+0...(**4597**,4603)..+34...(**4637**,4643)..+0...(464
3,
4649)..+2...(4651,**4657**)..+0...(**4657**,4663)..+10...(**4673**,4679)..+44...(4723,**4
729**)..+54...(4783,**4789**)..-2...(4787,**4793**)..+0...(**4793**,4799)..+72...(4871,**48
77**)..+26...(4903,**4909**)..+22...(4931,**4937**)..+0...(**4937**,4943)..+8...(4951,**495
7**)..+10...(4967,**4973**)..+14...(4987,**4993**)..+0...(**4993**,4999)..+4...(5003,**5009**
)..+72...(
5081,5087)..+14...(**5101**,5107)..+0...(5107,**5113**)..+0...(**5113**,5119)..+28...(5
147,
5153)..+74...(5227,**5233**)..-2...(5231,**5237**)..+36...(**5273**,5279)..+18...(**5297**,
5303)..+0...(5303,
5309)..+72...(**5381**,5387)..+0...(5387,**5393**)..+0...(**5393**,5399)..+8...(5407,**54
13**)..+0...(**5413**,5419)..+12...(5431,**5437**)..+0...(**5437**,5443)..+0...(5443,**5449**
)..+22...(5471,
5477)..+0...(**5477**,5483)..+18...(**5501**,5507)..+14...(**5521**,5527)..+30...(**5557**,
5563)..+0...(5563,
5569)..+72...(**5641**,5647)..+0...(5647,**5653**)..-2...(5651,**5657**)..-4...(**5653**,56
59)..+24...(5683,
5689)..+22...(5711,**5717**)..+20...(**5737**,5743)..+0...(5743,**5749**)..+52...(**5801**,
5807)..+0...(5807,
5813)..+8...(**5821**,5827)..+16...(5843,**5849**)..+2...(5851,**5857**)..+4...(**5861**,58
67)..+30...(
5897,5903)..+78...(**5981**,5987)..+50...(**6037**,6043)..+4...(6047,**6053**)..+14...(
6067,
6073)..+0...(**6073**,6079)..+118...(**6197**,6203)..+8...(6211,**6217**)..+40...(**6257**,
6263)..+0...(6263,
6269)..+2...(6271,**6277**)..+34...(6311,**6317**)..+0...(**6317**,6323)..+0...(6323,**63
29**)..+8...(**6337**,6343)..+10...(**6353**,6359)..+2...(**6361**,6367)..+0...(6367,**6373**
..+0...(
6373,6379)..+42...(**6421**,6427)..+120...(6547,**6553**)..+10...(6563,**6569**)..+2...
(6571,
6577)..+76...(**6653**,6659)..+14...(**6673**,6679)..+24...(6703,**6709**)..+114...(682
3,
6829)..-2...(6827,**6833**)..+24...(**6857**,6863)..+0...(6863,**6869**)..+42...(6911,**6
917**)..+44...(**6961**,6967)..+4...(6971,**6977**)..+0...(**6977**,6983)..+8...(6991,**699
7**)..+16...(**7013**,7019)..+84...(7103,**7109**)..+12...(**7121**,7127)..+60...(7187,**71
93**)..+14...(7207,**7213**)..+0...(**7213**,7219)..+18...(**7237**,7243)..+4...(7247,**725
3**)..+158...(7411,**7417**)..+34...(7451,**7457**)..+24...(**7481**,7487)..+30...(**7517**,7

523)..+0...(7523,
7529)..+12...(**7541**,7547)..+30...(**7577**,7583)..+0...(7583,**7589**)..+54...(7643,
7649)..+32...(**7681**,7687)..+30...(**7717**,7723)..+30...(**7753**,7759)..+58...(**7817**
,7823)..+0...(7823,
7829)..+38...(7867,**7873**)..+0...(**7873**,7879)..-2...(**7877**,7883)..+18...(**7901**,7
907)..+20...(7927,
7933)..+78...(8011,**8017**)..+36...(**8053**,8059)..+22...(**8081**,8087)..+0...(8087,
8093)..+18...(8111,**8117**)..+0...(**8117**,8123)..+38...(**8161**,8167)..+64...(8231,
8237)..+0...(**8237**,8243)..+20...(8263,**8269**)..+18...(8287,**8293**)..-2...(8291,**8**
297)..+14...(8311,**8317**)..+46...(8363,**8369**)..+54...(8423,**8429**)..+32...(**8461**,
8467)..+54...(
8521,8527)..+10...(**8537**,8543)..+80...(8623,**8629**)..+12...(**8641**,8647)..+16...
(8663,
8669)..+24...(**8693**,8699)..+8...(8707,**8713**)..+0...(**8713**,8719)..+12...(8731,**8**
737)..+4...(**8741**,8747)..+0...(8747,**8753**)..+78...(8831,**8837**)..+24...(**8861**,88
67)..+20...(8887,
8893)..+30...(8923,**8929**)..+34...(8963,**8969**)..+32...(**9001**,9007)..+0...(9007,
9013)..+30...(9043,**9049**)..+54...(9103,**9109**)..+18...(9127,**9133**)..+18...(9151
,
9157)..+24...(**9181**,9187)..+16...(9203,**9209**)..+12...(**9221**,9227)..+50...(**9277**
,9283)..+54...(
9337,9343)..+0...(9343,**9349**)..+22...(9371,**9377**)..+14...(9391,**9397**)..+0...(**9**
397,9403)..+10...(**9413**,9419)..+12...(9431,**9437**)..-4...(**9433**,9439)..+22...(**9**
461,9467)..+0...(9467,**9473**)..+0...(**9473**,9479)..+12...(9491,**9497**)..+36...(**95**
33,9539)..+74...(**9613**,9619)..+4...(9623,**9629**)..+14...(9643,**9649**)..+84...(**97**
33,9739)..+4...(9743,**9749**)..+32...(**9781**,9787)..+24...(9811,**9817**)..+16...(**98**
33,9839)..+12...(9851,**9857**)..+44...(**9901**,9907)..+16...(9923,**9929**)..+38...(9
967,**9973**)

Insgesamt sind es 411 Semizwillinge im Abstand 6

Semizwillinge im Abstand 8 (ohne die Primzahl 2) im Intervall von 2 bis 10000

(3,11)..-6...(
5,13)..-2...(11,19)..+4...(23,31)..-2...(**29,37**)..+16...(**53,61**)..-2...(59,67
)..+4...(71,79)..+10...(
89,97)..+4...(**101,109**)..+22...(131,139)..+10...(**149,157**)..+16...(**173,181**)..
+10...(191,199)..+34...(
233,241)..+22...(263,271)..-2...(**269,277**)..+82...(359,367)..+22...(**389,397**)
..+4...(
401,409)..+22...(431,439)..+10...(**449,457**)..+22...(479,487)..+4...(491,499)
..+64...(563,571)..-2...(
569,577)..+16...(**593,601**)..-2...(599,607)..+46...(**653,661**)..+22...(683,691)
..+10...(
701,709)..+10...(719,727)..+16...(743,751)..+10...(**761,769**)..+52...(**821,829**
)..+82...(911,919)..+10...(
929,937)..+46...(983,991)..+22...(**1013,1021**)..+10...(1031,1039)..+22...(**106**
1,1069)..+40...(**1109,1117**)..+46...(1163,1171)..+22...(**1193,1201**)..+22...(12
23,1231)..-2...(
1229,1237)..+46...(1283,1291)..-2...(**1289,1297**)..+22...(1319,1327)..+46...(
1373,1381)..+58...(1439,1447)..+4...(1451,1459)..+22...(**1481,1489**)..+34...(
1523,1531)..+28...(1559,1567)..+4...(1571,1579)..+22...(
1601,1609)..+4...(**1613,1621**)..-2...(1619,1627)..+106...(**1733,1741**)..+82...(
1823,1831)..+40...(1871,1879)..+100...(1979,1987)..+16...(2003,2011)..+70..
.(
2081,2089)..+40...(**2129,2137**)..+16...(**2153,2161**)..+52...(**2213,2221**)..+22...
(2243,2251)..+22...(
2273,2281)..+52...(**2333,2341**)..-2...(2339,2347)..+34...(**2381,2389**)..+70...(
2459,2467)..+64...(2531,2539)..+4...(2543,2551)..-2...(
2549,2557)..+52...(**2609,2617**)..+46...(2663,2671)..+28...(2699,2707)..+4...(
2711,2719)..+22...(
2741,2749)..+40...(**2789,2797**)..+46...(2843,2851)..+28...(2879,2887)..+22...
(
2909,2917)..+46...(2963,2971)..+40...(3011,3019)..+22...(**3041,3049**)..+160..
.(
3209,3217)..+4...(**3221,3229**)..+22...(3251,3259)..+40...(3299,3307)..+16...(
3323,3331)..+118...(
3449,3457)..+4...(**3461,3469**)..+22...(3491,3499)..+34...(**3533,3541**)..-2...(3
539,3547)..+76...(3623,3631)..+70...(
3701,3709)..+10...(3719,3727)..+34...(**3761,3769**)..+112...(**3881,3889**)..+22..
.(3911,3919)..+4...(3923,3931)..+82...(
4013,4021)..-2...(4019,4027)..+22...(**4049,4057**)..+34...(4091,4099)..+112...
(4211,4219)..+34...(
4253,4261)..+28...(**4289,4297**)..+52...(**4349,4357**)..+226...(4583,4591)..+52..
.(4643,4651)..-2...(
4649,4657)..+64...(**4721,4729**)..+22...(4751,4759)..+34...(**4793,4801**)..+142..
.(4943,4951)..+52...(5003,5011)..+40...(5051,5059)..+40...(5099,5107)..+64.
..(5171,5179)..+10...(
5189,5197)..+76...(**5273,5281**)..+118...(5399,5407)..+34...(**5441,5449**)..+22..
.(5471,5479)..+40...(5519,5527)..+46...(
5573,5581)..+58...(5639,5647)..+4...(5651,5659)..+34...(**5693,5701**)..+40...(
5741,5749)..+34...(5783,5791)..+22...(**5813,5821**)..+22...(5843,5851)..-2...(
5849,5857)..+4...(**5861,5869**)..+160...(**6029,6037**)..+76...(**6113,6121**)..+22...
(6143,6151)..+52...(6203,6211)..+10...(
6221,6229)..+34...(6263,6271)..-2...(**6269,6277**)..+52...(**6329,6337**)..+16...(
6353,6361)..-2...(6359,6367)..+22...(**6389,6397**)..+76...(**6473,6481**)..+40...(
6521,6529)..+34...(6563,6571)..-2...(**6569,6577**)..+22...(6599,6607)..+46...(
6653,6661)..+40...(**6701,6709**)..+124...(**6833,6841**)..+22...(6863,6871)..+28..
.(6899,6907)..+52...(6959,6967)..+16...(6983,6991)..+28...(7019,7027)..+94.
..(
7121,7129)..+22...(7151,7159)..+52...(7211,7219)..+10...(**7229,7237**)..+214..
.(7451,7459)..+22...(
7481,7489)..+10...(7499,7507)..+22...(**7529,7537**)..+4...(**7541,7549**)..+34...(

7583,7591)..+82...(
7673,7681)..+10...(7691,7699)..+220...(7919,7927)..+82...(**8009,8017**)..+64...(
8081,8089)..+4...(**8093,8101**)..+70...(8171,8179)..+190...(**8369,8377**)..+46...(8423,8431)..+82...(
8513,8521)..+52...(**8573,8581**)..+88...(**8669,8677**)..+4...(**8681,8689**)..+10...(8699,8707)..+46...(
8753,8761)..+70...(8831,8839)..+94...(**8933,8941**)..+22...(8963,8971)..+28...(8999,9007)..+34...(
9041,9049)..+10...(9059,9067)..+106...(**9173,9181**)..+130...(9311,9319)..+22...(
9341,9349)..+64...(**9413,9421**)..+10...(9431,9439)..+100...(9539,9547)..+76...(9623,9631)..+58...(
9689,9697)..+106...(9803,9811)..+40...(9851,9859)..+64...(9923,9931)..+10...(**9941,9949**)

Insgesamt sind es 208 Semizwillinge im Abstand 8

Semizwillinge im Abstand 10 (ohne die Primzahl 2) im Intervall von 2 bis 10000

(3,
13)..-6...(7,**17**)..-4...(**13**,23)..-4...(19,**29**)..+2...(31,**41**)..-4...(**37**,47)..-
4...(43,
53)..+8...(**61**,71)..+2...(**73**,83)..-4...(79,**89**)..+8...(**97**,107)..-4...(103,**113**
)..+14...(127,
137)..+2...(139,**149**)..+8...(**157**,167)..-4...(163,**173**)..+8...(**181**,191)..+32..
.(223,
233)..-4...(**229**,239)..+2...(**241**,251)..+20...(271,**281**)..+2...(283,**293**)..+14.
..(307,
317)..+20...(**337**,347)..+2...(**349**,359)..+14...(**373**,383)..-4...(379,**389**)..+20
...(
409,419)..+2...(**421**,431)..+2...(**433**,443)..-4...(439,**449**)..+8...(**457**,467)..+
32...(499,
509)..+38...(547,**557**)..+20...(**577**,587)..+20...(607,**617**)..+14...(631,**641**)..+
2...(643,
653)..+20...(**673**,683)..+8...(691,**701**)..+8...(**709**,719)..+14...(**733**,743)..+8.
..(751,
761)..+26...(787,**797**)..+14...(811,**821**)..+8...(**829**,839)..+14...(**853**,863)..+1
4...(
877,887)..+32...(919,**929**)..+8...(**937**,947)..+20...(967,**977**)..+32...(**1009**,101
9)..+2...(
1021,1031)..+8...(1039,**1049**)..+2...(1051,**1061**)..+26...(1087,**1097**)..-4...(**10**
93,1103)..+50...(**1153**,1163)..+8...(1171,**1181**)..+32...(**1213**,1223)..+26...(**12**
49,1259)..+20...(1279,**1289**)..+2...(1291,**1301**)..-4...(**1297**,1307)..+92...(139
9,
1409)..+14...(1423,**1433**)..-4...(**1429**,1439)..+32...(1471,**1481**)..+2...(1483,**1**
493)..-4...(**1489**,1499)..+44...(1543,**1553**)..-4...(**1549**,1559)..+38...(**1597**,16
07)..+2...(
1609,1619)..+8...(1627,**1637**)..+20...(**1657**,1667)..+32...(1699,**1709**)..+14...(
1723,
1733)..+44...(**1777**,1787)..+14...(**1801**,1811)..+50...(**1861**,1871)..-4...(1867,
1877)..+2...(1879,**1889**)..+98...(1987,**1997**)..-4...(**1993**,2003)..+14...(**2017**,2
027)..+2...(
2029,2039)..+14...(**2053**,2063)..+26...(**2089**,2099)..+32...(2131,**2141**)..+2...(
2143,
2153)..+50...(2203,**2213**)..+74...(2287,**2297**)..+44...(**2341**,2351)..-4...(2347,
2357)..+14...(2371,**2381**)..+2...(2383,**2393**)..-4...(**2389**,2399)..+38...(**2437**,2
447)..+20...(2467,
2477)..+44...(**2521**,2531)..+8...(2539,**2549**)..+98...(2647,**2657**)..+20...(**2677**,
2687)..-4...(2683,
2693)..-4...(**2689**,2699)..+20...(2719,**2729**)..+2...(2731,**2741**)..+26...(2767,**2**
777)..+14...(2791,**2801**)..+32...(**2833**,2843)..+8...(2851,**2861**)..+26...(2887,**2**
897)..+20...(**2917**,2927)..+26...(**2953**,2963)..+38...(**3001**,3011)..+68...(3079,
3089)..+20...(**3109**,3119)..+62...(**3181**,3191)..+122...(**3313**,3323)..-4...(3319
,
3329)..+32...(**3361**,3371)..+86...(**3457**,3467)..+50...(**3517**,3527)..+2...(**3529**,
3539)..+8...(3547,
3557)..+14...(3571,**3581**)..+2...(3583,**3593**)..+14...(3607,**3617**)..-4...(**3613**,3
623)..+68...(3691,
3701)..+8...(**3709**,3719)..+50...(**3769**,3779)..+14...(**3793**,3803)..+20...(3823,
3833)..+20...(**3853**,3863)..+44...(3907,**3917**)..+2...(3919,**3929**)..+74...(4003,
4013)..+116...(**4129**,4139)..+62...(**4201**,4211)..+8...(4219,**4229**)..+2...(4231,
4241)..+2...(4243,**4253**)..+8...(**4261**,4271)..+2...(**4273**,4283)..+44...(4327,**43**
37)..+2...(4339,**4349**)..+14...(4363,**4373**)..+68...(**4441**,4451)..-4...(4447,**445**
7)..+26...(4483,**4493**)..+14...(4507,**4517**)..-4...(**4513**,4523)..+116...(4639,**46**
49)..+14...(4663,**4673**)..+50...(4723,**4733**)..+50...(4783,**4793**)..-4...(**4789**,47
99)..+62...(
4861,4871)..+38...(**4909**,4919)..+14...(**4933**,4943)..+14...(**4957**,4967)..+26...
(

4993,5003)..-4...(4999,**5009**)..+2...(5011,**5021**)..+56...(**5077**,5087)..+92...(5
179,
5189)..+38...(5227,**5237**)..+86...(5323,**5333**)..+74...(5407,**5417**)..+14...(5431
,
5441)..+80...(**5521**,5531)..+32...(5563,**5573**)..+8...(**5581**,5591)..+50...(**5641**,
5651)..-4...(5647,
5657)..+2...(5659,**5669**)..+14...(5683,**5693**)..+8...(**5701**,5711)..+80...(5791,**5
801**)..+38...(5839,**5849**)..+2...(5851,**5861**)..-4...(**5857**,5867)..+2...(**5869**,587
9)..+158...(
6037,6047)..-4...(6043,**6053**)..+26...(6079,**6089**)..+2...(6091,**6101**)..+20...(**6
121**,6131)..+2...(**6133**,6143)..+20...(6163,**6173**)..+38...(6211,**6221**)..+26...(6
247,
6257)..+20...(**6277**,6287)..+14...(**6301**,6311)..+32...(6343,**6353**)..+26...(6379
,
6389)..+92...(**6481**,6491)..+62...(**6553**,6563)..+8...(6571,**6581**)..+98...(6679,
6689)..+2...(6691,**6701**)..+8...(**6709**,6719)..+62...(**6781**,6791)..+2...(**6793**,68
03)..+20...(6823,
6833)..+74...(6907,**6917**)..+32...(**6949**,6959)..+2...(**6961**,6971)..-4...(6967,**6
977**)..+14...(6991,**7001**)..+68...(**7069**,7079)..+98...(**7177**,7187)..+32...(7219,
7229)..+8...(**7237**,7247)..-4...(7243,**7253**)..+44...(**7297**,7307)..+14...(**7321**,7
331)..+146...(
7477,7487)..+2...(**7489**,7499)..+8...(7507,**7517**)..+20...(**7537**,7547)..+2...(**75
49**,7559)..+14...(**7573**,7583)..+56...(7639,**7649**)..+32...(**7681**,7691)..+26...(**7
717**,7727)..+140...(7867,**7877**)..-4...(**7873**,7883)..+44...(7927,**7937**)..+122...
(8059,
8069)..+32...(**8101**,8111)..+50...(**8161**,8171)..+38...(**8209**,8219)..+2...(**8221**,
8231)..+2...(
8233,8243)..+20...(8263,**8273**)..+14...(8287,**8297**)..+56...(**8353**,8363)..+14...
(
8377,8387)..+32...(8419,**8429**)..+98...(8527,**8537**)..+26...(8563,**8573**)..+26...
(8599,
8609)..+80...(**8689**,8699)..+32...(8731,**8741**)..-4...(**8737**,8747)..+74...(**8821**,
8831)..+8...(8839,
8849)..+74...(8923,**8933**)..+8...(**8941**,8951)..+50...(**9001**,9011)..+38...(**9049**,
9059)..+68...(9127,
9137)..+14...(9151,**9161**)..+38...(9199,**9209**)..+74...(9283,**9293**)..+110...(940
3,
9413)..+8...(**9421**,9431)..+32...(9463,**9473**)..+38...(9511,**9521**)..+92...(**9613**,
9623)..-4...(9619,
9629)..+50...(9679,**9689**)..+44...(**9733**,9743)..-4...(9739,**9749**)..+32...(**9781**,
9791)..+38...(**9829**,9839)..+92...(9931,**9941**)

Insgesamt sind es 270 Semizwillinge im Abstand 10

Semizwillinge im Abstand 100 (ohne die Primzahl 2) im Intervall von 2 bis 10000

(3,103)..-96...(7,107)..-94...(
13,113)..-82...(31,131)..-94...**(37,137)**..-70...(67,167)..-94...**(73,173)**..-9
4...(79,179)..-82...(
97,197)..-70...(127,227)..-88...(139,239)..-88...(151,251)..-94...**(157,257)**
..-94...(163,263)..-82...(
181,281)..-88...**(193,293)**..-82...(211,311)..-28...(283,383)..-52...(331,431
)..-82...(
349,449)..-82...(367,467)..-88...(379,479)..-70...**(409,509)**..-88...**(421,521**
)..-64...(
457,557)..-94...(463,563)..-76...(487,587)..-88...(499,599)..-58...**(541,641**
)..-94...(547,647)..-70...(
577,677)..-76...**(601,701)**..-82...(619,719)..-76...(643,743)..-82...**(661,761**
)..-88...(
673,773)..-64...**(709,809)**..-82...(727,827)..-88...(739,839)..-82...**(757,857**
)..-70...(787,887)..-76...(811,911)..-82...(
829,929)..-76...**(853,953)**..-76...**(877,977)**..-94...(883,983)..-64...(919,101
9)..-28...(991,1091)..-94...(
997,1097)..-88...**(1009,1109)**..-58...(1051,1151)..-88...(1063,1163)..-76...(
1087,1187)..-94...(
1093,1193)..-76...**(1117,1217)**..-94...(1123,1223)..-94...**(1129,1229)**..-28...
(
1201,1301)..+26...(1327,1427)..-46...**(1381,1481)**..-82...(1399,1499)..-76...
(1423,1523)..-70...(
1453,1553)..-94...(1459,1559)..-88...(1471,1571)..-88...(1483,1583)..-16...
(1567,1667)..-70...(
1597,1697)..-88...**(1609,1709)**..-88...**(1621,1721)**..+2...(1723,1823)..-76...(
1747,1847)..-70...(
1777,1877)..-88...**(1789,1889)**..-88...**(1801,1901)**..-70...(1831,1931)..-58...
(
1873,1973)..-94...(1879,1979)..+8...(1987,2087)..-88...(1999,2099)..-88...(
2011,2111)..-82...(
2029,2129)..-76...**(2053,2153)**..-40...**(2113,2213)**..-76...**(2137,2237)**..-94...
(2143,2243)..-4...(2239,2339)..-88...(2251,2351)..-70...(
2281,2381)..-88...**(2293,2393)**..-82...(2311,2411)..-70...**(2341,2441)**..-94...
(2347,2447)..-70...(
2377,2477)..+44...**(2521,2621)**..-64...**(2557,2657)**..-64...**(2593,2693)**..-16...
(
2677,2777)..-88...**(2689,2789)**..-70...(2719,2819)..-22...**(2797,2897)**..-94...
(2803,2903)..-46...(
2857,2957)..+62...(3019,3119)..-82...**(3037,3137)**..-70...(3067,3167)..-58...
(
3109,3209)..-88...**(3121,3221)**..+8...**(3229,3329)**..-70...(3259,3359)..-88...(
3271,3371)..-64...(3307,3407)..-94...(
3313,3413)..-52...**(3361,3461)**..-70...(3391,3491)..-58...**(3433,3533)**..-76...
(
3457,3557)..-40...**(3517,3617)**..-58...(3559,3659)..-88...(3571,3671)..+26...
(
3697,3797)..-64...**(3733,3833)**..-10...(3823,3923)..-76...(3847,3947)..-58...
(
3889,3989)..-82...(3907,4007)..-88...(3919,4019)..+8...(4027,4127)..-70...(
4057,4157)..-46...(4111,4211)..-82...**(4129,4229)**..-76...**(4153,4253)**..-94...
(4159,4259)..+14...(
4273,4373)..-76...**(4297,4397)**..-40...**(4357,4457)**..-94...(4363,4463)..-40...
(4423,4523)..-76...(4447,4547)..-64...(4483,4583)..-34...(
4549,4649)..-58...(4591,4691)..-88...(4603,4703)..-82...**(4621,4721)**..-70...
(4651,4751)..+38...(
4789,4889)..-58...(4831,4931)..-28...(4903,5003)..-94...**(4909,5009)**..-58...
(4951,5051)..-64...(4987,5087)..-88...(4999,5099)..+80...(5179,5279)..-82..
.(
5197,5297)..-88...**(5209,5309)**..-76...**(5233,5333)**..-52...**(5281,5381)**..+26...

(5407,5507)..-88...(5419,5519)..-88...(5431,5531)..+26...(
5557,5657)..-88...(**5569,5669**)..-28...(**5641,5741**)..-58...(5683,5783)..-82...
(
5701,5801)..-58...(5743,5843)..-94...(**5749,5849**)..-70...(5779,5879)..-52...
(5827,5927)..-88...(5839,5939)..-58...(
5881,5981)..-28...(**5953,6053**)..-10...(6043,6143)..-70...(**6073,6173**)..-52...
(
6121,6221)..-58...(6163,6263)..-64...(6199,6299)..-88...(6211,6311)..-94...
(
6217,6317)..-88...(**6229,6329**)..+44...(**6373,6473**)..-52...(**6421,6521**)..-70...
(6451,6551)..-82...(
6469,6569)..-88...(**6481,6581**)..-28...(**6553,6653**)..-34...(6619,6719)..-82...
(
6637,6737)..-76...(**6661,6761**)..-82...(6679,6779)..-88...(6691,6791)..-88...
(6703,6803)..-70...(
6733,6833)..-70...(6763,6863)..+8...(6871,6971)..-88...(6883,6983)..+44...(
7027,7127)..+2...(
7129,7229)..-22...(7207,7307)..+26...(**7333,7433**)..-82...(7351,7451)..-34...
(
7417,7517)..-58...(7459,7559)..-82...(**7477,7577**)..-88...(**7489,7589**)..-82...
(7507,7607)..-58...(
7549,7649)..-76...(**7573,7673**)..-82...(7591,7691)..-88...(7603,7703)..+14...
(
7717,7817)..-94...(7723,7823)..-82...(**7741,7841**)..-88...(**7753,7853**)..+140..
.(
7993,8093)..-82...(8011,8111)..-94...(**8017,8117**)..+74...(8191,8291)..-28...
(8263,8363)..-94...(
8269,8369)..-82...(8287,8387)..-58...(**8329,8429**)..+14...(8443,8543)..-16...
(8527,8627)..-64...(8563,8663)..-82...(
8581,8681)..-82...(8599,8699)..-58...(**8641,8741**)..-94...(8647,8747)..-40...
(8707,8807)..-88...(8719,8819)..-88...(8731,8831)..-94...(
8737,8837)..-76...(**8761,8861**)..+2...(8863,8963)..-34...(**8929,9029**)..-88...(
8941,9041)..+62...(9103,9203)..-94...(**9109,9209**)..-82...(9127,9227)..-70...
(
9157,9257)..-76...(**9181,9281**)..-40...(**9241,9341**)..-64...(**9277,9377**)..-58...
(9319,9419)..-82...(
9337,9437)..-46...(9391,9491)..-94...(**9397,9497**)..-76...(**9421,9521**)..-88...
(
9433,9533)..-94...(9439,9539)..+80...(9619,9719)..-76...(9643,9743)..-94...
(
9649,9749)..-16...(**9733,9833**)..-94...(9739,9839)..-52...(9787,9887)..-58...
(**9829,9929**)..-22...(9907,10007)..-40...(9967,10067)

Insgesamt sind es 260 Semizwillinge im Abstand 100

<u>Anmerkungen:</u>

1. Die Primzahlen der Obergruppe OG I wurden fett gedruckt und die Primzahlen der Obergruppe OG II nicht fett gedruckt.
2. Die zwischen den Zwillingen auftretenden Abstände wurden jeweils angegeben. Diese Abstände variieren sehr stark.

Anlage 3

Die Bestimmung von Zwillingen mod 2, mod 4, mod 6, mod 8, mod 10 und mod 100
im Intervall 1.000.000 bis 1.010.000:

Primzahlzwillinge (ohne die Primzahl 2)
im Intervall von 1000000 bis 1010000

(
1000037,1000039)..+172...(1000211,1000213)..+76...(1000289,1000291)..+136..
.(1000427,
1000429)..+148...(1000577,1000579)..+40...(1000619,1000621)..+46...(1000667
,
1000669)..+52...(1000721,1000723)..+124...(1000847,1000849)..+10...(1000859
,
1000861)..+58...(1000919,1000921)..+166...(1001087,1001089)..+232...(100132
1,1001323)..+64...(1001387,1001389)..+160...(1001549,1001551)..+256...(1001
807,
1001809)..+172...(1001981,1001983)..+166...(1002149,1002151)..+106...(10022
57,1002259)..+82...(1002341,1002343)..+4...(1002347,1002349)..+10...(100235
9,
1002361)..+358...(1002719,1002721)..+46...(1002767,1002769)..+82...(1002851
,
1002853)..+76...(1002929,1002931)..+70...(1003001,1003003)..+106...(1003109
,1003111)..+88...(1003199,
1003201)..+148...(1003349,1003351)..+10...(1003361,1003363)..+4...(1003367,
1003369)..+250...(1003619,1003621)..+196...(1003817,1003819)..+88...(100390
7,
1003909)..+208...(1004117,1004119)..+538...(1004657,1004659)..+10...(100466
9,1004671)..+76...(1004747,1004749)..+322...(1005071,1005073)..+58...(10051
31,
1005133)..+106...(1005239,1005241)..+130...(1005371,1005373)..+64...(100543
7,1005439)..+112...(1005551,1005553)..+64...(1005617,1005619)..+58...(10056
77,1005679)..+232...(1005911,1005913)..+238...(1006151,1006153)..+16...(100
5169,1006171)..+46...(1006217,1006219)..+82...(1006301,1006303)..+4...(1006
307,
1006309)..+22...(1006331,1006333)..+4...(1006337,1006339)..+52...(1006391,1
006393)..+76...(1006469,1006471)..+310...(1006781,1006783)..+94...(1006877,
1006879)..+142...(
1007021,1007023)..+94...(1007117,1007119)..+178...(1007297,1007299)..+298..
.(
007597,1007599)..+82...(1007681,1007683)..+46...(1007729,1007731)..+226...

007957,1007959)..+82...(1008041,1008043)..+364...(1008407,1008409)..+10...
1008419,
008421)..+166...(1008587,1008589)..+22...(1008611,1008613)..+166...(100877
,
008781)..+70...(1008851,1008853)..+4...(1008857,1008859)..+52...(1008911,1
08913)..+76...(1008989,1008991)..+166...(1009157,1009159)..+40...(1009199,
009201)..+88...(1009289,1009291)..+10...(1009301,1009303)..+16...(1009319,
009321)..+178...(1009499,1009501)..+148...(1009649,1009651)..+340...(10099
1,1009993)

nsgesamt sind es 84 Zwillinge

Semizwillinge im Abstand 4 (ohne die Primzahl 2) im Intervall von 1000000 bis 1010000

(
1000033,1000037)..+80...(**1000117,1000121**)..+62...(1000183,1000187)..+62...(
1000249,1000253)..+140...(**1000393,1000397**)..+26...(1000423,1000427)..+26...
(
1000453,1000457)..+80...(**1000537,1000541**)..+428...(**1000969,1000973**)..+26...
(1000999,1001003)..+20...(1001023,1001027)..+62...(
1001089,1001093)..+80...(**1001173,1001177**)..+146...(1001323,1001327)..+200..
.(1001527,1001531)..+152...(1001683,1001687)..+110...(
1001797,1001801)..+176...(**1001977,1001981**)..+92...(**1002073,1002077**)..+182..
.(1002259,1002263)..+80...(1002343,1002347)..+176...(1002523,1002527)..+92.
..(1002619,1002623)..+86...(
1002709,1002713)..+56...(**1002769,1002773**)..+44...(**1002817,1002821**)..+32...(
1002853,1002857)..+56...(**1002913,1002917**)..+170...(1003087,1003091)..+272..
.(1003363,1003367)..+260...(1003627,1003631)..+98...(
1003729,1003733)..+20...(**1003753,1003757**)..+152...(**1003909,1003913**)..+140..
.(
1004053,1004057)..+80...(**1004137,1004141**)..+308...(**1004449,1004453**)..+290..
.(1004743,1004747)..+230...(
1004977,1004981)..+332...(**1005313,1005317**)..+92...(**1005409,1005413**)..+230..
.(1005643,1005647)..+356...(1006003,1006007)..+80...(1006087,1006091)..+56.
..(1006147,1006151)..+38...(
1006189,1006193)..+44...(**1006237,1006241**)..+8...(**1006249,1006253**)..+50...(1
006303,1006307)..+26...(
1006333,1006337)..+206...(1006543,1006547)..+62...(**1006609,1006613**)..+20...
(
1006633,1006637)..+242...(1006879,1006883)..+50...(**1006933,1006937**)..+50...
(1006987,1006991)..+656...(1007647,1007651)..+68...(1007719,1007723)..+26..
.(
1007749,1007753)..+14...(1007767,1007771)..+86...(**1007857,1007861**)..+26...(
1007887,1007891)..+122...(
1008013,1008017)..+20...(**1008037,1008041**)..+188...(**1008229,1008233**)..+200..
.(
1008433,1008437)..+62...(1008499,1008503)..+104...(1008607,1008611)..+2...(
1008613,1008617)..+236...(**1008853,1008857**)..+2...(1008859,1008863)..+116...
(1008979,1008983)..+170...(
1009153,1009157)..+2...(1009159,1009163)..+26...(**1009189,1009193**)..+50...(1
009243,1009247)..+110...(
1009357,1009361)..+8...(**1009369,1009373**)..+110...(1009483,1009487)..+506...
(**1009993,1009997**)

Insgesamt sind es 80 Semizwillinge im Abstand 4

Semizwillinge im Abstand 6 (ohne die Primzahl 2)
im Intervall von 1000000 bis 1010000

(
1000033,1000039)..+148...(1000187,**1000193**)..+0...(**1000193**,1000199)..+198...
(
1000397,1000403)..+0...(1000403,**1000409**)..+14...(1000423,**1000429**)..+112...(
1000541,1000547)..+144...(1000691,**1000697**)..+320...(**1001017**,1001023)..+58..
.(
1001081,1001087)..+0...(1001087,**1001093**)..+60...(**1001153**,1001159)..+32...(1
001191,
1001197)..+124...(**1001321**,1001327)..+20...(1001347,**1001353**)..+28...(**1001381**
,1001387)..+176...(1001563,
1001569)..+18...(1001587,**1001593**)..+208...(**1001801**,1001807)..+134...(**100194**
1,1001947)..+0...(1001947,**1001953**)..+24...(**1001977**,1001983)..+0...(1001983,
1001989)..+88...(**1002077**,1002083)..+60...(1002143,**1002149**)..+92...(**1002241**,
1002247)..+10...(
1002257,1002263)..+78...(**1002341**,1002347)..-4...(1002343,**1002349**)..+78...(1
002427,
1002433)..+18...(1002451,**1002457**)..+24...(**1002481**,1002487)..+0...(1002487,**1**
002493)..+18...(1002511,**1002517**)..+0...(**1002517**,1002523)..+54...(**1002577**,10
02583)..+64...(1002647,
1002653)..+60...(**1002713**,1002719)..+48...(1002767,**1002773**)..+78...(1002851,
1002857)..+0...(**1002857**,1002863)..+24...(1002887,**1002893**)..+0...(**1002893**,10
02899)..+74...(
1002973,1002979)..+112...(1003091,**1003097**)..+0...(**1003097**,1003103)..+0...(1
003103,
1003109)..+84...(**1003193**,1003199)..+74...(**1003273**,1003279)..+82...(**1003361**,
1003367)..-4...(1003363,
1003369)..+42...(1003411,**1003417**)..+46...(1003463,**1003469**)..+74...(1003543,
1003549)..+72...(**1003621**,1003627)..+114...(**1003741**,1003747)..+0...(1003747,
1003753)..+4...(**1003757**,1003763)..+144...(1003907,**1003913**)..+44...(**1003957**,
1003963)..+64...(1004027,
1004033)..+24...(**1004057**,1004063)..+98...(**1004161**,1004167)..+106...(**1004273**
,1004279)..+8...(1004287,
1004293)..+24...(**1004317**,1004323)..+154...(**1004477**,1004483)..+78...(**1004561**
,1004567)..+84...(1004651,
1004657)..+14...(1004671,**1004677**)..+60...(**1004737**,1004743)..+0...(1004743,**1**
004749)..+162...(1004911,**1004917**)..+64...(**1004981**,1004987)..+20...(1005007,
1005013)..+0...(**1005013**,1005019)..+54...(**1005073**,1005079)..+22...(**1005101**,1
005107)..+96...(1005203,
1005209)..+8...(**1005217**,1005223)..+0...(1005223,**1005229**)..+58...(1005287,**10**
05293)..+344...(**1005637**,1005643)..+178...(**1005821**,1005827)..+0...(1005827,**1**
005833)..+98...(1005931,**1005937**)..+210...(1006147,**1006153**)..+10...(1006163,
1006169)..+2...(1006171,**1006177**)..+54...(1006231,**1006237**)..+64...(**1006301**,1
006307)..-4...(1006303,
1006309)..+22...(1006331,**1006337**)..-4...(**1006333**,1006339)..+22...(**1006361**,1
006367)..+96...(1006463,
1006469)..+38...(1006507,**1006513**)..+70...(1006583,**1006589**)..+258...(1006847
,
1006853)..+24...(**1006877**,1006883)..+8...(1006891,**1006897**)..+276...(**1007173**,
1007179)..+64...(1007243,
1007249)..+104...(**1007353**,1007359)..+22...(**1007381**,1007387)..+336...(100772
8,
1007729)..+24...(**1007753**,1007759)..+42...(**1007801**,1007807)..+0...(1007807,**1**
007813)..+0...(**1007813**,1007819)..+114...(**1007933**,1007939)..+32...(1007971,**1**
007977)..+54...(1008031,**1008037**)..+0...(**1008037**,1008043)..+138...(**1008181**,1
008187)..+0...(1008187,
1008193)..+0...(**1008193**,1008199)..+24...(1008223,**1008229**)..+4...(**1008233**,10
08239)..+18...(
1008257,1008263)..+54...(**1008317**,1008323)..+24...(1008347,**1008353**)..+20...(
1008373,1008379)..+22...(**1008401**,1008407)..+86...(**1008493**,1008499)..+42...(

1008541,1008547)..+60...(1008607,**1008613**)..-2...(1008611,**1008617**)..+156...(**1008773**,1008779)..+72...(1008851,**1008857**)..-4...(**1008853**,1008859)..-2...(**1008857**,1008863)..+120...(1008983,**1008989**)..+164...(**1009153**,1009159)..-2...(**1009157**,1009163)..+30...(**1009193**,1009199)..+2...(**1009201**,1009207)..+30...(**1009237**,1009243)..+190...(**1009433**,1009439)..+62...(**1009501**,1009507)..+24...(1009531,
1009537)..+84...(**1009621**,1009627)..+10...(**1009637**,1009643)..+0...(1009643,**1009649**)..+92...(**1009741**,1009747)..+34...(**1009781**,1009787)..+50...(**1009837**,1009843)..+148...(1009991,**1009997**)..+0...(**1009997**,1010003)

Insgesamt sind es 149 Semizwillinge im Abstand 6

Semizwillinge im Abstand 8 (ohne die Primzahl 2) im Intervall von 1000000 bis 1010000

(1000151,1000159)..+814...(
1000973,1000981)..+100...(**1001081,1001089**)..+214...(1001303,1001311)..+70...(
1001381,1001389)..+70...(1001459,1001467)..+154...(**1001621,1001629**)..+172...(
1001801,1001809)..+22...(1001831,1001839)..+94...(**1001933,1001941**)..+40...(
1001981,1001989)..+94...(1002083,1002091)..+10...(**1002101,1002109**)..+34...(
1002143,1002151)..+190...(
1002341,1002349)..+154...(1002503,1002511)..+58...(**1002569,1002577**)..+136...(
1002713,1002721)..+88...(**1002809,1002817**)..+46...(1002863,1002871)..+232...
(1003103,1003111)..+22...(
1003133,1003141)..+52...(**1003193,1003201**)..+160...(**1003361,1003369**)..+232...(
1003601,1003609)..+10...(1003619,1003627)..+106...(**1003733,1003741**)..+22...
(1003763,1003771)..+118...(
1003889,1003897)..+382...(1004279,1004287)..+76...(1004363,1004371)..+70...
(
1004441,1004449)..+4...(**1004453,1004461**)..+190...(1004651,1004659)..+10...(
1004669,1004677)..+226...(1004903,1004911)..+130...(**1005041,1005049**)..+22...(1005071,1005079)..+130...(
1005209,1005217)..+484...(**1005701,1005709**)..+454...(1006163,1006171)..-2...
(
1006169,1006177)..+64...(**1006241,1006249**)..+52...(**1006301,1006309**)..+22...(
1006331,1006339)..+94...(
1006433,1006441)..+22...(1006463,1006471)..+382...(**1006853,1006861**)..+22...
(1006883,1006891)..+88...(1006979,1006987)..+94...(
1007081,1007089)..+40...(**1007129,1007137**)..+172...(**1007309,1007317**)..+142...(1007459,1007467)..+52...(1007519,1007527)..+22...(
1007549,1007557)..+136...(**1007693,1007701**)..+10...(1007711,1007719)..+4...(
1007723,1007731)..+28...(1007759,1007767)..+52...(1007819,1007827)..+412...
(1008239,1008247)..+76...(1008323,1008331)..+70...(
1008401,1008409)..+154...(1008563,1008571)..+202...(**1008773,1008781**)..+28...(
1008809,1008817)..+34...(1008851,1008859)..+4...(1008863,1008871)..+112...(
1008983,1008991)..+202...(
1009193,1009201)..-2...(1009199,1009207)..+154...(**1009361,1009369**)..+130...
(1009499,1009507)..+94...(
1009601,1009609)..+34...(1009643,1009651)..+250...(**1009901,1009909**)

Insgesamt sind es 77 Semizwillinge im Abstand 8

Semizwillinge im Abstand 10 (ohne die Primzahl 2) im Intervall von 1000000 bis 1010000

(1000183,
1000193)..+110...(1000303,**1000313**)..+44...(**1000357**,1000367)..+26...(**1000393**
,1000403)..+134...(
1000537,1000547)..+32...(1000579,**1000589**)..+20...(**1000609**,1000619)..+50...(
1000669,1000679)..+170...(**1000849**,1000859)..+62...(**1000921**,1000931)..+86...
(
1001017,1001027)..+284...(1001311,**1001321**)..+80...(**1001401**,1001411)..+80...
(1001491,
1001501)..+128...(**1001629**,1001639)..+20...(1001659,**1001669**)..+44...(**1001713**
,1001723)..+74...(
1001797,1001807)..+14...(**1001821**,1001831)..+242...(**1002073**,1002083)..+8...(
1002091,
1002101)..+146...(1002247,**1002257**)..+32...(**1002289**,1002299)..+50...(**1002349**
,1002359)..+98...(
1002457,1002467)..+26...(**1002493**,1002503)..+14...(**1002517**,1002527)..+182...
(
1002709,1002719)..+68...(1002787,**1002797**)..+56...(**1002853**,1002863)..+176...
(1003039,
1003049)..+38...(1003087,**1003097**)..+254...(1003351,**1003361**)..+146...(100350
7,
1003517)..+92...(**1003609**,1003619)..+2...(**1003621**,1003631)..+116...(1003747,
1003757)..-4...(**1003753**,1003763)..+116...(1003879,**1003889**)..+8...(**1003897**,1
003907)..+146...(
1004053,1004063)..+230...(**1004293**,1004303)..+224...(1004527,**1004537**)..+14..
.(1004551,
1004561)..+98...(1004659,**1004669**)..+8...(**1004677**,1004687)..+50...(**1004737**,1
004747)..+230...(
1004977,1004987)..+32...(1005019,**1005029**)..+104...(**1005133**,1005143)..+86...
(
1005229,1005239)..+110...(**1005349**,1005359)..+68...(1005427,**1005437**)..+20...
(
1005457,1005467)..+26...(**1005493**,1005503)..+38...(**1005541**,1005551)..+86...(
1005637,1005647)..+104...(1005751,**1005761**)..+362...(1006123,**1006133**)..+20..
.(
1006153,1006163)..+68...(1006231,**1006241**)..+110...(1006351,**1006361**)..+350..
.(1006711,
1006721)..+248...(**1006969**,1006979)..+110...(**1007089**,1007099)..+20...(100711
9,
1007129)..+170...(1007299,**1007309**)..+290...(1007599,**1007609**)..+74...(100768
3,
1007693)..+8...(**1007701**,1007711)..+8...(1007719,**1007729**)..+20...(**1007749**,10
07759)..+272...(1008031,
1008041)..+158...(1008199,**1008209**)..+14...(1008223,**1008233**)..-4...(**1008229**,
1008239)..+8...(1008247,
1008257)..+152...(**1008409**,1008419)..+74...(**1008493**,1008503)..+104...(100860
7,
1008617)..+236...(**1008853**,1008863)..+38...(**1008901**,1008911)..+2...(**1008913**,
1008923)..+14...(
1008937,1008947)..+32...(1008979,**1008989**)..+164...(**1009153**,1009163)..+26...
(
1009189,1009199)..+38...(**1009237**,1009247)..+44...(1009291,**1009301**)..+326...
(1009627,**1009637**)..+290...(1009927,**1009937**)..+56...(**1009993**,1010003)

Insgesamt sind es 92 Semizwillinge im Abstand 10

Semizwillinge im Abstand 100 (ohne die Primzahl 2) im Intervall von 1000000 bis 1010000

(**1000033,1000133**)..-34...(1000099,1000199)..+14...(**1000213,1000313**)..-10...(1000303,1000403)..-46...(**1000357,1000457**)..+122...(1000579,1000679)..-58...(**1000621,1000721**)..+248...(**1000969,1001069**)..-88...(**1000981,1001081**)..-58...(1001023,1001123)..+68...(1001191,1001291)..+20...(1001311,1001411)..-64...(1001347,1001447)..-46...(**1001401,1001501**)..-70...(1001431,1001531)..+38...(**1001569,1001669**)..-82...(1001587,1001687)..-4...(1001683,1001783)..+194...(**1001977,1002077**)..-94...(1001983,1002083)..-34...(**1002049,1002149**)..-76...(**1002073,1002173**)..-82...(1002091,1002191)..+50...(**1002241,1002341**)..-94...(1002247,1002347)..-88...(1002259,1002359)..+44...(1002403,1002503)..-76...(1002427,1002527)..-4...(1002523,1002623)..-70...(**1002553,1002653**)..-34...(1002619,1002719)..-10...(**1002709,1002809**)..-88...(**1002721,1002821**)..-70...(1002751,1002851)..-64...(1002787,1002887)..-70...(**1002817,1002917**)..+86...(1003003,1003103)..+38...(**1003141,1003241**)..+122...(1003363,1003463)..-94...(**1003369,1003469**)..-52...(**1003417,1003517**)..+224...(**1003741,1003841**)..+116...(**1003957,1004057**)..-94...(1003963,1004063)..+338...(**1004401,1004501**)..-40...(**1004461,1004561**)..-10...(1004551,1004651)..+356...(1005007,1005107)..+80...(1005187,1005287)..-70...(**1005217,1005317**)..-4...(**1005313,1005413**)..+14...(1005427,1005527)..-46...(**1005481,1005581**)..-88...(**1005493,1005593**)..+68...(**1005661,1005761**)..+176...(**1005937,1006037**)..+26...(1006063,1006163)..-10...(**1006153,1006253**)..-22...(1006231,1006331)..-94...(**1006237,1006337**)..-70...(1006267,1006367)..-34...(**1006333,1006433**)..-40...(**1006393,1006493**)..+20...(**1006513,1006613**)..+38...(1006651,1006751)..+32...(1006783,1006883)..-4...(1006879,1006979)..-88...(1006891,1006991)..+326...(**1007317,1007417**)..-58...(1007359,1007459)..+38...(**1007497,1007597**)..+104...(1007701,1007801)..-82...(1007719,1007819)..+38...(**1007857,1007957**)..+44...(**1008001,1008101**)..-70...(1008031,1008131)..+26...(**1008157,1008257**)..-34...(1008223,1008323)..-76...(1008247,1008347)..+170...(**1008517,1008617**)..+320...(**1008937,1009037**)..+122...(1009159,1009259)..-70...(**1009189,1009289**)..-88...(**1009201,1009301**)..-58...(1009243,1009343)..+14...(**1009357,1009457**)..-70...(1009387,1009487)..-88...(1009399,1009499)..+2...(**1009501,1009601**)..-64...(**1009537,1009637**)..-10...(1009627,1009727)..+110...(**1009837,1009937**)

Insgesamt sind es 92 Semizwillinge im Abstand 100

Anmerkungen:

- Die Primzahlen der Obergruppe OG I wurden fett gedruckt und die Primzahlen der Obergruppe OG II nicht fett gedruckt.
- Zwischen den Zwillingen wurden die Abstände, die zwichen ihnen auftreten angegeben. Diese Abstände variieren sehr stark.

Anlage 4

Die Generierung von Primzahlen vom Typ SM_p und SF_p mit dem MuPAD - Testprogramm, die in *Folge 1*, *Folge 2* und *Folge 3* auftreten und von Zwillingen mod 4.

1. Die Primzahlen SM_p der *Folge 1* im Intervall i := 0 bis 30

- ```
  for i from 0 to 30 do: if (isprime(5+10*i-2)and isprime (5+10*i-2))
  then print (("i",-2, " = ", 5+10*i-2, "5+10*i", 5+10*i-2)) end_if: end_for: Folge 1:
  ```

```
"i", -2, " = ", 3, "5+10*i", 3          1.1
"i", -2, " = ", 13, "5+10*i", 13        1.2
"i", -2, " = ", 23, "5+10*i", 23        1.3
"i", -2, " = ", 43, "5+10*i", 43        1.4
"i", -2, " = ", 53, "5+10*i", 53        1.5
"i", -2, " = ", 73, "5+10*i", 73        1.6
"i", -2, " = ", 83, "5+10*i", 83        1.7
"i", -2, " = ", 103, "5+10*i", 103      1.8
"i", -2, " = ", 113, "5+10*i", 113      1.9
"i", -2, " = ", 163, "5+10*i", 163      1.10
"i", -2, " = ", 173, "5+10*i", 173      1.11
"i", -2, " = ", 193, "5+10*i", 193      1.12
"i", -2, " = ", 223, "5+10*i", 223      1.13
"i", -2, " = ", 233, "5+10*i", 233      1.14
"i", -2, " = ", 263, "5+10*i", 263      1.15
"i", -2, " = ", 283, "5+10*i", 283      1.16
"i", -2, " = ", 293, "5+10*i", 293      1.17
```

Anmerkungen:

Im Intervall i:= 0 bis 30 sind in *Folge 1* insgesamt 17 Primzahlen SM_p situiert, die in der Endziffer: EZ3, auftreten.

---

2. Die Primzahlen SF_p der *Folge 1* im Intervall i:= 0 bis 30

```
for i from 0 to 30 do: if (isprime(5+10*i+2)and isprime (5+10*i+2))
then print (("i",+2, " = ", 5+10*i+2, "5+10*i", 5+10*i+2)) end_if: end_for: Folge 1:
```

```
"i", 2, " = ", 7, "5+10*i", 7           1.1
"i", 2, " = ", 17, "5+10*i", 17         1.2
"i", 2, " = ", 37, "5+10*i", 37         1.3
"i", 2, " = ", 47, "5+10*i", 47         1.4
"i", 2, " = ", 67, "5+10*i", 67         1.5
"i", 2, " = ", 97, "5+10*i", 97         1.6
"i", 2, " = ", 107, "5+10*i", 107       1.7
```
```

```
"i", 2, " = ", 127, "5+10*i", 127                                1.8
"i", 2, " = ", 137, "5+10*i", 137                                1.9
"i", 2, " = ", 157, "5+10*i", 157                                1.10
"i", 2, " = ", 167, "5+10*i", 167                                1.11
"i", 2, " = ", 197, "5+10*i", 197                                1.12
"i", 2, " = ", 227, "5+10*i", 227                                1.13
"i", 2, " = ", 257, "5+10*i", 257                                1.14
"i", 2, " = ", 277, "5+10*i", 277                                1.15
"i", 2, " = ", 307, "5+10*i", 307                                1.16
```

<u>Anmerkung:</u>
Im Intervall i:= 0 bis 30 sind in *Folge 1* insgesamt 16 Primzahlen SF_p, EZ7 situiert.

---

3. Die Bestimmung der Zwillinge mod 4, die in der *Folge 1* im Intervall i:= 0 bis 30

- ```
  for i from 0 to 30 do: if (isprime(5+10*i-2)and isprime (5+10*i+2))
  then print (("i",-2, " = ", 5+10*i-2, "5+10*i", 5+10*i+2)) end_if: end_for: Folge 1:
  ```

```
"i", -2, " = ", 3, "5+10*i", 7                                   1.1
"i", -2, " = ", 13, "5+10*i", 17                                 1.2
"i", -2, " = ", 43, "5+10*i", 47                                 1.3
"i", -2, " = ", 103, "5+10*i", 107                               1.4
"i", -2, " = ", 163, "5+10*i", 167                               1.5
"i", -2, " = ", 193, "5+10*i", 197                               1.6
"i", -2, " = ", 223, "5+10*i", 227                               1.7
```

<u>Anmerkung:</u>

1. Im Intervall i:= 0 bis 30 sind in der *Folge 1* insgesamt 7 Zwillinge mod 4 situiert.
2. Aus der Liste dieser 7 Zwillinge mod 4 ist ersichtlich, dass die Primzahlen (SM_p, SF_p) die diese Zwillinge mod 4 bilden, jeweils den gleichen Obergruppen (OG I bzw. OG II) angehören

---

4. Die Bestimmung der Primzahlen SM_p der *Folge 2* im Intervall i:= 0 bia 30

- ```
  for i from 0 to 30 do: if (isprime(9+10*i-2)and isprime (9+10*i-2))
  ```
- ```
  then print (("i",-2, " = ", 9+10*i-2, "9+10*i", 9+10*i-2)) end_if: end_for: Folge 2:
  ```

```
"i", -2, " = ", 7, "9+10*i", 7                                   2.1
"i", -2, " = ", 17, "9+10*i", 17                                 2.2
"i", -2, " = ", 37, "9+10*i", 37                                 2.3
"i", -2, " = ", 47, "9+10*i", 47                                 2.4
"i", -2, " = ", 67, "9+10*i", 67                                 2.5
"i", -2, " = ", 97, "9+10*i", 97                                 2.6
"i", -2, " = ", 107, "9+10*i", 107                               2.7
"i", -2, " = ", 127, "9+10*i", 127                               2.8
"i", -2, " = ", 137, "9+10*i", 137                               2.9
"i", -2, " = ", 157, "9+10*i", 157                               2.10
"i", -2, " = ", 167, "9+10*i", 167                               2.11
```
```

```
"i", -2, " = ", 197, "9+10*i", 197                2.12
"i", -2, " = ", 227, "9+10*i", 227                2.13
"i", -2, " = ", 257, "9+10*i", 257                2.14
"i", -2, " = ", 277, "9+10*i", 277                2.15
"i", -2, " = ", 307, "9+10*i", 307                2.16
```

Anmerkungen:

1. Im Intervall i:= 0 bis 30 sind in *Folge 2* insgesamt 16 Primzahlen SM_p situiert.
2. Alle 16 Primzahlen SM_p treten in der Endziffer EZ7 auf.
3. Die Primzahlen SM_p sind dabei Elemente der Obergruppen: OG I oder. OG II.

---

4. Die Bestimmung der Primzahlen SF_p, die im Intervall i.= 0 bis 30 situiert sind:

- ```
  for i from 0 to 30 do: if (isprime(9+10*i+2)and isprime (9+10*i+2))
  then print (("i",+2, " = ", 9+10*i+2, "9+10*i", 9+10*i+2)) end_if: end_for: Folge 2:
  ```

```
"i", 2, " = ", 11, "9+10*i", 11                2.1
"i", 2, " = ", 31, "9+10*i", 31                2.2
"i", 2, " = ", 41, "9+10*i", 41                2.3
"i", 2, " = ", 61, "9+10*i", 61                2.4
"i", 2, " = ", 71, "9+10*i", 71                2.5
"i", 2, " = ", 101, "9+10*i", 101              2.6
"i", 2, " = ", 131, "9+10*i", 131              2.7
"i", 2, " = ", 151, "9+10*i", 151              2.8
"i", 2, " = ", 181, "9+10*i", 181              2.9
"i", 2, " = ", 191, "9+10*i", 191              2.10
"i", 2, " = ", 211, "9+10*i", 211              2.11
"i", 2, " = ", 241, "9+10*i", 241              2.12
"i", 2, " = ", 251, "9+10*i", 251              2.13
"i", 2, " = ", 271, "9+10*i", 271              2.14
"i", 2, " = ", 281, "9+10*i", 281              2.15
"i", 2, " = ", 311, "9+10*i", 311              2.16
```

Anmerkungen:

1. In der *Folge 2* sind im Intervall: i:= 0 bis 30 insgesamt 16 Primzahlen SF_p situiert.
2. Alle Primzahlen SF_p treten in der *Folge 2* in der Endziffer EZ1 auf und sind (fallweise) Elemente der Obergruppen: OG I bzw. OG II.

---

5. Die Bestimmung der Zwillinge mod 4, die in der *Folge 2* situiert sind;

```
for i from 0 to 30 do: if (isprime(9+10*i-2)and isprime (9+10*i+2))
then print (("i",-2, " = ", 9+10*i-2, "9+10*i", 9+10*i+2)) end_if: end_for: Folge 2:
```

```
"i", -2, " = ", 7, "9+10*i", 11                2.1
"i", -2, " = ", 37, "9+10*i", 41               2.2
```
```

```
"i", -2, " = ", 67, "9+10*i", 71        2.3
"i", -2, " = ", 97, "9+10*i", 101       2.4
"i", -2, " = ", 127, "9+10*i", 131      2.5
"i", -2, " = ", 277, "9+10*i", 281      2.6
"i", -2, " = ", 307, "9+10*i", 311      2.7
```

Anmerkungen:

1. Im Intervall i:= 0 bia 30 sind in der *Folge 2* insgesamt 7 Zwillinge mod 4 situiert.

2. Die Primzahlen SM_p und SF_p dieser Zwillinge mod 4 sind stets Elemente der gleichen Obergruppen: OG I bzw. OG II.

3. Alle Primzahlen SM_p dieser Zwillinge mod 4 haben stets die Endziffer EZ7 und alle Primzahlen SF_p stets die Endziffer EZ1.

---

6. Die Bestimmung der Primzahlen SM_p, die in der *Folge 3* situiert sind:

```
• for i from 0 to 30 do: if (isprime(11+10*i-2)and isprime (11+10*i-2))
• then print (("i",-2, " = ", 11+10*i-2, "11+10*i", 11+10*i-2)) end_if: end_for: Folge 3:
"i", -2, " = ", 19, "11+10*i", 19       3.1
"i", -2, " = ", 29, "11+10*i", 29       3.2
"i", -2, " = ", 59, "11+10*i", 59       3.3
"i", -2, " = ", 79, "11+10*i", 79       3.4
"i", -2, " = ", 89, "11+10*i", 89       3.5
"i", -2, " = ", 109, "11+10*i", 109     3.6
"i", -2, " = ", 139, "11+10*i", 139     3.7
"i", -2, " = ", 149, "11+10*i", 149     3.8
"i", -2, " = ", 179, "11+10*i", 179     3.9
"i", -2, " = ", 199, "11+10*i", 199     3.10
"i", -2, " = ", 229, "11+10*i", 229     3.11
"i", -2, " = ", 239, "11+10*i", 239     3.12
"i", -2, " = ", 269, "11+10*i", 269     3.13
```

Anmerkungen:

1. In der *Folge 3* treten im Intervall i:= 0 bis 30 insgesamt 13 Primzahlen SM_p, EZ9 auf.

2. Diese Primzahlen SM_p sind fallweise Elemente der Obergruppe OG I bzw. OG II.

---

7. Die Bestimmung der Primzahlen SF_p, die in der *Folge 3* situiert sind:

```
for i from 0 to 30 do: if (isprime(11+10*i+2)and isprime (11+10*i+2))
then print (("i",+2, " = ", 11+10*i+2, "11+10*i", 11+10*i+2)) end_if: end_for: Folge 3:
"i", 2, " = ", 13, "11+10*i", 13        3.1
"i", 2, " = ", 23, "11+10*i", 23        3.2
"i", 2, " = ", 43, "11+10*i", 43        3.3
"i", 2, " = ", 53, "11+10*i", 53        3.4
```

```
"i", 2, " = ", 73, "11+10*i", 73                3.5
"i", 2, " = ", 83, "11+10*i", 83                3.6
"i", 2, " = ", 103, "11+10*i", 103              3.7
"i", 2, " = ", 113, "11+10*i", 113              3.8
"i", 2, " = ", 163, "11+10*i", 163              3.9
"i", 2, " = ", 173, "11+10*i", 173              3.10
"i", 2, " = ", 193, "11+10*i", 193              3.11
"i", 2, " = ", 223, "11+10*i", 223              3.11
"i", 2, " = ", 233, "11+10*i", 233              3.12
"i", 2, " = ", 263, "11+10*i", 263              3.13
"i", 2, " = ", 283, "11+10*i", 283              3.14
"i", 2, " = ", 293, "11+10*i", 293              3.15
"i", 2, " = ", 313, "11+10*i", 313              3.16
```

<u>Anmerkungen:</u>

1. In der *Folge 3* sind im Intervall i:= 0 bis 30 insgesamt 16 Primzahlen SF_p situiert.
2. Diese Primzahlen SF_p treten fallweise in den Endziffern: EZ1, EZ3, EZ7 und EZ9 auf.

---

8. Die Bestimmung der in *Folge 3* im Intervall i:= 0 bis 30 situierten Zwillinge mod 4:

- `for i from 0 to 30 do: if (isprime(11+10*i-2)and isprime (11+10*i+2))`
- `then print (("i",-2, " = ", 11+10*i-2, "11+10*i", 11+10*i+2)) end_if: end_for:` *Folge 3:*

```
"i", -2, " = ", 19, "11+10*i", 23               3.1
"i", -2, " = ", 79, "11+10*i", 83               3.2
"i", -2, " = ", 109, "11+10*i", 113             3.3
"i", -2, " = ", 229, "11+10*i", 233             3.4
```

<u>Anmerkung:</u>

1. In der *Folge 3* treten im Intervall i:= 0 bis 30 insgesamt 4 Zwillinge mod 4 auf.
2. Die Primzahlen SM_p und SF_p, die in den Zwillingen mod 4 auftreten, sind stets (gleichzeitig) Elemente der Obergruppen: OG I bzw. OG II.

Die Bestimmung der Primzahlen SM_p und SF_p, die im Intervall i:= 10^350 bis 10^350 +6000 in *Folge 1*, *Folge 2* und *Folge 3* situiert sind und der hier situierten Zwillinge mod 4:

1. Die Bestimmung der Primzahlen SM_p, die in der *Folge 1* situiert sind.

- `for i from 10^350 to 10^350+3000 do: if (isprime(5+10*i-2)and isprime`
- `(5+10*i-2)) then print (("i",-2, " = ", "5+10*i", 5+10*i-2)) end_if: end_for:` Folge 1*:

```
"i", -2, " = ", "5+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000004953                                              1*.1

"i", -2, " = ", "5+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
```

```
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000007413                                            1*.2
```

```
"i", -2, " = ", "5+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000010183                                            1*.3

"i", -2, " = ", "5+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000015903                                            1*.4

"i", -2, " = ", "5+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000024523                                            1*.5
```

<u>Anmerkungen:</u>

1. In diesem Intervall sind 5 Primzahlen SM_p in der *Folge 1* situiert.

2. Die Primzahlen SM_p der *Folge 1* treten (allesamt) in der Endziffern: EZ3 auf.

---

2. Die Bestimmung der Primzahlen SF_p, die in *Folge 1* in diesem Zahlenbereich auftreten:

```
• for i from 10^350 to 10^350+3000  do: if (isprime(5+10*i+2)and isprime
• (5+10*i+2)) then print (("i",+2, " = ", "5+10*i", 5+10*i+2)) end_if: end_for:   Folge 1*:
```

```
"i", 2, " = ", "5+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000207                                            1*.1

"i", 2, " = ", "5+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000001447                                            1*.2

"i", 2, " = ", "5+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000005977                                            1*.3

"i", 2, " = ", "5+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000006687                                            1*.4

"i", 2, " = ", "5+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
```

```
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000009237                                       1*.5

"i", 2, " = ", "5+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000013507                                       1*.6

"i", 2, " = ", "5+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000018067                                       1*.7

"i", 2, " = ", "5+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000019777                                       1*.8

"i", 2, " = ", "5+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000022167                                       1*.9

"i", 2, " = ", "5+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000025557                                       1*.10

"i", 2, " = ", "5+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000026827                                       1*11
```

<u>Anmerkungen:</u>

1. In diesem Zahlenbereich sind in *Folge 1* insgesamt 11 Primzahlen SF_p situiert.

2. Diese Primzahlen SF_p haben (allesamt) die Endziffer EZ7..

---

3. Die Bestimmung der Zwillinge mod 4, die in diesem Zahlenbereich in der *Folge 1* auftreten:

<u>Anmerkung:</u>

. In diesem Zahlenbereich sind in der *Folge 1* keine Zwillinge mod 4 situiert.

. In dem erweiterten Intervall: 10^350 - 10^350 +6000 ist der folgende Zwilling mo 4 situiert:

```
for i from 10^350 to 10^350+6000  do: if (isprime(5+10*i-2)and isprime (5+10*i+2))
   then print (("i",-2, 5+10*i-2, " = ", "5+10*i", 5+10*i+2)) end_if: end_for: Folge 1*:

i", -2,

000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000057063

, " = ", "5+10*i",                                                        1*.1
```

```
1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000057067
```

Anmerkung:

Die Primzahlen (SM_p, SF_p) dieses Zwillings mod 4 sind (beide) Elemente der Obergruppe OG II.

---

4. Die Bestimmung der Primzahlen SM_p, die in der *Folge 2* situiert sind:

- for i from 10^350 to 10^350+3000 do: if (isprime(5+10*i-2)and isprime
- (5+10*i-2)) then print (("i",-2, " = ", "5+10*i", 5+10*i-2)) end_if: end_for: *Folge 2*:*

```
"i", -2, " = ", "5+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000004953                                      2*.1

"i", -2, " = ", "5+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000007413                                      2*.2

"i", -2, " = ", "5+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000010183                                      2*.3

"i", -2, " = ", "5+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000015903                                      2*.4

"i", -2, " = ", "5+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000024523                                      2*.5
```

Anmerkung:

1. In diesem Zahlenbereich sind in *Folge 2* insgesamt 5 Primzahlen SM_p situiert.

2. Es bestätigt sich die Gesetzmäßigkeit, dass alle Primzahlen SM_p in der *Folge 2* nur in der Endziffer EZ3 auftreten.

. Die Bestimmung der Primzahlen SF_p, die in der *Folge 2* auftreten:

for i from 10^350 to 10^350+3000 do: if (isprime (5+10*i+2)and isprime
(5+10*i+2)) then print (("i", +2, " = ", "5+10*i", 5+10*i+2)) end_if: end_for: *Folge 2*:*

```
i", 2, " = ", "5+10*i",

0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
```

```
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000207                                              2*.1

"i", 2, " = ", "5+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000001447                                              2*.2

"i", 2, " = ", "5+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000005977                                              2*.3

"i", 2, " = ", "5+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000006687                                              2*.4

"i", 2, " = ", "5+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000009237                                              2*.5

"i", 2, " = ", "5+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000013507                                              2*.6

"i", 2, " = ", "5+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000018067                                              2*.7

"i", 2, " = ", "5+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000019777                                              2*.8

"i", 2, " = ", "5+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000022167                                              2*.9

"i", 2, " = ", "5+10*i",

0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000025557                                              2*.10

"i", 2, " = ", "5+10*i",

0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000026827                                              2*.11
```

Anmerkung:

1. In diesem Zahlenbereich sin in der *Folge 2* insgesamt 11 Primzahlen SF_p situiert.

2 Alle Primzahlen SF_p in *Folge 2* treten (allesamt) in der Endziffer EZ7 auf.

---

6. Die Bestimmung der Zwillinge mod 4, die in der *Folge 2* auftreten:

Anmerkung:

1. In diesem (relativ großen) Zahlen-Intervall treten in der *Folge 2* keine Zwillinge mod 4 auf.

2. Im erweiterten Intervall von i:= 10^350 bis 10^350+600 ist der folgende Zwilling mod 4 situiert:

```
• for i from 10^350 to 10^350+6000 do: if (isprime (5+10*i-2)and isprime(5+10*i+2))
•   then print (("i", +2,5+10*i-2, " = ", "5+10*i", 5+10*i+2)) end_if: end_for:   Folge 2*:
"i", 2,

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000057063

  , " = ", "5+10*i",                                                              2*.1

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000057067
```

---

7. Die Bestimmung der Primzahlen SM_p, die in *Folge 3* situiert sind:

```
• for i from 10^350 to 10^350+3000 do: if (isprime (11+10*i-2)and isprime (11+10*i-2)) then
  print (("i", -2, " = ", "11+10*i", 11+10*i-2)) end_if: end_for:
•
"i", -2, " = ", "11+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000249                                              3*.1

"i", -2, " = ", "11+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000004879                                              3*.2

"i", -2, " = ", "11+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000008059                                              3*.3

"i", -2, " = ", "11+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000010119                                              3*.4

"i", -2, " = ", "11+10*i",

0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
```

```
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000010939
```

```
"i", -2, " = ", "11+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012549                                         3*.5

"i", -2, " = ", "11+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000014809                                         3*.6

"i", -2, " = ", "11+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000022879                                         3*.7

"i", -2, " = ", "11+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000023179                                         3*.8
```

<u>Anmerkung:</u>

1. In diesem Zahlen-Intervall sind insgesamt 8 Primzahlen SM_p in der *Folge 3* situiert.

2. Diese Primzahlen SM_p treten (allesamt) in der Endziffer EZ9 auf.

---

8. Die Bestimmung der Primzahlen SF_p, die in *Folge 3* situiert sind:

```
• for i from 10^350 to 10^350+3000 do: if (isprime (11+10*i+2)and isprime (11+10*i+2))
      then print (("i", +2, " = ", "11+10*i", 11+10*i+2)) end_if: end_for:      Folge 3*:

"i", 2, " = ", "11+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000004953                                         3*.1

"i", 2, " = ", "11+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000007413                                         3*.2

"i", 2, " = ", "11+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000010183                                         3*.3

"i", 2, " = ", "11+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000015903                                         3*.4
```

```
"i", 2, " = ", "11+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000024523                                        3*.5
```

Anmerkung:

1. In diesem Zahlen-Intervall sind in der *Folge 3* insgesamt 5 Primzahlen SF_p situiert.

2. Diese Primzahlen SF_p treten in den Endziffern: EZ1, EZ3, EZ7 und EZ9 auf.

---

9. Die Bestimmung der Zwillinge mod 4, die in der *Folge 3* situiert sind:

Anmerkung:

1. In diesem Zahlen-Intervall treten keine Zwillinge mod 4 in der *Folge 3* auf.

2. Dies ist eine Bestätigung der Asymptote-Gesetzmäßigkeit

3. Im erweiterten Intervall i:= 10^350 bis 10^350+6000 ist ein Zwilling mod 4 in *Folge 3* situiert, der nachfolgend notiert wird:

```
• for i from 10^350 to 10^350+6000 do: if (isprime (11+10*i-2)and isprime (11+10*i+2)
• then print (("i", +2,11+10*i-2, " = ", "11+10*i", 11+10*i+2)) end_if: end_for:
•
"i", 2,

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000054669

  , " = ", "11+10*i",                                                        3*.1

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000054673
```

Anmerkung:

1. In diesem Intervall ist ein Zwilling mod 4 in der *Folge 3* situiert.
2. Die Primzahlen SM_p, SF_p, die diesen Zwilling mod 4 bilden, treten in den Endziffern: EZ9, EZ3 auf.
3. Dieser Sachverhalt bestätigt das Asymptote-Gesetz.

Schlussfolgerungen:

1. Der Vergleich der Anzahlen der Primzaahlen SM_p, SF_p und der Zwillinge mod 4 in :den drei Folgen: bestätigt eindeutig das Asymptote-Gesetz.

Die Zahlen der Primzahlen im Intervall Δ= 0 bis 30	*Folge 1*	*Folge 2*	Folge 3
SM_p :=	17	16	13
SF_p :=	16	16	16
Zwillinge mod 4 :=	7	7	4
Die Zahlen der Primzahlen im Intervall 10^350 bis 10^350+3000 (Δ= 3000):			
SM_p :=	5	5	8
SF_p :=	11	11	5
Zwillinge mod 4 :=	0	0	0
Zwillinge mod 4 (Δ= 6000) :=	1	1	1

Die "ein-eindeutigen" <u>neun</u> Ordnungs-Folgen der Primzahlen vom Typ SM_p, SF_p und der Zwillinge mod 4, die in den Folgen: *Folge 1*, *Folge 2* und *Folge 3* auftreten, bilden die Grundlage für die Konzeption einer neuen (bisher unbekannten) Daten-Verschlüsselung.

Anlage 5

Die Generierung von Primzahlen vom Typ CM_p und CF_p mit dem MuPAD- Testprogamm und die Bestimmung von Zwillingen mod 2 in *Folge 1*, *Folge 2* und *Folge 3*:

1. Die Bestimmung der Primzahlen CM_p, die im Intervall i:= 0 bis 50 in *Folge 1* auftreten:

- `for i from 0 to 50 do: if (isprime (12+30*i-1) and isprime (12+30*i-1)) then print`
- `(("i", i,-1, " = ", 12+30*i-1, "12+30*i", "12+30*i", 12+30*i-1)) end_if: end_for:`

Folge 1

```
"i", 0, -1, " = ", 11, "12+30*i", "12+30*i", 11          1.1
"i", 1, -1, " = ", 41, "12+30*i", "12+30*i", 41          1.2
"i", 2, -1, " = ", 71, "12+30*i", "12+30*i", 71          1.3
"i", 3, -1, " = ", 101, "12+30*i", "12+30*i", 101        1.4
"i", 4, -1, " = ", 131, "12+30*i", "12+30*i", 131        1.5
"i", 6, -1, " = ", 191, "12+30*i", "12+30*i", 191        1.6
"i", 8, -1, " = ", 251, "12+30*i", "12+30*i", 251        1.7
"i", 9, -1, " = ", 281, "12+30*i", "12+30*i", 281        1.8
"i", 10, -1, " = ", 311, "12+30*i", "12+30*i", 311       1.9
"i", 13, -1, " = ", 401, "12+30*i", "12+30*i", 401       1.10
"i", 14, -1, " = ", 431, "12+30*i", "12+30*i", 431       1.11
"i", 15, -1, " = ", 461, "12+30*i", "12+30*i", 461       1.12
"i", 16, -1, " = ", 491, "12+30*i", "12+30*i", 491       1.13
"i", 17, -1, " = ", 521, "12+30*i", "12+30*i", 521       1.14
"i", 21, -1, " = ", 641, "12+30*i", "12+30*i", 641       1.15
"i", 23, -1, " = ", 701, "12+30*i", "12+30*i", 701       1.16
"i", 25, -1, " = ", 761, "12+30*i", "12+30*i", 761       1.17
"i", 27, -1, " = ", 821, "12+30*i", "12+30*i", 821       1.18
"i", 29, -1, " = ", 881, "12+30*i", "12+30*i", 881       1.19
"i", 30, -1, " = ", 911, "12+30*i", "12+30*i", 911       1.20
"i", 31, -1, " = ", 941, "12+30*i", "12+30*i", 941       1.21
"i", 32, -1, " = ", 971, "12+30*i", "12+30*i", 971       1.22
"i", 34, -1, " = ", 1031, "12+30*i", "12+30*i", 1031     1.23
"i", 35, -1, " = ", 1061, "12+30*i", "12+30*i", 1061     1.24
"i", 36, -1, " = ", 1091, "12+30*i", "12+30*i", 1091     1.25
"i", 38, -1, " = ", 1151, "12+30*i", "12+30*i", 1151     1.26
"i", 39, -1, " = ", 1181, "12+30*i", "12+30*i", 1181     1.27
"i", 43, -1, " = ", 1301, "12+30*i", "12+30*i", 1301     1.28
"i", 45, -1, " = ", 1361, "12+30*i", "12+30*i", 1361     1.29
"i", 48, -1, " = ", 1451, "12+30*i", "12+30*i", 1451     1.30
"i", 49, -1, " = ", 1481, "12+30*i", "12+30*i", 1481     1.31
```

```
"i", 50, -1, " = ", 1511, "12+30*i", "12+30*i", 1511        1.32
```

Anmerkung:

1. Im Intervall i:= 0 bis 50 sind in der *Folge 1* insgesamt 32 Primzahlen CM_p situiert.

---

2. Die Bestimmung der Primzahlen CF_p, die im Intervall i:= 0 bis 50 in *Folge 1* auftreten:

- `for i from 0 to 50 do: if (isprime (12+30*i+1) and isprime (12+30*i+1)) then print`
- `(("i", i,+1, " = ", 12+30*i+1, "12+30*i", "12+30*i", 12+30*i+1)) end_if: end_for:`

Folge 1:

```
"i", 0, 1, " = ", 13, "12+30*i", "12+30*i", 13          1.1
"i", 1, 1, " = ", 43, "12+30*i", "12+30*i", 43          1.2
"i", 2, 1, " = ", 73, "12+30*i", "12+30*i", 73          1.3
"i", 3, 1, " = ", 103, "12+30*i", "12+30*i", 103        1.4
"i", 5, 1, " = ", 163, "12+30*i", "12+30*i", 163        1.5
"i", 6, 1, " = ", 193, "12+30*i", "12+30*i", 193        1.6
"i", 7, 1, " = ", 223, "12+30*i", "12+30*i", 223        1.7
"i", 9, 1, " = ", 283, "12+30*i", "12+30*i", 283        1.8
"i", 10, 1, " = ", 313, "12+30*i", "12+30*i", 313       1.9
"i", 12, 1, " = ", 373, "12+30*i", "12+30*i", 373       1.10
"i", 14, 1, " = ", 433, "12+30*i", "12+30*i", 433       1.11
"i", 15, 1, " = ", 463, "12+30*i", "12+30*i", 463       1.12
"i", 17, 1, " = ", 523, "12+30*i", "12+30*i", 523       1.13
"i", 20, 1, " = ", 613, "12+30*i", "12+30*i", 613       1.14
"i", 21, 1, " = ", 643, "12+30*i", "12+30*i", 643       1.15
"i", 22, 1, " = ", 673, "12+30*i", "12+30*i", 673       1.16
"i", 24, 1, " = ", 733, "12+30*i", "12+30*i", 733       1.17
"i", 27, 1, " = ", 823, "12+30*i", "12+30*i", 823       1.18
"i", 28, 1, " = ", 853, "12+30*i", "12+30*i", 853       1.19
"i", 29, 1, " = ", 883, "12+30*i", "12+30*i", 883       1.20
"i", 34, 1, " = ", 1033, "12+30*i", "12+30*i", 1033     1.21
"i", 35, 1, " = ", 1063, "12+30*i", "12+30*i", 1063     1.22
"i", 36, 1, " = ", 1093, "12+30*i", "12+30*i", 1093     1.23
"i", 37, 1, " = ", 1123, "12+30*i", "12+30*i", 1123     1.24
"i", 38, 1, " = ", 1153, "12+30*i", "12+30*i", 1153     1.25
"i", 40, 1, " = ", 1213, "12+30*i", "12+30*i", 1213     1.26
"i", 43, 1, " = ", 1303, "12+30*i", "12+30*i", 1303     1.27
"i", 47, 1, " = ", 1423, "12+30*i", "12+30*i", 1423     1.28
"i", 48, 1, " = ", 1453, "12+30*i", "12+30*i", 1453     1.29
"i", 49, 1, " = ", 1483, "12+30*i", "12+30*i", 1483     1.30
```

Anmerkung:

1. In *Folge 1* treten im Intervall i:= 0 bis 50 insgesamt 30 Primzahlen CF_p auf

---

3. Die Bestimmung von Zwillingen mod 2, die in *Folge 1* im Intervall i:= 0 bis 50 auftreten:

- ```
  for i from 0 to 50 do: if (isprime (12+30*i-1) and isprime (12+30*i+1)) then print
  (("i", i,-1, " = ", 12+30*i-1, "12+30*i", "12+30*i", 12+30*i+1)) end_if: end_for:
  ```

Folge 1:

```
"i", 0, -1, " = ", 11, "12+30*i", "12+30*i", 13          1.1
"i", 1, -1, " = ", 41, "12+30*i", "12+30*i", 43          1.2
"i", 2, -1, " = ", 71, "12+30*i", "12+30*i", 73          1.3
"i", 3, -1, " = ", 101, "12+30*i", "12+30*i", 103        1.4
"i", 6, -1, " = ", 191, "12+30*i", "12+30*i", 193        1.5
"i", 9, -1, " = ", 281, "12+30*i", "12+30*i", 283        1.6
"i", 10, -1, " = ", 311, "12+30*i", "12+30*i", 313       1.7
"i", 14, -1, " = ", 431, "12+30*i", "12+30*i", 433       1.8
"i", 15, -1, " = ", 461, "12+30*i", "12+30*i", 463       1.9
"i", 17, -1, " = ", 521, "12+30*i", "12+30*i", 523       1.10
"i", 21, -1, " = ", 641, "12+30*i", "12+30*i", 643       1.11
"i", 27, -1, " = ", 821, "12+30*i", "12+30*i", 823       1.12
"i", 29, -1, " = ", 881, "12+30*i", "12+30*i", 883       1.13
"i", 34, -1, " = ", 1031, "12+30*i", "12+30*i", 1033     1.14
"i", 35, -1, " = ", 1061, "12+30*i", "12+30*i", 1063     1.15
"i", 36, -1, " = ", 1091, "12+30*i", "12+30*i", 1093     1.16
"i", 38, -1, " = ", 1151, "12+30*i", "12+30*i", 1153     1.17
"i", 43, -1, " = ", 1301, "12+30*i", "12+30*i", 1303     1.18
"i", 48, -1, " = ", 1451, "12+30*i", "12+30*i", 1453     1.19
"i", 49, -1, " = ", 1481, "12+30*i", "12+30*i", 1483     1.20
```

<u>Anmerkung:</u>

1. Im Intervall i:= 0 bis 50 sind in *Folge 1* insgesamt 20 Zwillinge mod 2 situiert.
2. Die Primzahlen CM_p, SF_p treten (allesamt) in den Endziffern: EZ1, EZ3 auf.
3. Im Intervall i:= 0 bis 50 treten hingegen 32 Primzahlen CM_p und 30 Primzahlen CF_p auf.

---

4. Die Bestimmung der Primzahlen CM_p, die im Intervall i:= 0 bis 50 in *Folge 2* auftreten:

- ```
  for i from 0 to 50 do: if (isprime (18+30*i-1) and isprime (18+30*i-1)) then print
  (("i", i, -1, " = ", 18+30*i-1, "18+30*i", "18+30*i", 18+30*i-1)) end_if: end_for:
  ```

Folge 2:

```
"i", 0, -1, " = ", 17, "18+30*i", "18+30*i", 17       2.1
"i", 1, -1, " = ", 47, "18+30*i", "18+30*i", 47       2.2
"i", 3, -1, " = ", 107, "18+30*i", "18+30*i", 107     2.3
"i", 4, -1, " = ", 137, "18+30*i", "18+30*i", 137     2.4
"i", 5, -1, " = ", 167, "18+30*i", "18+30*i", 167     2.5
```

```
"i", 6, -1, " = ", 197, "18+30*i", "18+30*i", 197          2.6
"i", 7, -1, " = ", 227, "18+30*i", "18+30*i", 227          2.7
"i", 8, -1, " = ", 257, "18+30*i", "18+30*i", 257          2.8
"i", 10, -1, " = ", 317, "18+30*i", "18+30*i", 317         2.9
"i", 11, -1, " = ", 347, "18+30*i", "18+30*i", 347         2.10
"i", 15, -1, " = ", 467, "18+30*i", "18+30*i", 467         2.11
"i", 18, -1, " = ", 557, "18+30*i", "18+30*i", 557         2.12
"i", 19, -1, " = ", 587, "18+30*i", "18+30*i", 587         2.13
"i", 20, -1, " = ", 617, "18+30*i", "18+30*i", 617         2.14
"i", 21, -1, " = ", 647, "18+30*i", "18+30*i", 647         2.15
"i", 22, -1, " = ", 677, "18+30*i", "18+30*i", 677         2.16
"i", 26, -1, " = ", 797, "18+30*i", "18+30*i", 797         2.17
"i", 27, -1, " = ", 827, "18+30*i", "18+30*i", 827         2.18
"i", 28, -1, " = ", 857, "18+30*i", "18+30*i", 857         2.19
"i", 29, -1, " = ", 887, "18+30*i", "18+30*i", 887         2.20
"i", 31, -1, " = ", 947, "18+30*i", "18+30*i", 947         2.21
"i", 32, -1, " = ", 977, "18+30*i", "18+30*i", 977         2.22
"i", 36, -1, " = ", 1097, "18+30*i", "18+30*i", 1097       2.23
"i", 39, -1, " = ", 1187, "18+30*i", "18+30*i", 1187       2.24
"i", 40, -1, " = ", 1217, "18+30*i", "18+30*i", 1217       2.25
"i", 42, -1, " = ", 1277, "18+30*i", "18+30*i", 1277       2.26
"i", 43, -1, " = ", 1307, "18+30*i", "18+30*i", 1307       2.27
"i", 45, -1, " = ", 1367, "18+30*i", "18+30*i", 1367       2.28
"i", 47, -1, " = ", 1427, "18+30*i", "18+30*i", 1427       2.29
"i", 49, -1, " = ", 1487, "18+30*i", "18+30*i", 1487       2.30
```

<u>Anmerkung:</u>

1. Im Intervall i:= 0 bis 50 sind in der *Folge 2* insgesamt 30 Primzahlen CM_p situiert.
2. Alle Primzahlen CM_p haben die Endziffer EZ7.

---

5. Die Bestimmung der Primzahlen CF_p, die in der *Folge 2* situiert sind:

- `for i from 0 to 50 do: if (isprime (18+30*i+1) and isprime (18+30*i+1)) then print`
- `(("i", i,+1, " = ", 18+30*i+1, "18+30*i", "18+30*i", 18+30*i+1)) end_if: end_for:`

Folge 2:

```
"i", 0, 1, " = ", 19, "18+30*i", "18+30*i", 19       2.1
"i", 2, 1, " = ", 79, "18+30*i", "18+30*i", 79       2.2
"i", 3, 1, " = ", 109, "18+30*i", "18+30*i", 109     2.3
"i", 4, 1, " = ", 139, "18+30*i", "18+30*i", 139     2.4
"i", 6, 1, " = ", 199, "18+30*i", "18+30*i", 199     2.5
"i", 7, 1, " = ", 229, "18+30*i", "18+30*i", 229     2.6
"i", 11, 1, " = ", 349, "18+30*i", "18+30*i", 349    2.7
```

```
"i", 12, 1, " = ", 379, "18+30*i", "18+30*i", 379          2.8
"i", 13, 1, " = ", 409, "18+30*i", "18+30*i", 409          2.9
"i", 14, 1, " = ", 439, "18+30*i", "18+30*i", 439          2.10
"i", 16, 1, " = ", 499, "18+30*i", "18+30*i", 499          2.11
"i", 20, 1, " = ", 619, "18+30*i", "18+30*i", 619          2.12
"i", 23, 1, " = ", 709, "18+30*i", "18+30*i", 709          2.13
"i", 24, 1, " = ", 739, "18+30*i", "18+30*i", 739          2.14
"i", 25, 1, " = ", 769, "18+30*i", "18+30*i", 769          2.15
"i", 27, 1, " = ", 829, "18+30*i", "18+30*i", 829          2.16
"i", 28, 1, " = ", 859, "18+30*i", "18+30*i", 859          2.17
"i", 30, 1, " = ", 919, "18+30*i", "18+30*i", 919          2.18
"i", 33, 1, " = ", 1009, "18+30*i", "18+30*i", 1009        2.19
"i", 34, 1, " = ", 1039, "18+30*i", "18+30*i", 1039        2.20
"i", 35, 1, " = ", 1069, "18+30*i", "18+30*i", 1069        2.21
"i", 37, 1, " = ", 1129, "18+30*i", "18+30*i", 1129        2.22
"i", 41, 1, " = ", 1249, "18+30*i", "18+30*i", 1249        2.23
"i", 42, 1, " = ", 1279, "18+30*i", "18+30*i", 1279        2.24
"i", 46, 1, " = ", 1399, "18+30*i", "18+30*i", 1399        2.25
"i", 47, 1, " = ", 1429, "18+30*i", "18+30*i", 1429        2.26
"i", 48, 1, " = ", 1459, "18+30*i", "18+30*i", 1459        2.27
"i", 49, 1, " = ", 1489, "18+30*i", "18+30*i", 1489        2.28
```

Anmerkung:

1. Im Intervall i:= 0 bis 50 treten in der *Folge 2* insgesamt 28 Primzahlen CF_p auf.

2. Alle Primzahlen CF_p haben die Endziffer EZ9.

---

6. Die Bestimmung der Zwillinge mod 2, die im Intervall i:= 0 bis 50 in *Folge 2* auftreten:

- `for i from 0 to 50 do: if (isprime (18+30*i-1) and isprime (18+30*i+1) then print`
- `(("i", i,-1, " = ", 18+30*i-1, "18+30*i", "18+30*i", 18+30*i+1)) end_if: end_for:`

Folge 2:

```
"i", 0, -1, " = ", 17, "18+30*i", "18+30*i", 19            2.1
"i", 3, -1, " = ", 107, "18+30*i", "18+30*i", 109          2.2
"i", 4, -1, " = ", 137, "18+30*i", "18+30*i", 139          2.3
"i", 6, -1, " = ", 197, "18+30*i", "18+30*i", 199          2.4
"i", 7, -1, " = ", 227, "18+30*i", "18+30*i", 229          2.5
"i", 11, -1, " = ", 347, "18+30*i", "18+30*i", 349         2.6
"i", 20, -1, " = ", 617, "18+30*i", "18+30*i", 619         2.7
"i", 27, -1, " = ", 827, "18+30*i", "18+30*i", 829         2.8
"i", 28, -1, " = ", 857, "18+30*i", "18+30*i", 859         2.9
"i", 42, -1, " = ", 1277, "18+30*i", "18+30*i", 1279       2.10
"i", 47, -1, " = ", 1427, "18+30*i", "18+30*i", 1429       2.11
"i", 49, -1, " = ", 1487, "18+30*i", "18+30*i", 1489       2.12
```

Anmerkung:

1. Im Intervall i:= 0 bis 50 treten in *Folge 2* insgesamt 12 Zwillinge mod 2 auf.

2. Die Primzahlen CM_p, CF_p treten allesamt in den Endziffern: EZ7, EZ9 auf.

---

7. Die Bestimmung der Primzahlen CM_p, die in der *Folge 3* situiert sind:

```
• for i from 0 to 50 do: if (isprime (30+30*i-1) and isprime (30+30*i-1)) then print
• (("i", i,+1, " = ", 30+30*i-1, "30+30*i", "30+30*i", 30+30*i-1)) end_if: end_for:
                                                                          Folge 3:
        "i", 0, 1, " = ", 29, "30+30*i", "30+30*i", 29          3.1
        "i", 1, 1, " = ", 59, "30+30*i", "30+30*i", 59          3.2
        "i", 2, 1, " = ", 89, "30+30*i", "30+30*i", 89          3.3
        "i", 4, 1, " = ", 149, "30+30*i", "30+30*i", 149        3.4
        "i", 5, 1, " = ", 179, "30+30*i", "30+30*i", 179        3.5
        "i", 7, 1, " = ", 239, "30+30*i", "30+30*i", 239        3.6
        "i", 8, 1, " = ", 269, "30+30*i", "30+30*i", 269        3.7
        "i", 11, 1, " = ", 359, "30+30*i", "30+30*i", 359       3.8
        "i", 12, 1, " = ", 389, "30+30*i", "30+30*i", 389       3.9
        "i", 13, 1, " = ", 419, "30+30*i", "30+30*i", 419       3.10
        "i", 14, 1, " = ", 449, "30+30*i", "30+30*i", 449       3.11
        "i", 15, 1, " = ", 479, "30+30*i", "30+30*i", 479       3.12
        "i", 16, 1, " = ", 509, "30+30*i", "30+30*i", 509       3.13
        "i", 18, 1, " = ", 569, "30+30*i", "30+30*i", 569       3.14
        "i", 19, 1, " = ", 599, "30+30*i", "30+30*i", 599       3.15
        "i", 21, 1, " = ", 659, "30+30*i", "30+30*i", 659       3.16
        "i", 23, 1, " = ", 719, "30+30*i", "30+30*i", 719       3.17
        "i", 26, 1, " = ", 809, "30+30*i", "30+30*i", 809       3.18
        "i", 27, 1, " = ", 839, "30+30*i", "30+30*i", 839       3.19
        "i", 30, 1, " = ", 929, "30+30*i", "30+30*i", 929       3.20
        "i", 33, 1, " = ", 1019, "30+30*i", "30+30*i", 1019     3.21
        "i", 34, 1, " = ", 1049, "30+30*i", "30+30*i", 1049     3.22
        "i", 36, 1, " = ", 1109, "30+30*i", "30+30*i", 1109     3.23
        "i", 40, 1, " = ", 1229, "30+30*i", "30+30*i", 1229     3.24
        "i", 41, 1, " = ", 1259, "30+30*i", "30+30*i", 1259     3.25
        "i", 42, 1, " = ", 1289, "30+30*i", "30+30*i", 1289     3.26
        "i", 43, 1, " = ", 1319, "30+30*i", "30+30*i", 1319     3.27
        "i", 46, 1, " = ", 1409, "30+30*i", "30+30*i", 1409     3.28
        "i", 47, 1, " = ", 1439, "30+30*i", "30+30*i", 1439     3.29
        "i", 49, 1, " = ", 1499, "30+30*i", "30+30*i", 1499     3.30
```

Anmerkung:

1. In der *Folge 3* sind im Intervall i:= 0 bis 50 insgesamt 30 Primzahlen CM_p situiert.

– Alle Primzahlen CM_p der *Folge 3* haben die Endziffer EZ9.

---

8. Die Bestimmung der Primzahlen CF_p, die in der *Folge 3* situiert sind:

- `for i from 0 to 50 do: if (isprime (30+30*i+1) and isprime (30+30*i+1)) then print`
- `(("i", i,+1, " = ", 30+30*i+1, "30+30*i", "30+30*i", 30+30*i+1)) end_if: end_for:`

```
                                                              Folge 3:
"i", 0, 1, " = ", 31, "30+30*i", "30+30*i", 31               3.1
"i", 1, 1, " = ", 61, "30+30*i", "30+30*i", 61               3.2
"i", 4, 1, " = ", 151, "30+30*i", "30+30*i", 151             3.3
"i", 5, 1, " = ", 181, "30+30*i", "30+30*i", 181             3.4
"i", 6, 1, " = ", 211, "30+30*i", "30+30*i", 211             3.5
"i", 7, 1, " = ", 241, "30+30*i", "30+30*i", 241             3.6
"i", 8, 1, " = ", 271, "30+30*i", "30+30*i", 271             3.7
"i", 10, 1, " = ", 331, "30+30*i", "30+30*i", 331            3.8
"i", 13, 1, " = ", 421, "30+30*i", "30+30*i", 421            3.9
"i", 17, 1, " = ", 541, "30+30*i", "30+30*i", 541            3.10
"i", 18, 1, " = ", 571, "30+30*i", "30+30*i", 571            3.11
"i", 19, 1, " = ", 601, "30+30*i", "30+30*i", 601            3.12
"i", 20, 1, " = ", 631, "30+30*i", "30+30*i", 631            3.13
"i", 21, 1, " = ", 661, "30+30*i", "30+30*i", 661            3.14
"i", 22, 1, " = ", 691, "30+30*i", "30+30*i", 691            3.15
"i", 24, 1, " = ", 751, "30+30*i", "30+30*i", 751            3.16
"i", 26, 1, " = ", 811, "30+30*i", "30+30*i", 811            3.17
"i", 32, 1, " = ", 991, "30+30*i", "30+30*i", 991            3.18
"i", 33, 1, " = ", 1021, "30+30*i", "30+30*i", 1021          3.19
"i", 34, 1, " = ", 1051, "30+30*i", "30+30*i", 1051          3.20
"i", 38, 1, " = ", 1171, "30+30*i", "30+30*i", 1171          3.21
"i", 39, 1, " = ", 1201, "30+30*i", "30+30*i", 1201          3.22
"i", 40, 1, " = ", 1231, "30+30*i", "30+30*i", 1231          3.23
"i", 42, 1, " = ", 1291, "30+30*i", "30+30*i", 1291          3.24
"i", 43, 1, " = ", 1321, "30+30*i", "30+30*i", 1321          3.25
"i", 45, 1, " = ", 1381, "30+30*i", "30+30*i", 1381          3.26
"i", 48, 1, " = ", 1471, "30+30*i", "30+30*i", 1471          3.27
"i", 50, 1, " = ", 1531, "30+30*i", "30+30*i", 1531          3.28
```

<u>merkung:</u>

Im Intervall i:= 0 bis 50 sind in der *Folge 3* insgesamt 28 Primzahlen CF_p situiert.
Alle Primzahlen der *Folge 3* haben die Endziffer EZ1

Die Bestimmung der Zwillinge mod 2, die in der *Folge 3* situiert sind;

```
for i from 0 to 50 do: if (isprime (30+30*i-1) and isprime (30+30*i+1)) then print
(("i", i,-1, " = ", 30+30*i-1, "30+30*i", "30+30*i", 30+30*i+1)) end_if: end_for:
                                                              Folge 3:
```

```
"i", 0, -1, " = ", 29, "30+30*i", "30+30*i", 31            3.1
"i", 1, -1, " = ", 59, "30+30*i", "30+30*i", 61            3.2
"i", 4, -1, " = ", 149, "30+30*i", "30+30*i", 151          3.3
"i", 5, -1, " = ", 179, "30+30*i", "30+30*i", 181          3.4
"i", 7, -1, " = ", 239, "30+30*i", "30+30*i", 241          3.5
"i", 8, -1, " = ", 269, "30+30*i", "30+30*i", 271          3.6
"i", 13, -1, " = ", 419, "30+30*i", "30+30*i", 421         3.7
"i", 18, -1, " = ", 569, "30+30*i", "30+30*i", 571         3.8
"i", 19, -1, " = ", 599, "30+30*i", "30+30*i", 601         3.9
"i", 21, -1, " = ", 659, "30+30*i", "30+30*i", 661         3.10
"i", 26, -1, " = ", 809, "30+30*i", "30+30*i", 811         3.11
"i", 33, -1, " = ", 1019, "30+30*i", "30+30*i", 1021       3.12
"i", 34, -1, " = ", 1049, "30+30*i", "30+30*i", 1051       3.13
"i", 40, -1, " = ", 1229, "30+30*i", "30+30*i", 1231       3.14
"i", 42, -1, " = ", 1289, "30+30*i", "30+30*i", 1291       3.15
"i", 43, -1, " = ", 1319, "30+30*i", "30+30*i", 1321       3.16
```

<u>Anmerkung:</u>

1- Im Intervall i:= 0 bis 50 sind in der *Folge 3* insgesamt 16 Zwillinge mod 2 situiert.

2. Alle Primzahlen (CM_p, CF_p), die diese Zwillinge mod 2 bilden, haben die Endziffern: EZ9, EZ1.

<u>Die Schlussfolgerungen:</u>

1. Aus den Ermittlungen der Primzahlen CM_p und CF_p, die in den Folgen: *Folge 1, Folge 2, Folge 3* im Intervall i:= 0 bis 50 auftreten und aus den Bestimmungen der in diesen drei Folgen auftretenden Zwillinge mod 2, ist erkennbar, dass bei der "Entstehung" von Zwillingen mod 2 die folgende "reversiblen" Und-Bedingungen in den drei Folgen erfüllt sein müssen:

   $CF_p = CM_p+2 \Leftrightarrow. CM_p = CF_p -2$ ....: Die Und-Verkopplungen der Primzahlen CM_p und SF_p

2. Da diese Und-Bedingungen nicht für alle Primzahlen CM_p und CF_p erfüllt werden, treten in der Regel mehr Primzahlen CM_p und SF_p auf als Zwillinge mod 2.

---

10. Die Bestimmung der Primzahlen CM_p und CF_p in *Folge 1, Folge 2* und *Folge 3* und der Zwillinge mod 2 im Intervall: i:= 10^20 bis 10^20+500:

10.1 Die Bestimmung der Primzahlen CM_p in der *Folge 1*:

```
for i from 10^20 to 10^20+500 do: if (isprime (12+30*i-1) and isprime (12+30*i-1))
•       then print(("i", i, -1, " =", 12+30*i-1, "12+30*i-1")) end_if: end_for:
"i", 100000000000000000033, -1, " =", 3000000000000000001001, "12+30*i-1"
"i", 100000000000000000035, -1, " =", 3000000000000000001061, "12+30*i-1"
"i", 100000000000000000053, -1, " =", 3000000000000000001601, "12+30*i-1"
"i", 100000000000000000055, -1, " =", 3000000000000000001661, "12+30*i-1"
"i", 100000000000000000074, -1, " =", 3000000000000000002231, "12+30*i-1"
"i", 100000000000000000081, -1, " =", 3000000000000000002441, "12+30*i-1"
"i", 100000000000000000095, -1, " =", 3000000000000000002861, "12+30*i-1"
```

```
"i", 100000000000000000100, -1, " =", 3000000000000000003011, "12+30*i-1"
"i", 100000000000000000132, -1, " =", 3000000000000000003971, "12+30*i-1"
"i", 100000000000000000135, -1, " =", 3000000000000000004061, "12+30*i-1"
"i", 100000000000000000145, -1, " =", 3000000000000000004361, "12+30*i-1"
"i", 100000000000000000160, -1, " =", 3000000000000000004811, "12+30*i-1"
"i", 100000000000000000161, -1, " =", 3000000000000000004841, "12+30*i-1"
"i", 100000000000000000182, -1, " =", 3000000000000000005471, "12+30*i-1"
"i", 100000000000000000218, -1, " =", 3000000000000000006551, "12+30*i-1"
"i", 100000000000000000231, -1, " =", 3000000000000000006941, "12+30*i-1"
"i", 100000000000000000235, -1, " =", 3000000000000000007061, "12+30*i-1"
"i", 100000000000000000288, -1, " =", 3000000000000000008651, "12+30*i-1"
"i", 100000000000000000299, -1, " =", 3000000000000000008981, "12+30*i-1"
"i", 100000000000000000316, -1, " =", 3000000000000000009491, "12+30*i-1"
"i", 100000000000000000319, -1, " =", 3000000000000000009581, "12+30*i-1"
"i", 100000000000000000321, -1, " =", 3000000000000000009641, "12+30*i-1"
"i", 100000000000000000322, -1, " =", 3000000000000000009671, "12+30*i-1"
"i", 100000000000000000348, -1, " =", 3000000000000000010451, "12+30*i-1"
"i", 100000000000000000371, -1, " =", 3000000000000000011141, "12+30*i-1"
"i", 100000000000000000383, -1, " =", 3000000000000000011501, "12+30*i-1"
"i", 100000000000000000386, -1, " =", 3000000000000000011591, "12+30*i-1"
"i", 100000000000000000387, -1, " =", 3000000000000000011621, "12+30*i-1"
"i", 100000000000000000413, -1, " =", 3000000000000000012401, "12+30*i-1"
"i", 100000000000000000435, -1, " =", 3000000000000000013061, "12+30*i-1"
"i", 100000000000000000442, -1, " =", 3000000000000000013271, "12+30*i-1"
"i", 100000000000000000443, -1, " =", 3000000000000000013301, "12+30*i-1"
"i", 100000000000000000464, -1, " =", 3000000000000000013931, "12+30*i-1"
"i", 100000000000000000481, -1, " =", 3000000000000000014441, "12+30*i-1"
```

Anmerkung:

1. Im betrachteten Intervall sind in der *Folge 1* insgesamt 32 Primzahlen CM_p situiert.

2. Alle Primzahlen CM_p haben die Endziffer EZ1

---

10. 2 Die Bestimmung der Primzahlen CF_p in der *Folge 1:*

- ```
  for i from 10^20 to 10^20+500 do: if (isprime (12+30*i+1) and isprime (12+30*i+1))
     then print(("i", i, +1, " =", 12+30*i+1, "12+30*i+1")) end_if: end_for:
  ```

```
"i", 100000000000000000011, 1, " =", 3000000000000000000343, "12+30*i+1"
"i", 100000000000000000022, 1, " =", 3000000000000000000673, "12+30*i+1"
"i", 100000000000000000024, 1, " =", 3000000000000000000733, "12+30*i+1"
"i", 100000000000000000033, 1, " =", 3000000000000000001003, "12+30*i+1"
"i", 100000000000000000055, 1, " =", 3000000000000000001663, "12+30*i+1"
"i", 100000000000000000124, 1, " =", 3000000000000000003733, "12+30*i+1"
```
```

```
"i", 100000000000000000126, 1, " =", 3000000000000000003793, "12+30*i+1"
"i", 100000000000000000127, 1, " =", 3000000000000000003823, "12+30*i+1"
"i", 100000000000000000129, 1, " =", 3000000000000000003883, "12+30*i+1"
"i", 100000000000000000155, 1, " =", 3000000000000000004663, "12+30*i+1"
"i", 100000000000000000167, 1, " =", 3000000000000000005023, "12+30*i+1"
"i", 100000000000000000171, 1, " =", 3000000000000000005143, "12+30*i+1"
"i", 100000000000000000221, 1, " =", 3000000000000000006643, "12+30*i+1"
"i", 100000000000000000223, 1, " =", 3000000000000000006703, "12+30*i+1"
"i", 100000000000000000231, 1, " =", 3000000000000000006943, "12+30*i+1"
"i", 100000000000000000243, 1, " =", 3000000000000000007303, "12+30*i+1"
"i", 100000000000000000244, 1, " =", 3000000000000000007333, "12+30*i+1"
"i", 100000000000000000248, 1, " =", 3000000000000000007453, "12+30*i+1"
"i", 100000000000000000266, 1, " =", 3000000000000000007993, "12+30*i+1"
"i", 100000000000000000273, 1, " =", 3000000000000000008203, "12+30*i+1"
"i", 100000000000000000306, 1, " =", 3000000000000000009193, "12+30*i+1"
"i", 100000000000000000321, 1, " =", 3000000000000000009643, "12+30*i+1"
"i", 100000000000000000323, 1, " =", 3000000000000000009703, "12+30*i+1"
"i", 100000000000000000347, 1, " =", 3000000000000000010423, "12+30*i+1"
"i", 100000000000000000354, 1, " =", 3000000000000000010633, "12+30*i+1"
"i", 100000000000000000365, 1, " =", 3000000000000000010963, "12+30*i+1"
"i", 100000000000000000393, 1, " =", 3000000000000000011803, "12+30*i+1"
"i", 100000000000000000405, 1, " =", 3000000000000000012163, "12+30*i+1"
"i", 100000000000000000416, 1, " =", 3000000000000000012493, "12+30*i+1"
"i", 100000000000000000427, 1, " =", 3000000000000000012823, "12+30*i+1"
"i", 100000000000000000439, 1, " =", 3000000000000000013183, "12+30*i+1"
"i", 100000000000000000466, 1, " =", 3000000000000000013993, "12+30*i+1"
"i", 100000000000000000479, 1, " =", 3000000000000000014383, "12+30*i+1"
"i", 100000000000000000482, 1, " =", 3000000000000000014473, "12+30*i+1"
"i", 100000000000000000500, 1, " =", 3000000000000000015013, "12+30*i+1"
```

<u>Anmerkung:</u>

1. In der *Folge 1* sind insgesamt 37 Primzahlen CF_p situiert.

2. Alle Primzahlen CF_p haben die Endziffer EZ3

10.3 Die Bestimmung der Zwillinge mod 2, die in der *Folge 1* situiert sind:

- ```
  for i from 10^20 to 10^20+500 do: if (isprime (12+30*i-1) and isprime (12+30*i+1))
     then print(("i", i, -1, " =", 12+30*i-1, "12+30*i+1")) end_if: end_for:
  ```

```
"i", 100000000000000000033, -1, " =", 3000000000000000001001, "12+30*i+1"
"i", 100000000000000000055, -1, " =", 3000000000000000001661, "12+30*i+1"
"i", 100000000000000000231, -1, " =", 3000000000000000006941, "12+30*i+1"
"i", 100000000000000000321, -1, " =", 3000000000000000009641, "12+30*i+1"
```
```

Anmerkung:

1. In der *Folge 1* sind 4 Zwillinge mod 2 situiert,
2. Dagegen treten im gleichen Intervall 32 Primzahlen CM_p und 37 Primzahlen CF_p in *Folge 1* auf.

---

10. 4 Die Bestimmung der Primzahlen CM_p, die in *Folge 2* situiert sind:

- ```
  for i from 10^20 to 10^20+500 do: if (isprime (18+30*i-1) and isprime (18+30*i-1))
     then print(("i", i, -1, " =", 18+30*i-1, "18+30*i-1")) end_if: end_for:
  ```

```
"i", 100000000000000000002, -1, " =", 3000000000000000000077, "18+30*i-1"
"i", 100000000000000000025, -1, " =", 3000000000000000000767, "18+30*i-1"
"i", 100000000000000000048, -1, " =", 3000000000000000001457, "18+30*i-1"
"i", 100000000000000000051, -1, " =", 3000000000000000001547, "18+30*i-1"
"i", 100000000000000000058, -1, " =", 3000000000000000001757, "18+30*i-1"
"i", 100000000000000000088, -1, " =", 3000000000000000002657, "18+30*i-1"
"i", 100000000000000000093, -1, " =", 3000000000000000002807, "18+30*i-1"
"i", 100000000000000000097, -1, " =", 3000000000000000002927, "18+30*i-1"
"i", 100000000000000000160, -1, " =", 3000000000000000004817, "18+30*i-1"
"i", 100000000000000000164, -1, " =", 3000000000000000004937, "18+30*i-1"
"i", 100000000000000000176, -1, " =", 3000000000000000005297, "18+30*i-1"
"i", 100000000000000000193, -1, " =", 3000000000000000005807, "18+30*i-1"
"i", 100000000000000000220, -1, " =", 3000000000000000006617, "18+30*i-1"
"i", 100000000000000000227, -1, " =", 3000000000000000006827, "18+30*i-1"
"i", 100000000000000000233, -1, " =", 3000000000000000007007, "18+30*i-1"
"i", 100000000000000000235, -1, " =", 3000000000000000007067, "18+30*i-1"
"i", 100000000000000000258, -1, " =", 3000000000000000007757, "18+30*i-1"
"i", 100000000000000000261, -1, " =", 3000000000000000007847, "18+30*i-1"
"i", 100000000000000000272, -1, " =", 3000000000000000008177, "18+30*i-1"
"i", 100000000000000000296, -1, " =", 3000000000000000008897, "18+30*i-1"
"i", 100000000000000000307, -1, " =", 3000000000000000009227, "18+30*i-1"
"i", 100000000000000000327, -1, " =", 3000000000000000009827, "18+30*i-1"
"i", 100000000000000000334, -1, " =", 3000000000000000010037, "18+30*i-1"
"i", 100000000000000000340, -1, " =", 3000000000000000010217, "18+30*i-1"
"i", 100000000000000000429, -1, " =", 3000000000000000012887, "18+30*i-1"
"i", 100000000000000000449, -1, " =", 3000000000000000013487, "18+30*i-1"
"i", 100000000000000000456, -1, " =", 3000000000000000013697, "18+30*i-1"
"i", 100000000000000000495, -1, " =", 3000000000000000014867, "18+30*i-1"
```

Anmerkung:

1. In der *Folge 2* sind 28 Primzahlen CM_p situiert.
2. Alle Primzahlen CM_p haben die Endziffer EZ7

---

10. 5 Die Bestimmung der Primzahlen CF_p, die in der *Folge 2* situiert sind:

- ```
  for i from 10^20 to 10^20+500 do: if (isprime (18+30*i+1) and isprime (18+30*i+1))
     then print(("i", i, +1, " =", 18+30*i+1, "18+30*i+1")) end_if: end_for:
  ```

```
"i", 10000000000000000007, 1, " =", 300000000000000000229, "18+30*i+1"
"i", 10000000000000000010, 1, " =", 300000000000000000319, "18+30*i+1"
"i", 10000000000000000011, 1, " =", 300000000000000000349, "18+30*i+1"
"i", 10000000000000000012, 1, " =", 300000000000000000379, "18+30*i+1"
"i", 10000000000000000044, 1, " =", 300000000000000001339, "18+30*i+1"
"i", 10000000000000000049, 1, " =", 300000000000000001489, "18+30*i+1"
"i", 10000000000000000101, 1, " =", 300000000000000003049, "18+30*i+1"
"i", 10000000000000000112, 1, " =", 300000000000000003379, "18+30*i+1"
"i", 10000000000000000142, 1, " =", 300000000000000004279, "18+30*i+1"
"i", 10000000000000000150, 1, " =", 300000000000000004519, "18+30*i+1"
"i", 10000000000000000173, 1, " =", 300000000000000005209, "18+30*i+1"
"i", 10000000000000000189, 1, " =", 300000000000000005689, "18+30*i+1"
"i", 10000000000000000213, 1, " =", 300000000000000006409, "18+30*i+1"
"i", 10000000000000000231, 1, " =", 300000000000000006949, "18+30*i+1"
"i", 10000000000000000259, 1, " =", 300000000000000007789, "18+30*i+1"
"i", 10000000000000000305, 1, " =", 300000000000000009169, "18+30*i+1"
"i", 10000000000000000332, 1, " =", 300000000000000009979, "18+30*i+1"
"i", 10000000000000000338, 1, " =", 300000000000000010159, "18+30*i+1"
"i", 10000000000000000355, 1, " =", 300000000000000010669, "18+30*i+1"
"i", 10000000000000000364, 1, " =", 300000000000000010939, "18+30*i+1"
"i", 10000000000000000368, 1, " =", 300000000000000011059, "18+30*i+1"
"i", 10000000000000000369, 1, " =", 300000000000000011089, "18+30*i+1"
"i", 10000000000000000381, 1, " =", 300000000000000011449, "18+30*i+1"
"i", 10000000000000000387, 1, " =", 300000000000000011629, "18+30*i+1"
"i", 10000000000000000402, 1, " =", 300000000000000012079, "18+30*i+1"
"i", 10000000000000000406, 1, " =", 300000000000000012199, "18+30*i+1"
"i", 10000000000000000421, 1, " =", 300000000000000012649, "18+30*i+1"
"i", 10000000000000000422, 1, " =", 300000000000000012679, "18+30*i+1"
"i", 10000000000000000423, 1, " =", 300000000000000012709, "18+30*i+1"
"i", 10000000000000000436, 1, " =", 300000000000000013099, "18+30*i+1"
"i", 10000000000000000439, 1, " =", 300000000000000013189, "18+30*i+1"
"i", 10000000000000000442, 1, " =", 300000000000000013279, "18+30*i+1"
"i", 10000000000000000450, 1, " =", 300000000000000013519, "18+30*i+1"
"i", 10000000000000000472, 1, " =", 300000000000000014179, "18+30*i+1"
"i", 10000000000000000484, 1, " =", 300000000000000014539, "18+30*i+1"
"i", 10000000000000000495, 1, " =", 300000000000000014869, "18+30*i+1"
```

Anmerkung:

- In der *Folge 2* sind 36 Primzahlen CF_p situiert.
- Alle (diese) Primzahlen CF_p haben die Endziffer EZ9.

---

0. 6 Die Bestimmung der Zwillinge mod 2, die in *Folge 2* situiert sind:

- ```
  for i from 10^20 to 10^20+500 do: if (isprime (18+30*i-1) and isprime (18+30*i+1))
     then print(("i", i, -1, " =", 18+30*i-1, "18+30*i+1")) end_if: end_for:
  ```

```
"i", 100000000000000000495, -1, " =", 3000000000000000014867, "18+30*i+1"
```

<u>Anmerkung:</u>

1. In der *Folge 2* ist nur 1 Zwilling mod 2 situiert.

10.7.. Die Bestimmung der Primzahlen CM_p, die in der *Folge 3* situiert sind:

```
for i from 10^20 to 10^20+500 do: if (isprime (30+30*i-1) and isprime (30+30*i-1))
   then print(("i", i, -1, " =", 30+30*i-1, "30+30*i-1")) end_if: end_for:
```

```
"i", 100000000000000000003, -1, " =", 3000000000000000000119, "30+30*i-1"
"i", 100000000000000000010, -1, " =", 3000000000000000000329, "30+30*i-1"
"i", 100000000000000000028, -1, " =", 3000000000000000000869, "30+30*i-1"
"i", 100000000000000000037, -1, " =", 3000000000000000001139, "30+30*i-1"
"i", 100000000000000000074, -1, " =", 3000000000000000002249, "30+30*i-1"
"i", 100000000000000000076, -1, " =", 3000000000000000002309, "30+30*i-1"
"i", 100000000000000000107, -1, " =", 3000000000000000003239, "30+30*i-1"
"i", 100000000000000000114, -1, " =", 3000000000000000003449, "30+30*i-1"
"i", 100000000000000000124, -1, " =", 3000000000000000003749, "30+30*i-1"
"i", 100000000000000000125, -1, " =", 3000000000000000003779, "30+30*i-1"
"i", 100000000000000000161, -1, " =", 3000000000000000004859, "30+30*i-1"
"i", 100000000000000000173, -1, " =", 3000000000000000005219, "30+30*i-1"
"i", 100000000000000000177, -1, " =", 3000000000000000005339, "30+30*i-1"
"i", 100000000000000000187, -1, " =", 3000000000000000005639, "30+30*i-1"
"i", 100000000000000000199, -1, " =", 3000000000000000005999, "30+30*i-1"
"i", 100000000000000000200, -1, " =", 3000000000000000006029, "30+30*i-1"
"i", 100000000000000000215, -1, " =", 3000000000000000006479, "30+30*i-1"
"i", 100000000000000000226, -1, " =", 3000000000000000006809, "30+30*i-1"
"i", 100000000000000000235, -1, " =", 3000000000000000007079, "30+30*i-1"
"i", 100000000000000000238, -1, " =", 3000000000000000007169, "30+30*i-1"
"i", 100000000000000000285, -1, " =", 3000000000000000008579, "30+30*i-1"
"i", 100000000000000000293, -1, " =", 3000000000000000008819, "30+30*i-1"
"i", 100000000000000000298, -1, " =", 3000000000000000008969, "30+30*i-1"
"i", 100000000000000000315, -1, " =", 3000000000000000009479, "30+30*i-1"
"i", 100000000000000000332, -1, " =", 3000000000000000009989, "30+30*i-1"
"i", 100000000000000000333, -1, " =", 3000000000000000010019, "30+30*i-1"
"i", 100000000000000000336, -1, " =", 3000000000000000010109, "30+30*i-1"
"i", 100000000000000000340, -1, " =", 3000000000000000010229, "30+30*i-1"
"i", 100000000000000000383, -1, " =", 3000000000000000011519, "30+30*i-1"
"i", 100000000000000000395, -1, " =", 3000000000000000011879, "30+30*i-1"
"i", 100000000000000000402, -1, " =", 3000000000000000012089, "30+30*i-1"
"i", 100000000000000000405, -1, " =", 3000000000000000012179, "30+30*i-1"
"i", 100000000000000000451, -1, " =", 3000000000000000013559, "30+30*i-1"
"i", 100000000000000000452, -1, " =", 3000000000000000013589, "30+30*i-1"
```
```

```
"i", 100000000000000000487, -1, " =", 3000000000000000014639, "30+30*i-1"
```

Anmerkung:

1. In der *Folge 3* sind 32 Primzahlen CM_p situiert, die alle die Endziffer EZ9 haben

---

10. 8 Die Bestimmung der Primzahhlen CF_p, die in der *Folge 3* situiert sind:

- for i from 10^20 to 10^20+500 do: if (isprime (30+30*i+1) and isprime (30+30*i+1)) then print(("i", i, +1, " =", 30+30*i+1, "30+30*i+1")) end_if: end_for:

```
"i", 100000000000000000018, 1, " =", 3000000000000000000571, "30+30*i+1"
"i", 100000000000000000022, 1, " =", 3000000000000000000691, "30+30*i+1"
"i", 100000000000000000033, 1, " =", 3000000000000000001021, "30+30*i+1"
"i", 100000000000000000047, 1, " =", 3000000000000000001441, "30+30*i+1"
"i", 100000000000000000115, 1, " =", 3000000000000000003481, "30+30*i+1"
"i", 100000000000000000117, 1, " =", 3000000000000000003541, "30+30*i+1"
"i", 100000000000000000131, 1, " =", 3000000000000000003961, "30+30*i+1"
"i", 100000000000000000159, 1, " =", 3000000000000000004801, "30+30*i+1"
"i", 100000000000000000164, 1, " =", 3000000000000000004951, "30+30*i+1"
"i", 100000000000000000181, 1, " =", 3000000000000000005461, "30+30*i+1"
"i", 100000000000000000184, 1, " =", 3000000000000000005551, "30+30*i+1"
"i", 100000000000000000219, 1, " =", 3000000000000000006601, "30+30*i+1"
"i", 100000000000000000230, 1, " =", 3000000000000000006931, "30+30*i+1"
"i", 100000000000000000265, 1, " =", 3000000000000000007981, "30+30*i+1"
"i", 100000000000000000271, 1, " =", 3000000000000000008161, "30+30*i+1"
"i", 100000000000000000275, 1, " =", 3000000000000000008281, "30+30*i+1"
"i", 100000000000000000283, 1, " =", 3000000000000000008521, "30+30*i+1"
"i", 100000000000000000286, 1, " =", 3000000000000000008611, "30+30*i+1"
"i", 100000000000000000319, 1, " =", 3000000000000000009601, "30+30*i+1"
"i", 100000000000000000326, 1, " =", 3000000000000000009811, "30+30*i+1"
"i", 100000000000000000363, 1, " =", 3000000000000000010921, "30+30*i+1"
"i", 100000000000000000381, 1, " =", 3000000000000000011461, "30+30*i+1"
"i", 100000000000000000395, 1, " =", 3000000000000000011881, "30+30*i+1"
"i", 100000000000000000404, 1, " =", 3000000000000000012151, "30+30*i+1"
"i", 100000000000000000428, 1, " =", 3000000000000000012871, "30+30*i+1"
"i", 100000000000000000454, 1, " =", 3000000000000000013651, "30+30*i+1"
"i", 100000000000000000472, 1, " =", 3000000000000000014191, "30+30*i+1"
"i", 100000000000000000491, 1, " =", 3000000000000000014761, "30+30*i+1"
```

Anmerkung:

1. In der *Folge 3* sind 28 Primzahlen CF_p situiert, die alle die Endziffer EZ1 haben.

---

10. 9 Die Bestimmung der Zwillinge mod 2, die in der *Folge 3* situiert sind:

- for i from 10^20 to 10^20+500 do: if (isprime (30+30*i-1) and isprime (30+30*i+1)) then print(("i", i, -1, " =", 30+30*i-1, "30+30*i+1")) end_if: end_for:

```
"i", 1000000000000000000395, -1, " =", 3000000000000000011879, "30+30*i+1"
```

Anmerkung:

1. In der *Folge 3* ist (nur) ein Zwilling mod 2 situiert.

Die Schlussfolgerungen:

1. Die Primzahlen CM_p, CF_p und die aus ihnen fallweise kreierten Zwillinge mod 2 treten in (jeweils) drei Ordnungs-Folgen auf.

2. Diese (insgesamt) neun Ordnungs-Folgen können in Intervall-Abschnitte gegliedert werden und damit der Daten-Verschlüsselung dienen. Diese Aufgabe erfordert eine durchgehende Bestimmung dieser Primzahlen und der aus ihnen kreierten Zwillinge mod 2.

Der Nachweis für die Gültigkeit des Asymptote-Gesetzes für die Zwillinge mod 2:

1. Die Zahl der Zwillinge mod 2 im Intervall i:= 0 bis 1000:

-
```
for i from 0 to 1000 do: if (isprime (i-1) and isprime (i+1))
  then print (("i", i-1, " = ", i+1)) end_if: end_for:
```

```
"i", 3, " = ", 5
"i", 5, " = ", 7
"i", 11, " = ", 13
"i", 17, " = ", 19
"i", 29, " = ", 31
"i", 41, " = ", 43
"i", 59, " = ", 61
"i", 71, " = ", 73
"i", 101, " = ", 103
"i", 107, " = ", 109
"i", 137, " = ", 139
"i", 149, " = ", 151
"i", 179, " = ", 181
"i", 191, " = ", 193
"i", 197, " = ", 199
"i", 227, " = ", 229
"i", 239, " = ", 241
"i", 269, " = ", 271
"i", 281, " = ", 283
"i", 311, " = ", 313
"i", 347, " = ", 349
"i", 419, " = ", 421
"i", 431, " = ", 433
"i", 461, " = ", 463
"i", 521, " = ", 523
```

```
"i", 569, " = ", 571
"i", 599, " = ", 601
"i", 617, " = ", 619
"i", 641, " = ", 643
"i", 659, " = ", 661
"i", 809, " = ", 811
"i", 821, " = ", 823
"i", 827, " = ", 829
"i", 857, " = ", 859
"i", 881, " = ", 883
```

Anmerkung:

1. In diesem Intervall 35 Zwillinge mod 2 situiert.

---

2. Die Zahl der Zwillinge mod 2 im Intervall i:= 1000 bis 2000:

- ```
  for i from 1000 to 2000 do: if (isprime (i-1) and isprime (i+1))
     then print (("i", i-1, " = ", i+1)) end_if: end_for:
  ```

```
"i", 1019, " = ", 1021
"i", 1031, " = ", 1033
"i", 1049, " = ", 1051
"i", 1061, " = ", 1063
"i", 1091, " = ", 1093
"i", 1151, " = ", 1153
"i", 1229, " = ", 1231
"i", 1277, " = ", 1279
"i", 1289, " = ", 1291
"i", 1301, " = ", 1303
"i", 1319, " = ", 1321
"i", 1427, " = ", 1429
"i", 1451, " = ", 1453
"i", 1481, " = ", 1483
"i", 1487, " = ", 1489
"i", 1607, " = ", 1609
"i", 1619, " = ", 1621
"i", 1667, " = ", 1669
"i", 1697, " = ", 1699
"i", 1721, " = ", 1723
"i", 1787, " = ", 1789
"i", 1871, " = ", 1873
"i", 1877, " = ", 1879
"i", 1931, " = ", 1933
"i", 1949, " = ", 1951
```
```

```
"i", 1997, " = ", 1999
```

Anmerkung:

1. In diesem Intervall sind 26 Zwillinge mod 2 situiert.

---

3. Die Zahl der Zwillinge mod 2 im Intervall i:= 2000 bis 3000:

- for i from 2000 to 3000 do: if (isprime (i-1) and isprime (i+1)) then print (("i", i-1, " = ", i+1)) end_if: end_for:

```
"i", 2027, " = ", 2029
"i", 2081, " = ", 2083
"i", 2087, " = ", 2089
"i", 2111, " = ", 2113
"i", 2129, " = ", 2131
"i", 2141, " = ", 2143
"i", 2237, " = ", 2239
"i", 2267, " = ", 2269
"i", 2309, " = ", 2311
"i", 2339, " = ", 2341
"i", 2381, " = ", 2383
"i", 2549, " = ", 2551
"i", 2591, " = ", 2593
"i", 2657, " = ", 2659
"i", 2687, " = ", 2689
"i", 2711, " = ", 2713
"i", 2729, " = ", 2731
"i", 2789, " = ", 2791
"i", 2801, " = ", 2803
"i", 2969, " = ", 2971
"i", 2999, " = ", 3001
```

Anmerkung:

1. In diesem Intervall sind 21 Zwillinge mod 2 situiert.

---

4. Die Zahl der Zwillinge mod 2 im Intervall i:= 3000 bis 4000:

for i from 3000 to 4000 do: if (isprime (i-1) and isprime (i+1)) then print (("i", i-1, " = ", i+1)) end_if: end_for:

```
"i", 2999, " = ", 3001
"i", 3119, " = ", 3121
"i", 3167, " = ", 3169
"i", 3251, " = ", 3253
"i", 3257, " = ", 3259
"i", 3299, " = ", 3301
```

```
"i"; 3329, " = ", 3331
"i", 3359, " = ", 3361
"i", 3371, " = ", 3373
"i", 3389, " = ", 3391
"i", 3461, " = ", 3463
"i", 3467, " = ", 3469
"i", 3527, " = ", 3529
"i", 3539, " = ", 3541
"i", 3557, " = ", 3559
"i", 3581, " = ", 3583
"i", 3671, " = ", 3673
"i", 3767, " = ", 3769
"i", 3821, " = ", 3823
"i", 3851, " = ", 3853
"i", 3917, " = ", 3919
"i", 3929, " = ", 3931
```

Anmerkung:

1. In diesem Intervall sind 22 Zwillinge mod 2 situiert.

---

5. Die Zahl der Zwillinge mod 2 im Intervall i:= 10^100 bis 10^100+40000:

```
• for i from 10^100 to 10^100+1000 do: if (isprime (i-1) and isprime (i+1))
      then print (("i", i-1, " = ", i+1)) end_if: end_for:
• for i from 10^100 to 10^100+10000 do: if (isprime (i-1) and isprime (i+1))
      then print (("i", i-1, " = ", i+1)) end_if: end_for:
• for i from 10^100 to 10^100+20000 do: if (isprime (i-1) and isprime (i+1))
      then print (("i", i-1, " = ", i+1)) end_if: end_for:
• for i from 10^100+20000 to 10^100+30000 do: if (isprime (i-1) and isprime (i+1))
      then print (("i", i-1, " = ", i+1)) end_if: end_for:
• for i from 10^100+30000 to 10^100+40000 do: if (isprime (i-1) and isprime (i+1))
      then print (("i", i-1, " = ", i+1)) end_if: end_for:
```

```
"i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000035737

, " = ",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000035739
```

Anmerkung:

1. In diesem (großen) Intervall ist nur 1 Zwilling mod 2 situiert.
2. Dieser Zwilling ist im "Unter-Intervall" i:= 10^100+30000 bis 10^100+40000 situiert.

---

6. Die Zahl der Zwillinge mod 2 im Intervall i:= 10^200 bis 10^200 + 400000

```
• for i from 10^200 to 10^200+100000 do: if (isprime (i-1) and isprime (i+1))
      then print (("i", i-1, " = ", i+1)) end_if: end_for:
• for i from 10^200+100000 to 10^200+200000 do: if (isprime (i-1) and isprime (i+1))
      then print (("i", i-1, " = ", i+1)) end_if: end_for:
```

```
"i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000168961

   , " = ",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000168963

"i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000170449

   , " = ",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000170451

"i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000176227

   , " = ",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000176229
```

Anmerkung:

1 Im Intervall i:= 10^200 bis 10^200+100000 treten keine Zwillinge mod 2 auf.

2. Im (gleich großen) Intervall i:= 10^200+100000 bis 10^200 +200000 sind hingegen 3 Zwillinge mod 2 situiert.

3. Dies ist ein Beispiel für die Schwankungen der "Dichten der Zwillinge mod 2", die nach dem Asymptote-Gesetz (wie ersichtlich) stattfinden können.

Die Bestimmung der Primzahlen CM_p und CF_p in großen Zahlenbereichen:

1. Die Primzahlen $CM_p = n_g - 1$ im Intervall i:= 10^100 bis 10^100+2000:

```
• for i from 10^100 to 10^100+2000 do: if isprime (i-1)
     then print ("i", " = ", i-1) end_if: end_for:

"i", " = ",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000267

"i", " = ",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000949

"i", " = ",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000001243

"i", " = ",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000001293
```

```
"i", " = ",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000001983
```

---

2. Die Primzahlen CF_p im Intervall i:= 10^100 bis 10^100+2000:

- ```
  for i from 10^100 to 10^100+2000 do: if isprime (i+1)
      then print ("i", " = ", i+1) end_if: end_for:
  ```

```
"i", " = ",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000267

"i", " = ",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000949

"i", " = ",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000001243

"i", " = ",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000001293

"i", " = ",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000001983
```

<u>Anmerkung:</u>

1. In diesem Intervall sind 5 Primzahlen CM_p und 5 Primtzahlen CF_p situiert

2. Diese Primzahlen bilden jedoch keine Zwillinge mod 2, da kein Paar dieser Primzahlen im Abstand von $CM_p + 2 = CF_p$ vorkommt..

---

3. Die Primzahlen CM_p im Intervall i:= 10^200 bis 10^200+2000:

- ```
  for i from 10^200 to 10^200+2000 do: if isprime (i-1)
      then print ("i", " = ", i-1) end_if: end_for:
  ```

```
"i", " = ",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000357

"i", " = ",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000627

"i", " = ",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000799

"i", " = ",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000001849
```

---

4. Die Primzahlen CF_p im Intervall i:= 10^200 bis 10^200+2000:

-
```
for i from 10^200 to 10^200+2000 do: if isprime (i+1)
   then print ("i", " = ", i+1) end_if: end_for:
```

```
"i", " = ",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000357

"i", " = ",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000627

"i", " = ",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000799

"i", " = ",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000001849
```

Anmerkung:

1. Im Intervall i:= 10^200 bis 10^200+2000 sind jeweils 4 Primzahlen CM_p und CF_p situiert, die jedoch keine Zwillinge mod 2 bilden.

Anlage 6

Die Generierung von Primzahlen vom Typ CM_p, CF_p und der aus ihnen kreierten Zwillinge mod 6 mit dem MuPAD - Testprogramm.

1. Die Primzahlen CM_p der *Folge 1* im Intervall i:= 0 bis 30:

```
• for i from 0 to 30 do: if (isprime (14+10*i-3) and isprime (14+10*i-3))
• then print (("i",-3, 14+10*i-3," = ", "14+10*i", 14+10*i-3)) end_if: end_for:
```

```
"i", -3, 11, " = ", "14+10*i", 11
"i", -3, 31, " = ", "14+10*i", 31
"i", -3, 41, " = ", "14+10*i", 41
"i", -3, 61, " = ", "14+10*i", 61
"i", -3, 71, " = ", "14+10*i", 71
"i", -3, 101, " = ", "14+10*i", 101
"i", -3, 131, " = ", "14+10*i", 131
"i", -3, 151, " = ", "14+10*i", 151
"i", -3, 181, " = ", "14+10*i", 181
"i", -3, 191, " = ", "14+10*i", 191
"i", -3, 211, " = ", "14+10*i", 211
"i", -3, 241, " = ", "14+10*i", 241
"i", -3, 251, " = ", "14+10*i", 251
"i", -3, 271, " = ", "14+10*i", 271
"i", -3, 281, " = ", "14+10*i", 281
"i", -3, 311, " = ", "14+10*i", 311
```

Anmerkung:

1. Im Intervall i:= 0 bis 30 sind in *Folge 1* insgesamt 16 Primzahlen CM_p situiert.
2. Alle Primzahlen CM_p der *Folge 1* haben die Endziffer EZ1

---

2. Die Primzahlen CF_p der *Folge 1* im Intervaall i:= 0 bis 30

```
• for i from 0 to 30 do: if (isprime (14+10*i+3) and isprime (14+10*i+3))
  then print (("i",+3, 14+10*i-3," = ", "14+10*i", 14+10*i+3)) end_if: end_for:
```

```
"i", 3, 11, " = ", "14+10*i", 17
"i", 3, 31, " = ", "14+10*i", 37
"i", 3, 41, " = ", "14+10*i", 47
"i", 3, 61, " = ", "14+10*i", 67
"i", 3, 91, " = ", "14+10*i", 97
"i", 3, 101, " = ", "14+10*i", 107
"i", 3, 121, " = ", "14+10*i", 127
"i", 3, 131, " = ", "14+10*i", 137
"i", 3, 151, " = ", "14+10*i", 157
```

```
"i", 3, 161, " = ", "14+10*i", 167
"i", 3, 191, " = ", "14+10*i", 197
"i", 3, 221, " = ", "14+10*i", 227
"i", 3, 251, " = ", "14+10*i", 257
"i", 3, 271, " = ", "14+10*i", 277
"i", 3, 301, " = ", "14+10*i", 307
"i", 3, 311, " = ", "14+10*i", 317
```

Anmerkung:

1. Im Intervall i:= 0 bis 30 sind in *Folge 1* insgesamt 16 Primzahlen CF_p situiert.

2. Alle Primzahlen CF_p haben die Endziffer EZ7.

---

3. Die Bestimmung der Zwillinge mod 6, die im Intervall i:= 0 bis 30 in der *Folge 1* situiert sind.

-
```
for i from 0 to 30 do: if (isprime (14+10*i-3) and isprime (14+10*i+3))
then print (("i",-3, 14+10*i-3," = ", "14+10*i", 14+10*i+3)) end_if: end_for:
```

```
"i", -3, 11, " = ", "14+10*i", 17
"i", -3, 31, " = ", "14+10*i", 37
"i", -3, 41, " = ", "14+10*i", 47
"i", -3, 61, " = ", "14+10*i", 67
"i", -3, 101, " = ", "14+10*i", 107
"i", -3, 131, " = ", "14+10*i", 137
"i", -3, 151, " = ", "14+10*i", 157
"i", -3, 191, " = ", "14+10*i", 197
"i", -3, 251, " = ", "14+10*i", 257
"i", -3, 271, " = ", "14+10*i", 277
"i", -3, 311, " = ", "14+10*i", 317
```

Anmerkung:

1. In der *Folge 1* sind insgesamt 11 Zwillinge mod 6 situiert.

2. Die Primzahlen (CM_p, CF_p) der Zwillinge mod 6 sind stets Elemente der Obergruppen OG I und OG II

---

4. Die Bestimmung der Primzahlen CM_p, die in der *Folge 2* situiert sind:

-
```
for i from 0 to 30 do: if (isprime (16+10*i-3) and isprime (16+10*i-3))
then print (("i",-3, 16+10*i-3," = ", "16+10*i", 16+10*i-3)) end_if: end_for:
```

```
"i", -3, 13, " = ", "16+10*i", 13
"i", -3, 23, " = ", "16+10*i", 23
"i", -3, 43, " = ", "16+10*i", 43
"i", -3, 53, " = ", "16+10*i", 53
"i", -3, 73, " = ", "16+10*i", 73
"i", -3, 83, " = ", "16+10*i", 83
"i", -3, 103, " = ", "16+10*i", 103
```

```
"i", -3, 113, " = ", "16+10*i", 113
"i", -3, 163, " = ", "16+10*i", 163
"i", -3, 173, " = ", "16+10*i", 173
"i", -3, 193, " = ", "16+10*i", 193
"i", -3, 223, " = ", "16+10*i", 223
"i", -3, 233, " = ", "16+10*i", 233
"i", -3, 263, " = ", "16+10*i", 263
"i", -3, 283, " = ", "16+10*i", 283
"i", -3, 293, " = ", "16+10*i", 293
"i", -3, 313, " = ", "16+10*i", 313
```

Anmerkung:

1. In der *Folge 2* sind insgesamt 17 Primzahlen CM_p situiert.
2. Alle Primzahlen CM_p haben die Endziffer EZ3

---

5. Die Bestimmung der Primzahlen CF_p, die in der *Folge 2* situiert sind.

-
```
for i from 0 to 30 do: if (isprime (16+10*i+3) and isprime (16+10*i+3))
then print (("i",+3, 16+10*i+3," = ", "16+10*i", 16+10*i+3)) end_if: end_for:
```

```
"i", 3, 19, " = ", "16+10*i", 19
"i", 3, 29, " = ", "16+10*i", 29
"i", 3, 59, " = ", "16+10*i", 59
"i", 3, 79, " = ", "16+10*i", 79
"i", 3, 89, " = ", "16+10*i", 89
"i", 3, 109, " = ", "16+10*i", 109
"i", 3, 139, " = ", "16+10*i", 139
"i", 3, 149, " = ", "16+10*i", 149
"i", 3, 179, " = ", "16+10*i", 179
"i", 3, 199, " = ", "16+10*i", 199
"i", 3, 229, " = ", "16+10*i", 229
"i", 3, 239, " = ", "16+10*i", 239
"i", 3, 269, " = ", "16+10*i", 269
```

Anmerkung:

1. In der *Folge 2* sind insgesamt 13 Primzahlen CF_p situiert.
2. Alle Primzahlen CF_p haben die Endziffer EZ9

---

6. Die Bestimmung der Zwillinge mod 6, die in der *Folge 2* situiert sind:

-
```
for i from 0 to 30 do: if (isprime (16+10*i-3) and isprime (16+10*i+3))
then print (("i",-3, 16+10*i-3," = ", "16+10*i", 16+10*i+3)) end_if: end_for:
```

```
"i", -3, 13, " = ", "16+10*i", 19
"i", -3, 23, " = ", "16+10*i", 29
"i", -3, 53, " = ", "16+10*i", 59
```

```
"i", -3, 73, " = ", "16+10*i", 79
"i", -3, 83, " = ", "16+10*i", 89
"i", -3, 103, " = ", "16+10*i", 109
"i", -3, 173, " = ", "16+10*i", 179
"i", -3, 193, " = ", "16+10*i", 199
"i", -3, 223, " = ", "16+10*i", 229
"i", -3, 233, " = ", "16+10*i", 239
"i", -3, 263, " = ", "16+10*i", 269
```

Anmerkung:

1. In der *Folge 2* sind insgesamt 11 Zwillinge mod 6 situiert.
2. Die Primzahlen aller Zwillinge mod 6 sind stets Elemente der Obergruppen OG I und OG II.

---

7. Die Bestimmung der Primzahlen CM_p, die in der *Folge 3* situiert sind:

-
```
for i from 0 to 30 do: if (isprime (10+10*i-3) and isprime (10+10*i-3))
then print (("i",-3, 10+10*i-3," = ", "10+10*i", 10+10*i-3)) end_if: end_for:
```

```
"i", -3, 7, " = ", "10+10*i", 7
"i", -3, 17, " = ", "10+10*i", 17
"i", -3, 37, " = ", "10+10*i", 37
"i", -3, 47, " = ", "10+10*i", 47
"i", -3, 67, " = ", "10+10*i", 67
"i", -3, 97, " = ", "10+10*i", 97
"i", -3, 107, " = ", "10+10*i", 107
"i", -3, 127, " = ", "10+10*i", 127
"i", -3, 137, " = ", "10+10*i", 137
"i", -3, 157, " = ", "10+10*i", 157
"i", -3, 167, " = ", "10+10*i", 167
"i", -3, 197, " = ", "10+10*i", 197
"i", -3, 227, " = ", "10+10*i", 227
"i", -3, 257, " = ", "10+10*i", 257
"i", -3, 277, " = ", "10+10*i", 277
"i", -3, 307, " = ", "10+10*i", 307
```

Anmerkung:

1. In der *Folge 3* sind insgesamt 13 Primzahlen CM_p situiert.
2. Alle Primzahlen CM_p haben die Endziffer EZ7.

---

8. Die Bestimmung der Primzahlen CF_p, die in der *Folge 3* situiert sind:

-
```
for i from 0 to 30 do: if (isprime (10+10*i+3) and isprime (10+10*i+3))
then print (("i",+3, 10+10*i+3," = ", "10+10*i", 10+10*i+3)) end_if: end_for:
```

```
"i", 3, 13, " = ", "10+10*i", 13
"i", 3, 23, " = ", "10+10*i", 23
```

```
"i", 3, 43, " = ", "10+10*i", 43
"i", 3, 53, " = ", "10+10*i", 53
"i", 3, 73, " = ", "10+10*i", 73
"i", 3, 83, " = ", "10+10*i", 83
"i", 3, 103, " = ", "10+10*i", 103
"i", 3, 113, " = ", "10+10*i", 113
"i", 3, 163, " = ", "10+10*i", 163
"i", 3, 173, " = ", "10+10*i", 173
"i", 3, 193, " = ", "10+10*i", 193
"i", 3, 223, " = ", "10+10*i", 223
"i", 3, 233, " = ", "10+10*i", 233
"i", 3, 263, " = ", "10+10*i", 263
"i", 3, 283, " = ", "10+10*i", 283
"i", 3, 293, " = ", "10+10*i", 293
"i", 3, 313, " = ", "10+10*i", 313
```

Anmerkung:

1. In der *Folge 3* sind insgesamt 17 Primzahlen CF_p situiert.

2. Alle Primzahlen CF_p haben die Endziffer EZ3

---

9. Die Bestimmung der Zwillinge mod 6, die in der *Folge 3* situiert sind:

-
```
for i from 0 to 30 do: if (isprime (10+10*i-3) and isprime (10+10*i+3))
then print (("i",-3, 10+10*i-3," = ", "10+10*i", 10+10*i+3)) end_if: end_for:
```

```
"i", -3, 7, " = ", "10+10*i", 13
"i", -3, 17, " = ", "10+10*i", 23
"i", -3, 37, " = ", "10+10*i", 43
"i", -3, 47, " = ", "10+10*i", 53
"i", -3, 67, " = ", "10+10*i", 73
"i", -3, 97, " = ", "10+10*i", 103
"i", -3, 107, " = ", "10+10*i", 113
"i", -3, 157, " = ", "10+10*i", 163
"i", -3, 167, " = ", "10+10*i", 173
"i", -3, 227, " = ", "10+10*i", 233
"i", -3, 257, " = ", "10+10*i", 263
"i", -3, 277, " = ", "10+10*i", 283
"i", -3, 307, " = ", "10+10*i", 313
```

Anmerkung:

1. In der *Folge 3* sind insgesamt 13 Zwillinge mod 6 situiert

---

Die Bestimmung der Primzahlen CM_p und CF_p sowie der Zwillinge mod 6, die in *Folge 1*, *Folge 2* und *Folge 3* im Intervall i:= 10^350 bis 10^350 +3000 situiert sind:

1. Die Bestimmung der Primzahlen CM_p, die in der *Folge 1* situiert sind:

- `for i from 10^350 to 10^350+2000 do: if (isprime (14+10*i-3) and isprime (14+10*i-3)) then print (("i", -3, " = ", "14+10*i", 14+10*i-3)) end_if: end_for:`

```
"i", -3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000501

"i", -3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000008601

"i", -3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000009891

"i", -3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000011721

"i", -3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000011931

"i", -3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012001

"i", -3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012021

"i", -3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012301

"i", -3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012621
```

<u>Anmerkung:</u>

1. In der *Folge 1* sind insgesamt 9 Primzahlen CM_p situiert, die allesamt die Endziffer EZ1 haben.

---

2. Die Bestimmung der Primzahlen CF_P, die in der *Folge 1* situiert sind:

- ```
for i from 10^350 to 10^350+2000 do: if (isprime (14+10*i+3) and isprime (14+10*i+3))
    then print (("i", +3, " = ", "14+10*i", 14+10*i+3)) end_if: end_for:
```

```
"i", 3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000207

"i", 3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000001447

"i", 3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000005977

"i", 3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000006687

"i", 3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000009237

"i", 3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000013507

"i", 3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000018067

"i", 3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000019777
```

<u>Anmerkung:</u>

1. In der *Folge 1* sind insgesamt 8 Primzahlen CF_P situiert, die alle die Endziffer EZ7 haben.

---

3. Die Bestimmung der Zwillinge mod 6, die im erweiterten Intervall in der *Folge 1* situiert sind:

- ```
for i from 10^350 to 10^350+14000 do: if (isprime (14+10*i-3) and isprime
(14+10*i+3))then print (("i", -3, 14+10*i-3, " = ", "14+10*i", 14+10*i+3)) end_if: end_for:
```

```
"i", -3,
```

```
1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000137641

, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000137647
```

<u>Anmerkung:</u>

1. Im erweiterten Intervall i:= 10^350 bis 10^350 +14000 ist in der *Folge 1* ein Zwilling mod 6 situiert.

---

4. Die Bestimmung der in der *Folge 2* situierten Primzahlen CM_p:

- ```
  for i from 10^350 to 10^350+2000 do: if (isprime (16+10*i-3) and isprime (16+10*i-3))
      then print (("i", -3, " = ", "16+10*i", 16+10*i-3)) end_if: end_for:
  ```

```
"i", -3, " = ", "16+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000004953

"i", -3, " = ", "16+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000007413

"i", -3, " = ", "16+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000010183

"i", -3, " = ", "16+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000015203
```

<u>Anmerkung:</u>

1. In der *Folge 2* sind insgesamt 4 Primzahlen CM_p situiert, die allesamt die Endziffer EZ3 haben.

---

5. Die Bestimmung der in der *Folge 2* situierten Primzahlen CF_p:

- ```
  for i from 10^350 to 10^350+2000 do: if (isprime (16+10*i+3) and isprime (16+10*i+3))
      then print (("i", +3, " = ", "16+10*i", 16+10*i+3)) end_if: end_for:
  ```

```
"i", 3, " = ", "16+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000249

"i", 3, " = ", "16+10*i",
```

```
1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000004879

"i", 3, " = ", "16+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000008059

"i", 3, " = ", "16+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000010119

"i", 3, " = ", "16+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000010939

"i", 3, " = ", "16+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012549

"i", 3, " = ", "16+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000014809
```

<u>Anmerkungen:</u>

1. In der *Folge 2* sind insgesamt 7 Primzahlen CF_p situiert, die allesamt die Endziffer EZ9 haben.

---

6. Die Bestimmung der in der *Folge 2* situierten Zwillinge mod 6:
<u>Anmerkung:</u>
In der *Folge 2* ist im Intervall i:= 10^350 bis 10^350 +2000 kein Zwilling mod 6 situiert und auch nicht im erweiterten Intervall i:= 10^350 bis 10^350 +100000.

---

7. Die Bestimmung der in der *Folge 3* situierten Primzahlen CM_p:

- ```
  for i from 10^350 to 10^350+2000 do: if (isprime (10+10*i-3) and isprime (10+10*i-3))
      then print (("i", -3, " = ", "10+10*i", 10+10*i-3)) end_if: end_for:
  ```

```
"i", -3, " = ", "10+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000207

"i", -3, " = ", "10+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000001447
```
```

```
"i", -3, " = ", "10+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000005977

"i", -3, " = ", "10+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000006687

"i", -3, " = ", "10+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000009237

"i", -3, " = ", "10+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000013507

"i", -3, " = ", "10+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000018067

"i", -3, " = ", "10+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000019777
```

<u>Anmerkung:</u>

1. In der *Folge 3* sind insgesamt 8 Primzahlen CM_p situiert.

2. Alle Primzahlen CM_p haben die Endziffer EZ7

---

8. Die Bestimmung der in der *Folge 3* situierten Primzahlen CF_p:

- ```
  for i from 10^350 to 10^350+2000 do: if (isprime (10+10*i+3) and isprime (10+10*i+3))
      then print (("i", -3, " = ", "10+10*i", 10+10*i+3)) end_if: end_for:
  ```

```
"i", -3, " = ", "10+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000004953

"i", -3, " = ", "10+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000007413

"i", -3, " = ", "10+10*i",
```
```

```
1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000010183

"i", -3, " = ", "10+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000015903
```

<u>Anmerkung:</u>

1. In der *Folge 3* sind insgesamt 4 Primzahlen CF_p situiert.

2. Alle Primzahlen CF_p haben die Endziffer EZ3.

---

9. Die Bestimmung der in der *Folge 3* situierten Zwillinge mod 6:

- ```
  for i from 10^350+8000 to 10^350+10000 do: if (isprime (10+10*i-3) and isprime
  (10+10*i+3)) then print (("i", -3, 10+10*i-3, " = ", "10+10*i", 10+10*i+3)) end_if:
  end_for:
  ```

```
"i", -3,

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000083607

  , " = ", "10+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000083613
```

<u>Anmerkung:</u>

1. In der *Folge 3* ist im Intervall i:= 10^350+8000 bis 10^350+10000 ein Zwilling mod 6 situiert.

2. In diesem Zwilling treten die Primzahlen (CM_p, CF_p) in den Endziffern EZ7, EZ3 auf.

3. Die Primzahl CM_p ist ein Element der Obergruppe OG II und die Primzahl CF_p ein Element der Obergruppe OG I.

<u>Schlussfolgerungen:</u>

1. Das Asymptote-Gesetz wird bestätigt, da in dem wesentlich größeren Intervall i:= 10^350 bis 10^350 +2000 wesentlich weniger Primzahlen CM_p und CF_p auftreten, als im Intervall i:= 0 bis 30.

2. Das Asymptote-Gesetz wird, wie ersichtlich auch für die Zwillinge mod 6 erfüllt.

```

Anlage 6

Die Generierung von Primzahlen vom Typ CM_p, CF_p und der aus ihnen kreierten Zwillinge mod 6 mit dem MuPAD - Testprogramm.

1. Die Primzahlen CM_p der *Folge 1* im Intervall i:= 0 bis 30:

```
• for i from 0 to 30 do: if (isprime (14+10*i-3) and isprime (14+10*i-3))
• then print (("i",-3, 14+10*i-3," = ", "14+10*i", 14+10*i-3)) end_if: end_for:
```

```
"i", -3, 11, " = ", "14+10*i", 11
"i", -3, 31, " = ", "14+10*i", 31
"i", -3, 41, " = ", "14+10*i", 41
"i", -3, 61, " = ", "14+10*i", 61
"i", -3, 71, " = ", "14+10*i", 71
"i", -3, 101, " = ", "14+10*i", 101
"i", -3, 131, " = ", "14+10*i", 131
"i", -3, 151, " = ", "14+10*i", 151
"i", -3, 181, " = ", "14+10*i", 181
"i", -3, 191, " = ", "14+10*i", 191
"i", -3, 211, " = ", "14+10*i", 211
"i", -3, 241, " = ", "14+10*i", 241
"i", -3, 251, " = ", "14+10*i", 251
"i", -3, 271, " = ", "14+10*i", 271
"i", -3, 281, " = ", "14+10*i", 281
"i", -3, 311, " = ", "14+10*i", 311
```

Anmerkung:

1. Im Intervall i:= 0 bis 30 sind in *Folge 1* insgesamt 16 Primzahlen CM_p situiert.
2. Alle Primzahlen CM_p der *Folge 1* haben die Endziffer EZ1

---

2. Die Primzahlen CF_p der *Folge 1* im Intervaall i:= 0 bis 30

```
• for i from 0 to 30 do: if (isprime (14+10*i+3) and isprime (14+10*i+3))
  then print (("i",+3, 14+10*i-3," = ", "14+10*i", 14+10*i+3)) end_if: end_for:
```

```
"i", 3, 11, " = ", "14+10*i", 17
"i", 3, 31, " = ", "14+10*i", 37
"i", 3, 41, " = ", "14+10*i", 47
"i", 3, 61, " = ", "14+10*i", 67
"i", 3, 91, " = ", "14+10*i", 97
"i", 3, 101, " = ", "14+10*i", 107
"i", 3, 121, " = ", "14+10*i", 127
"i", 3, 131, " = ", "14+10*i", 137
"i", 3, 151, " = ", "14+10*i", 157
```

```
"i", 3, 161, " = ", "14+10*i", 167
"i", 3, 191, " = ", "14+10*i", 197
"i", 3, 221, " = ", "14+10*i", 227
"i", 3, 251, " = ", "14+10*i", 257
"i", 3, 271, " = ", "14+10*i", 277
"i", 3, 301, " = ", "14+10*i", 307
"i", 3, 311, " = ", "14+10*i", 317
```

Anmerkung:

1. Im Intervall i:= 0 bis 30 sind in *Folge 1* insgesamt 16 Primzahlen CF_p situiert.

2. Alle Primzahlen CF_p haben die Endziffer EZ7.

---

3. Die Bestimmung der Zwillinge mod 6, die im Intervall i:= 0 bis 30 in der *Folge 1* situiert sind.

-
```
for i from 0 to 30 do: if (isprime (14+10*i-3) and isprime (14+10*i+3))
then print (("i",-3, 14+10*i-3," = ", "14+10*i", 14+10*i+3)) end_if: end_for:
```

```
"i", -3, 11, " = ", "14+10*i", 17
"i", -3, 31, " = ", "14+10*i", 37
"i", -3, 41, " = ", "14+10*i", 47
"i", -3, 61, " = ", "14+10*i", 67
"i", -3, 101, " = ", "14+10*i", 107
"i", -3, 131, " = ", "14+10*i", 137
"i", -3, 151, " = ", "14+10*i", 157
"i", -3, 191, " = ", "14+10*i", 197
"i", -3, 251, " = ", "14+10*i", 257
"i", -3, 271, " = ", "14+10*i", 277
"i", -3, 311, " = ", "14+10*i", 317
```

Anmerkung:

1. In der *Folge 1* sind insgesamt 11 Zwillinge mod 6 situiert.

2. Die Primzahlen (CM_p, CF_p) der Zwillinge mod 6 sind stets Elemente der Obergruppen OG I und OG II

---

4. Die Bestimmung der Primzahlen CM_p, die in der *Folge 2* situiert sind:

-
```
for i from 0 to 30 do: if (isprime (16+10*i-3) and isprime (16+10*i-3))
then print (("i",-3, 16+10*i-3," = ", "16+10*i", 16+10*i-3)) end_if: end_for:
```

```
"i", -3, 13, " = ", "16+10*i", 13
"i", -3, 23, " = ", "16+10*i", 23
"i", -3, 43, " = ", "16+10*i", 43
"i", -3, 53, " = ", "16+10*i", 53
"i", -3, 73, " = ", "16+10*i", 73
"i", -3, 83, " = ", "16+10*i", 83
"i", -3, 103, " = ", "16+10*i", 103
```

```
"i", -3, 113, " = ", "16+10*i", 113
"i", -3, 163, " = ", "16+10*i", 163
"i", -3, 173, " = ", "16+10*i", 173
"i", -3, 193, " = ", "16+10*i", 193
"i", -3, 223, " = ", "16+10*i", 223
"i", -3, 233, " = ", "16+10*i", 233
"i", -3, 263, " = ", "16+10*i", 263
"i", -3, 283, " = ", "16+10*i", 283
"i", -3, 293, " = ", "16+10*i", 293
"i", -3, 313, " = ", "16+10*i", 313
```

Anmerkung:

1. In der *Folge 2* sind insgesamt 17 Primzahlen CM_p situiert.
2. Alle Primzahlen CM_p haben die Endziffer EZ3

---

5. Die Bestimmung der Primzahlen CF_p, die in der *Folge 2* situiert sind.

- ```
  for i from 0 to 30 do: if (isprime (16+10*i+3) and isprime (16+10*i+3))
  then print (("i",+3, 16+10*i+3," = ", "16+10*i", 16+10*i+3)) end_if: end_for:
  ```

```
"i", 3, 19, " = ", "16+10*i", 19
"i", 3, 29, " = ", "16+10*i", 29
"i", 3, 59, " = ", "16+10*i", 59
"i", 3, 79, " = ", "16+10*i", 79
"i", 3, 89, " = ", "16+10*i", 89
"i", 3, 109, " = ", "16+10*i", 109
"i", 3, 139, " = ", "16+10*i", 139
"i", 3, 149, " = ", "16+10*i", 149
"i", 3, 179, " = ", "16+10*i", 179
"i", 3, 199, " = ", "16+10*i", 199
"i", 3, 229, " = ", "16+10*i", 229
"i", 3, 239, " = ", "16+10*i", 239
"i", 3, 269, " = ", "16+10*i", 269
```

Anmerkung:

1. In der *Folge 2* sind insgesamt 13 Primzahlen CF_p situiert.
2. Alle Primzahlen CF_p haben die Endziffer EZ9

---

6. Die Bestimmung der Zwillinge mod 6, die in der *Folge 2* situiert sind:

- ```
  for i from 0 to 30 do: if (isprime (16+10*i-3) and isprime (16+10*i+3))
  then print (("i",-3, 16+10*i-3," = ", "16+10*i", 16+10*i+3)) end_if: end_for:
  ```

```
"i", -3, 13, " = ", "16+10*i", 19
"i", -3, 23, " = ", "16+10*i", 29
"i", -3, 53, " = ", "16+10*i", 59
```

```
"i", -3, 73, " = ", "16+10*i", 79
"i", -3, 83, " = ", "16+10*i", 89
"i", -3, 103, " = ", "16+10*i", 109
"i", -3, 173, " = ", "16+10*i", 179
"i", -3, 193, " = ", "16+10*i", 199
"i", -3, 223, " = ", "16+10*i", 229
"i", -3, 233, " = ", "16+10*i", 239
"i", -3, 263, " = ", "16+10*i", 269
```

Anmerkung:

1. In der *Folge 2* sind insgesamt 11 Zwillinge mod 6 situiert.

2. Die Primzahlen aller Zwillinge mod 6 sind stets Elemente der Obergruppen OG I und OG II.

---

7. Die Bestimmung der Primzahlen CM_p, die in der *Folge 3* situiert sind:

-
```
for i from 0 to 30 do: if (isprime (10+10*i-3) and isprime (10+10*i-3))
then print (("i",-3, 10+10*i-3," = ", "10+10*i", 10+10*i-3)) end_if: end_for:
```

```
"i", -3, 7, " = ", "10+10*i", 7
"i", -3, 17, " = ", "10+10*i", 17
"i", -3, 37, " = ", "10+10*i", 37
"i", -3, 47, " = ", "10+10*i", 47
"i", -3, 67, " = ", "10+10*i", 67
"i", -3, 97, " = ", "10+10*i", 97
"i", -3, 107, " = ", "10+10*i", 107
"i", -3, 127, " = ", "10+10*i", 127
"i", -3, 137, " = ", "10+10*i", 137
"i", -3, 157, " = ", "10+10*i", 157
"i", -3, 167, " = ", "10+10*i", 167
"i", -3, 197, " = ", "10+10*i", 197
"i", -3, 227, " = ", "10+10*i", 227
"i", -3, 257, " = ", "10+10*i", 257
"i", -3, 277, " = ", "10+10*i", 277
"i", -3, 307, " = ", "10+10*i", 307
```

Anmerkung:

1. In der *Folge 3* sind insgesamt 13 Primzahlen CM_p situiert.

2. Alle Primzahlen CM_p haben die Endziffer EZ7.

---

8. Die Bestimmung der Primzahlen CF_p, die in der *Folge 3* situiert sind:

-
```
for i from 0 to 30 do: if (isprime (10+10*i+3) and isprime (10+10*i+3))
then print (("i",+3, 10+10*i+3," = ", "10+10*i", 10+10*i+3)) end_if: end_for:
```

```
"i", 3, 13, " = ", "10+10*i", 13
"i", 3, 23, " = ", "10+10*i", 23
```

```
"i", 3, 43, " = ", "10+10*i", 43
"i", 3, 53, " = ", "10+10*i", 53
"i", 3, 73, " = ", "10+10*i", 73
"i", 3, 83, " = ", "10+10*i", 83
"i", 3, 103, " = ", "10+10*i", 103
"i", 3, 113, " = ", "10+10*i", 113
"i", 3, 163, " = ", "10+10*i", 163
"i", 3, 173, " = ", "10+10*i", 173
"i", 3, 193, " = ", "10+10*i", 193
"i", 3, 223, " = ", "10+10*i", 223
"i", 3, 233, " = ", "10+10*i", 233
"i", 3, 263, " = ", "10+10*i", 263
"i", 3, 283, " = ", "10+10*i", 283
"i", 3, 293, " = ", "10+10*i", 293
"i", 3, 313, " = ", "10+10*i", 313
```

Anmerkung:

1. In der *Folge 3* sind insgesamt 17 Primzahlen CF_p situiert.

2. Alle Primzahlen CF_p haben die Endziffer EZ3

---

9. Die Bestimmung der Zwillinge mod 6, die in der *Folge 3* situiert sind:

- ```
  for i from 0 to 30 do: if (isprime (10+10*i-3) and isprime (10+10*i+3))
  then print (("i",-3, 10+10*i-3," = ", "10+10*i", 10+10*i+3)) end_if: end_for:
  ```

```
"i", -3, 7, " = ", "10+10*i", 13
"i", -3, 17, " = ", "10+10*i", 23
"i", -3, 37, " = ", "10+10*i", 43
"i", -3, 47, " = ", "10+10*i", 53
"i", -3, 67, " = ", "10+10*i", 73
"i", -3, 97, " = ", "10+10*i", 103
"i", -3, 107, " = ", "10+10*i", 113
"i", -3, 157, " = ", "10+10*i", 163
"i", -3, 167, " = ", "10+10*i", 173
"i", -3, 227, " = ", "10+10*i", 233
"i", -3, 257, " = ", "10+10*i", 263
"i", -3, 277, " = ", "10+10*i", 283
"i", -3, 307, " = ", "10+10*i", 313
```

Anmerkung:

1. In der *Folge 3* sind insgesamt 13 Zwillinge mod 6 situiert

---

Die Bestimmung der Primzahlen CM_p und CF_p sowie der Zwillinge mod 6, die in *Folge 1*, *Folge 2* und *Folge 3* im Intervall i:= 10^350 bis 10^350 +3000 situiert sind:

1. Die Bestimmung der Primzahlen CM_p, die in der *Folge 1* situiert sind:
```

- ```
  for i from 10^350 to 10^350+2000 do: if (isprime (14+10*i-3) and isprime (14+10*i-3))
      then print (("i", -3, " = ", "14+10*i", 14+10*i-3)) end_if: end_for:
  ```

```
"i", -3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000501

"i", -3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000008601

"i", -3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000009891

"i", -3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000011721

"i", -3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000011931

"i", -3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012001

"i", -3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012021

"i", -3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012301

"i", -3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012621
```

<u>Anmerkung:</u>

1. In der *Folge 1* sind insgesamt 9 Primzahlen CM_p situiert, die allesamt die Endziffer EZ1 haben.

```

2. Die Bestimmung der Primzahlen CF_p, die in der *Folge 1* situiert sind:

- ```
  for i from 10^350 to 10^350+2000 do: if (isprime (14+10*i+3) and isprime (14+10*i+3))
      then print (("i", +3, " = ", "14+10*i", 14+10*i+3)) end_if: end_for:
  ```

```
"i", 3, " = ", "14+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000207

"i", 3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000001447

"i", 3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000005977

"i", 3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000006687

"i", 3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000009237

"i", 3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000013507

"i", 3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000018067

"i", 3, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000019777
```

<u>Anmerkung:</u>

1. In der *Folge 1* sind insgesamt 8 Primzahlen CF_p situiert, die alle die Endziffer EZ7 haben.

---

3. Die Bestimmung der Zwillinge mod 6, die im erweiterten Intervall in der *Folge 1* situiert sind:

- ```
  for i from 10^350 to 10^350+14000 do: if (isprime (14+10*i-3) and isprime
  (14+10*i+3))then print (("i", -3, 14+10*i-3, " = ", "14+10*i", 14+10*i+3)) end_if: end_for:
  ```

```
"i", -3,
```

```
1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000137641

, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000137647
```

Anmerkung:

1. Im erweiterten Intervall i:= 10^350 bis 10^350 +14000 ist in der *Folge 1* ein Zwilling mod 6 situiert.

---

4. Die Bestimmung der in der *Folge 2* situierten Primzahlen CM_p:

- ```
  for i from 10^350 to 10^350+2000 do: if (isprime (16+10*i-3) and isprime (16+10*i-3))
     then print (("i", -3, " = ", "16+10*i", 16+10*i-3)) end_if: end_for:
  ```

```
"i", -3, " = ", "16+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000004953

"i", -3, " = ", "16+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000007413

"i", -3, " = ", "16+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000010183

"i", -3, " = ", "16+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000015903
```

Anmerkung:

1. In der *Folge 2* sind insgesamt 4 Primzahlen CM_p situiert, die allesamt die Endziffer EZ3 haben.

---

5. Die Bestimmung der in der *Folge 2* situierten Primzahlen CF_p:

- ```
  for i from 10^350 to 10^350+2000 do: if (isprime (16+10*i+3) and isprime (16+10*i+3))
     then print (("i", +3, " = ", "16+10*i", 16+10*i+3)) end_if: end_for:
  ```

```
"i", 3, " = ", "16+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000249

"i", 3, " = ", "16+10*i",
```

```
1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000004879

"i", 3, " = ", "16+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000008059

"i", 3, " = ", "16+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000010119

"i", 3, " = ", "16+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000010939

"i", 3, " = ", "16+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012549

"i", 3, " = ", "16+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000014809
```

Anmerkungen:

1. In der *Folge 2* sind insgesamt 7 Primzahlen CF_p situiert, die allesamt die Endziffer EZ9 haben.

---

6. Die Bestimmung der in der *Folge 2* situierten Zwillinge mod 6:
   Anmerkung:
   In der *Folge 2* ist im Intervall i:= 10^350 bis 10^350 +2000 kein Zwilling mod 6 situiert und auch nicht im erweiterten Intervall i:= 10^350 bis 10^350 +100000.

---

7. Die Bestimmung der in der *Folge 3* situierten Primzahlen CM_p:

- `for i from 10^350 to 10^350+2000 do: if (isprime (10+10*i-3) and isprime (10+10*i-3)) then print (("i", -3, " = ", "10+10*i", 10+10*i-3)) end_if: end_for:`

```
"i", -3, " = ", "10+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000207

"i", -3, " = ", "10+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000001447
```

```
"i", -3, " = ", "10+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000005977

"i", -3, " = ", "10+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000006687

"i", -3, " = ", "10+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000009237

"i", -3, " = ", "10+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000013507

"i", -3, " = ", "10+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000018067

"i", -3, " = ", "10+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000019777
```

<u>Anmerkung:</u>

1. In der *Folge 3* sind insgesamt 8 Primzahlen CM_p situiert.

2. Alle Primzahlen CM_p haben die Endziffer EZ7

---

8. Die Bestimmung der in der *Folge 3* situierten Primzahlen CF_p:

- `for i from 10^350 to 10^350+2000 do: if (isprime (10+10*i+3) and isprime (10+10*i+3)) then print (("i", -3, " = ", "10+10*i", 10+10*i+3)) end_if: end_for:`

```
"i", -3, " = ", "10+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000004953

"i", -3, " = ", "10+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000007413

"i", -3, " = ", "10+10*i",
```

```
1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000010183
```

```
"i", -3, " = ", "10+10*i",
```

```
1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000015903
```

Anmerkung:

1. In der *Folge 3* sind insgesamt 4 Primzahlen CF_p situiert.

2. Alle Primzahlen CF_p haben die Endziffer EZ3.

---

9. Die Bestimmung der in der *Folge 3* situierten Zwillinge mod 6:

- ```
  for i from 10^350+8000 to 10^350+10000 do: if (isprime (10+10*i-3) and isprime
  (10+10*i+3)) then print (("i", -3, 10+10*i-3, " = ", "10+10*i", 10+10*i+3)) end_if:
  end_for:
  ```

```
"i", -3,
```

```
1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000083607
```

```
, " = ", "10+10*i",
```

```
1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000083613
```

Anmerkung:

1. In der *Folge 3* ist im Intervall i:= 10^350+8000 bis 10^350+10000 ein Zwilling mod 6 situiert.

2. In diesem Zwilling treten die Primzahlen (CM_p, CF_p) in den Endziffern EZ7, EZ3 auf.

3. Die Primzahl CM_p ist ein Element der Obergruppe OG II und die Primzahl CF_p ein Element der Obergruppe OG I.

Schlussfolgerungen:

1. Das Asymptote-Gesetz wird bestätigt, da in dem wesentlich größeren Intervall i:= 10^350 bis 10^350 +2000 wesentlich weniger Primzahlen CM_p und CF_p auftreten, als im Intervall i:= 0 bis 30.

2. Das Asymptote-Gesetz wird, wie ersichtlich auch für die Zwillinge mod 6 erfüllt.

```

Anlage 7

Die Bestimmung der Primzahlen SM_p und SF_p und der Zwillinge mod 8 in den Intervallen i:= 0 bis 30 und i:= 10^350 bis 10^350+2000 mit dem MuPAD - Test-Programm.

1. Die Bestimmung der Primzahlen SM_P, die in der *Folge 1* situiert sind:

- ```
  for i from 0 to 30 do: if (isprime (7+10*i-4) and isprime (7+10*i-4))
   then print (("i", -4, " = ", "17+10*i", 7+10*i-4)) end_if: end_for:
  ```

```
"i", -4, " = ", "17+10*i", 3
"i", -4, " = ", "17+10*i", 13
"i", -4, " = ", "17+10*i", 23
"i", -4, " = ", "17+10*i", 43
"i", -4, " = ", "17+10*i", 53
"i", -4, " = ", "17+10*i", 73
"i", -4, " = ", "17+10*i", 83
"i", -4, " = ", "17+10*i", 103
"i", -4, " = ", "17+10*i", 113
"i", -4, " = ", "17+10*i", 163
"i", -4, " = ", "17+10*i", 173
"i", -4, " = ", "17+10*i", 193
"i", -4, " = ", "17+10*i", 223
"i", -4, " = ", "17+10*i", 233
"i", -4, " = ", "17+10*i", 263
"i", -4, " = ", "17+10*i", 283
"i", -4, " = ", "17+10*i", 293
```

<u>Anmerkung:</u>

1. In der *Folge 1* sind 17 Primzahlen SM_p situiert.
2. Alle Primzahlen SM_p haben die Endziffer EZ3

---

2. Die Bestimmung der Primzahlen SF_p, die in der *Folge 1* situiert sind:

- ```
  for i from 0 to 30 do: if (isprime (7+10*i+4) and isprime (7+10*i+4))
  then print (("i", +4, " = ", "17+10*i", 7+10*i+4)) end_if: end_for:
  ```

```
"i", 4, " = ", "17+10*i", 11
"i", 4, " = ", "17+10*i", 31
"i", 4, " = ", "17+10*i", 41
"i", 4, " = ", "17+10*i", 61
"i", 4, " = ", "17+10*i", 71
"i", 4, " = ", "17+10*i", 101
"i", 4, " = ", "17+10*i", 131
```

```
"i", 4, " = ", "17+10*i", 151
"i", 4, " = ", "17+10*i", 181
"i", 4, " = ", "17+10*i", 191
"i", 4, " = ", "17+10*i", 211
"i", 4, " = ", "17+10*i", 241
"i", 4, " = ", "17+10*i", 251
"i", 4, " = ", "17+10*i", 271
"i", 4, " = ", "17+10*i", 281
"i", 4, " = ", "17+10*i", 311
```

Anmerkung:

1. In der *Folge 1* sind 16 Primzahlen SF_p situiert.

2. Alle Primzahlen SF_p haben die Endziffer EZ1.

---

3. Die Bestimmung der Zwillinge mod 8, die in der *Folge 1* situiert sind:

- ```
  for i from 0 to 30 do: if (isprime (7+10*i-4) and isprime (7+10*i+4))
  then print (("i", -4,7+10*i-4, " = ", "7+10*i", 7+10*i+4)) end_if: end_for:
  ```

```
"i", -4, 3, " = ", "7+10*i", 11
"i", -4, 23, " = ", "7+10*i", 31
"i", -4, 53, " = ", "7+10*i", 61
"i", -4, 173, " = ", "7+10*i", 181
"i", -4, 233, " = ", "7+10*i", 241
"i", -4, 263, " = ", "7+10*i", 271
```

Anmerkung:

1. In der *Folge 1* sind insgesamt 6 Zwillinge mod 8 situiert.

2. Die Primzahlen (SM_p, SF_p), die in den Zwillingen mod 8 auftreten, sind stets Elemente der gleichen Obergruppen: OG I und OG I oder aber OG II und OG II.

---

4. Die Bestimmung der Primzahlen SM_p, die in der *Folge 2* situiert sind:

- ```
  for i from 0 to 30 do: if (isprime (13+10*i-4) and isprime (13+10*i-4))
  then print (("i", -4, " = ", "13+10*i", 13+10*i-4)) end_if: end_for:
  ```

```
"i", -4, " = ", "13+10*i", 19
"i", -4, " = ", "13+10*i", 29
"i", -4, " = ", "13+10*i", 59
"i", -4, " = ", "13+10*i", 79
"i", -4, " = ", "13+10*i", 89
"i", -4, " = ", "13+10*i", 109
"i", -4, " = ", "13+10*i", 139
"i", -4, " = ", "13+10*i", 149
"i", -4, " = ", "13+10*i", 179
```

```
"i", -4, " = ", "13+10*i", 199
"i", -4, " = ", "13+10*i", 229
"i", -4, " = ", "13+10*i", 239
"i", -4, " = ", "13+10*i", 269
```

Anmerkung:

1. In der *Folge 2* sind insgesamt 13 Primzahlen SM_p situiert.
2. Alle Primzahlen SM_p haben die Endziffer EZ9

---

6. Die Bestimmung der Zwillinge mod 8, die in der *Folge 2* situiert sind:

- ```
  for i from 0 to 30 do: if (isprime (13+10*i-4) and isprime (13+10*i+4))
  then print (("i", -4, 13+10*i-4, " = ", "13+10*i", 13+10*i+4)) end_if: end_for:
  ```

```
"i", -4, 29, " = ", "13+10*i", 37
"i", -4, 59, " = ", "13+10*i", 67
"i", -4, 89, " = ", "13+10*i", 97
"i", -4, 149, " = ", "13+10*i", 157
"i", -4, 269, " = ", "13+10*i", 277
```

Anmerkung:

1. In der *Folge 2* sind insgesamt 5 Zwillinge mod 8 situiert.
2. Die Primzahlen (SM_p, SF_p), die in den Zwillingen mod 8 auftreten, sind stets Elemente der gleichen Obergruppen OG I und OG I oder aber OG II und OG II.

---

7. Die Bestimmung der Primzahlen SM_p, die in der *Folge 3* situiert sind:

- ```
  for i from 0 to 30 do: if (isprime (15+10*i-4) and isprime (15+10*i-4))
  then print (("i", -4, " = ", "15+10*i", 15+10*i-4)) end_if: end_for:
  ```

```
"i", -4, " = ", "15+10*i", 11
"i", -4, " = ", "15+10*i", 31
"i", -4, " = ", "15+10*i", 41
"i", -4, " = ", "15+10*i", 61
"i", -4, " = ", "15+10*i", 71
"i", -4, " = ", "15+10*i", 101
"i", -4, " = ", "15+10*i", 131
"i", -4, " = ", "15+10*i", 151
"i", -4, " = ", "15+10*i", 181
"i", -4, " = ", "15+10*i", 191
"i", -4, " = ", "15+10*i", 211
"i", -4, " = ", "15+10*i", 241
"i", -4, " = ", "15+10*i", 251
"i", -4, " = ", "15+10*i", 271
"i", -4, " = ", "15+10*i", 281
"i", -4, " = ", "15+10*i", 311
```

Anmerkung:

1. In der *Folge 3.* sind insgesamt 16 Primzahlen SM_p situiert.
2. Alle Primzahlen SM_p haben die Endziffer EZ1.

---

8. Die Bestimmung der Primzahlen SF_p, die in der *Folge 3* situiert sind:

-
```
for i from 0 to 30 do: if (isprime (15+10*i+4) and isprime (15+10*i+4))
 then print (("i", +4, " = ", "15+10*i", 15+10*i+4)) end_if: end_for:
```

```
"i", 4, " = ", "15+10*i", 19
"i", 4, " = ", "15+10*i", 29
"i", 4, " = ", "15+10*i", 59
"i", 4, " = ", "15+10*i", 79
"i", 4, " = ", "15+10*i", 89
"i", 4, " = ", "15+10*i", 109
"i", 4, " = ", "15+10*i", 139
"i", 4, " = ", "15+10*i", 149
"i", 4, " = ", "15+10*i", 179
"i", 4, " = ", "15+10*i", 199
"i", 4, " = ", "15+10*i", 229
"i", 4, " = ", "15+10*i", 239
"i", 4, " = ", "15+10*i", 269
```

Anmerkung:

1. In der *Folge 3* sind insgesamt 13 Primzahlen SF_p situiert.

2. Alle Primzahlen SF_p haben die Endziffer EZ9.

---

9. Die Bestimmung der Zwillinge mod 8, die in der *Folge 3* situiert sind:

-
```
for i from 0 to 30 do: if (isprime (15+10*i-4) and isprime (15+10*i+4))
then print (("i", -4, 15+10*i-4, " = ", "15+10*i", 15+10*i+4)).end_if: end_for:
```

```
"i", -4, 11, " = ", "15+10*i", 19
"i", -4, 71, " = ", "15+10*i", 79
"i", -4, 101, " = ", "15+10*i", 109
"i", -4, 131, " = ", "15+10*i", 139
"i", -4, 191, " = ", "15+10*i", 199
```

Anmerkung:

1. In der *Folge 3* sind insgesamt 5 Zwillinge mod 8 situiert.

2. Die Primzahlen (SM_p, SF_p) die in diese Zwillinge mod 8 eingehen, sind stets beide Elemente der Obergruppe OG I und OG I oder aber der Obergruppe OG II und OG II.

Die Bestimmung der Primzahlen SM_p und SF_p sowie der Zwillinge mod 8, die im Intervall i:= 10^350 bis 10^350+2000 in *Folge 1*, *Folge 2* und *Folge 3* situiert sind:

1. Die Bestimmung der Primzahlen SM_p, die in der *Folge 1* situiert sind:

- ```
  for i from 10^350 to 10^350+2000 do: if (isprime (7+10*i-4) and isprime (7+10*i-4))
      then print (("i", -4, " = ", "7+10*i", 7+10*i-4)) end_if: end_for:
  ```

```
"i", -4, " = ", "7+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000004953

"i", -4, " = ", "7+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000007413

"i", -4, " = ", "7+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000010183

"i", -4, " = ", "7+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000015903
```

Anmerkung:

1. In der *Folge 1* sind insgesamt 4 Primzahlen SM_p situiert.
2. Alle Primzahlen SM_p haben die Endziffer EZ3.

---

2. Die Bestimmung der Primzahlen SF_p, die in der *Folge 1* situiert sind:

- ```
  for i from 10^350 to 10^350+2000 do: if (isprime (7+10*i+4) and isprime (7+10*i+4))
      then print (("i", +4, " = ", "7+10*i", 7+10*i+4)) end_if: end_for:
  ```

```
"i", 4, " = ", "7+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000501

"i", 4, " = ", "7+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000008601

"i", 4, " = ", "7+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000009891

"i", 4, " = ", "7+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000011721

"i", 4, " = ", "7+10*i",
```

```
100000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000011931

"i", 4, " = ", "7+10*i",

100000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000012001

"i", 4, " = ", "7+10*i",

100000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000012021

"i", 4, " = ", "7+10*i",

100000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000012301

"i", 4, " = ", "7+10*i",

100000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000012621
```

Anmerkung:

1. In der *Folge 1* sind insgesamt 9 Primzahlen SF_p situiert.
2. Alle Primzahlen SF_p haben die Endziffer EZ1.

---

3. Die Bestimmung der Zwillinge mod 8, die in der *Folge 1* situiert sind:

Anmerkung:
In der *Folge 1* ist kein Zwilling mod 4 situiert - auch nicht im erweiterten Intervall bis 10^350 +10000.

---

4. Die Bestimmung der Primzahlen SM_p, die in der *Folge 2* situiert sind:

- ```
  for i from 10^350 to 10^350+2000 do: if (isprime (13+10*i-4) and isprime (13+10*i-4))
      then print (("i", -4, " = ", "13+10*i", 13+10*i-4)) end_if: end_for:
  ```

```
"i", -4, " = ", "13+10*i",

100000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000249

"i", -4, " = ", "13+10*i",

100000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000004879

"i", -4, " = ", "13+10*i",

100000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
```
```

```
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000008059

"i", -4, " = ", "13+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000010119

"i", -4, " = ", "13+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000010939

"i", -4, " = ", "13+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012549

"i", -4, " = ", "13+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000014809
```

Anmerkung:

1. In der *Folge 2* sind insgesamt 7 Primzahlen SM_p situiert, die alle die Endziffer EZ9 haben.

---

5. Die Bestimmung der Primzahlen SF_p, die in der *Folge 2* situiert sind:

```
• for i from 10^350 to 10^350+2000 do: if (isprime (13+10*i+4) and isprime (13+10*i+4))
    then print (("i",+4, " = ", "13+10*i", 13+10*i+4)) end_if: end_for:

"i", 4, " = ", "13+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000207

"i", 4, " = ", "13+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000001447

"i", 4, " = ", "13+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000005977

"i", 4, " = ", "13+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000006687

"i", 4, " = ", "13+10*i",
```

```
1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000009237

"i", 4, " = ", "13+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000013507

"i", 4, " = ", "13+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000018067

"i", 4, " = ", "13+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000019777
```

Anmerkung:

1. In der *Folge 2* sind insgesamt 8 Primzahlen SF_p situiert, die alle die Endziffer EZ7 haben.

---

6. Die Bestimmung der Zwillinge mod 2, die in der *Folge 2* situiert sind:

Anmerkung: In der *Folge 2* ist kein Zwilling mod 8 im Intervall: i:= 10^350 bis 10^350+2000 situiert.

---

7. Die Bestimmung der Primzahlen SM_p, die in der *Folge 3* situiert sind:

- ```
  for i from 10^350 to 10^350+2000 do: if (isprime (15+10*i-4) and isprime (15+10*i-4))
      then print (("i",-4, " = ", "15+10*i", 15+10*i-4)) end_if: end_for:
  ```

```
"i", -4, " = ", "15+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000501

"i", -4, " = ", "15+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000008601

"i", -4, " = ", "15+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000009891

"i", -4, " = ", "15+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000011721

"i", -4, " = ", "15+10*i",
```
```

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000011931

"i", -4, " = ", "15+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000012001

"i", -4, " = ", "15+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000012021

"i", -4, " = ", "15+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000012301

"i", -4, " = ", "15+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000012621

<u>Anmerkung:</u>

1. In der *Folge 3* sind insgesamt 9 Primzahlen SM_p situiert, die alle die Endziffer EZ1 haben.

---

8. Die Bestimmung der Primzahlen SF_p, die in der *Folge 3* situiert sind:

-
```
for i from 10^350 to 10^350+2000 do: if (isprime (15+10*i+4) and isprime (15+10*i+4))
    then print (("i", +4, " = ", "15+10*i", 15+10*i+4)) end_if: end_for:
```

"i", 4, " = ", "15+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000249

"i", 4, " = ", "15+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000004879

"i", 4, " = ", "15+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000008059

"i", 4, " = ", "15+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000010119

```
"i", 4, " = ", "15+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000010939

"i", 4, " = ", "15+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000012549

"i", 4, " = ", "15+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000014809
```

<u>Anmerkung:</u>

1. In der *Folge 3* sind insgesamt 7 Primzahlen SF_p situiert, die alle die Endziffer EZ9 haben.

---

9. Die Bestimmung der Zwillinge mod 8, die in der *Folge 3* situiert sind:

<u>Anmerkungen und Schlussfolgerungen:</u>

1. Im betrachteten Intervall sind in der *Folge 3* keine Zwillinge mod 8 situiert. Des weiteren sind in dieser Folge auch im erweiterten Intervall i:= 10^350 bis 10^350+100.000 keine Zwillinge mod 8 situiert.

2. Dieser Sachverhalt ist eine Folge des Asymptote-Gesetzes.

3. Die Bildung von Zwillingen mod 8 aus der Menge der Primzahlen: SM_p und SF_p erfordert die Erfüllung der (wechselseitigen) Und-Bedingung:

   $SF_p - 4 <=> SM_p + 4$ aus der sich die Erfüllungs-Bedingung: $SM_p + 8 = SF_p$ ergibt.

Anlage 8

Die Bestimmung der Primzahlen CM_p und CF_p sowie der Zwillinge mod 10, die in unterschiedlichen Intervallen i:= 0 bis 30 und i:= 10^150 bis 10^150 +2000 situiert sind mit dem MuPAD - Test-Programm

Einführung:

Aus den Bildungsgesetzen der Zwillinge mod 2n, n ≥1 ergibt sich, dass Zwillinge mod 10 in vier Folgen: *Folge 1*, *Folge 2*, *Folge 3* und *Folge 4* auftreten können. Aus diesem Grund sind die Primzahlen CM_p und CF_p, die Zwillinge mod 10 bilden können in diesen vier Folgen zu ermitteln.

1. Die Bestimmung der Primzahlen CM_p, die in der *Folge 1* situiert sind:

- ```
  for i from 0 to 30 do: if (isprime (6+10*i-5) and isprime (6+10*i-5))
     then print (("i",-5, " = ", "6+10*i", 6+10*i-5)) end_if: end_for:
  ```

```
"i", -5, " = ", "6+10*i", 11
"i", -5, " = ", "6+10*i", 31
"i", -5, " = ", "6+10*i", 41
"i", -5, " = ", "6+10*i", 61
"i", -5, " = ", "6+10*i", 71
"i", -5, " = ", "6+10*i", 101
"i", -5, " = ", "6+10*i", 131
"i", -5, " = ", "6+10*i", 151
"i", -5, " = ", "6+10*i", 181
"i", -5, " = ", "6+10*i", 191
"i", -5, " = ", "6+10*i", 211
"i", -5, " = ", "6+10*i", 241
"i", -5, " = ", "6+10*i", 251
"i", -5, " = ", "6+10*i", 271
"i", -5, " = ", "6+10*i", 281.
```

Anmerkung:

1. In der *Folge 1* sind insgesamt 14 Primzahlen CM_p situiert.
2. Alle (diese) Primzahlen CM_p haben die Endziffer EZ1.

---

2. Die Bestimmung der Primzahlen CF_p, die in der *Folge 1* situiert sind:

- ```
  for i from 0 to 30 do: if (isprime (6+10*i+5) and isprime (6+10*i+5))
     then print (("i",+5, " = ", "6+10*i", 6+10*i+5)) end_if: end_for:
  ```

```
"i", 5, " = ", "6+10*i", 11
"i", 5, " = ", "6+10*i", 31
"i", 5, " = ", "6+10*i", 41
"i", 5, " = ", "6+10*i", 61
```

```
"i", 5, " = ", "6+10*i", 71
"i", 5, " = ", "6+10*i", 101
"i", 5, " = ", "6+10*i", 131
"i", 5, " = ", "6+10*i", 151
"i", 5, " = ", "6+10*i", 181
"i", 5, " = ", "6+10*i", 191
"i", 5, " = ", "6+10*i", 211
"i", 5, " = ", "6+10*i", 241
"i", 5, " = ", "6+10*i", 251
"i", 5, " = ", "6+10*i", 271
"i", 5, " = ", "6+10*i", 281
"i", 5, " = ", "6+10*i", 311
```

Anmerkung:

1. In der *Folge 1* sind insgesamt 16 Primzahlen CF_p situiert.

2. Alle (diese) Primzahlen CF_p haben die Endziffer EZ1.

---

3. Die Bestimmung der Zwillinge mod 10, die in der *Folge 1* situiert sind.

-
```
for i from 0 to 30 do: if (isprime (6+10*i-5) and isprime (6+10*i+5))
then print (("i",-5, 6+10*i-5, " = ", "6+10*i", 6+10*i+5)) end_if: end_for:
```

```
"i", -5, 31, " = ", "6+10*i", 41
"i", -5, 61, " = ", "6+10*i", 71
"i", -5, 181, " = ", "6+10*i", 191
"i", -5, 241, " = ", "6+10*i", 251
"i", -5, 271, " = ", "6+10*i", 281
```

Anmerkung:

1. In der *Folge 1* sind insgesamt 5 Zwillinge mod 10 situiert.

2. Die Primzahlen (CM_p, CF_p), die diese Zwillinge mod 10 bilden, haben die Endziffer EZ1, jedoch sind diese Primzahlen (jeweils) Elemente der Obergruppen OG I und OG II bzw. vice versa Elemente der Obegruppe OG II und OG I.

---

4. Die Bestimmung der Primzahlen CM_p, die in der *Folge 2* situiert sind:

-
```
for i from 0 to 30 do: if (isprime (8+10*i-5) and isprime (8+10*i-5))
then print (("i",-5, " = ", "8+10*i", 8+10*i-5)) end_if: end_for:
```

```
"i", -5, " = ", "8+10*i", 3
"i", -5, " = ", "8+10*i", 13
"i", -5, " = ", "8+10*i", 23
"i", -5, " = ", "8+10*i", 43
"i", -5, " = ", "8+10*i", 53
"i", -5, " = ", "8+10*i", 73
"i", -5, " = ", "8+10*i", 83
```

```
"i", -5, " = ", "8+10*i", 103
"i", -5, " = ", "8+10*i", 113
"i", -5, " = ", "8+10*i", 163
"i", -5, " = ", "8+10*i", 173
"i", -5, " = ", "8+10*i", 193
"i", -5, " = ", "8+10*i", 223
"i", -5, " = ", "8+10*i", 233
"i", -5, " = ", "8+10*i", 263
"i", -5, " = ", "8+10*i", 283
"i", -5, " = ", "8+10*i", 293
```

Anmerkung:

1. In der *Folge 2* sind insgesamt 17 Primzahlen CM_p situiert.

2. Alle (diese) Primzahlen CM_p haben die Endziffer EZ3

---

5. Die Bestimmung der Primzahlen CF_p, die in der *Folge 2* situiert sind:

```
for i from 0 to 30 do: if (isprime (8+10*i+5) and isprime (8+10*i+5))
  then print (("i",+5, " = ", "8+10*i", 8+10*i+5)) end_if: end_for:
```

```
"i", 5, " = ", "8+10*i", 13
"i", 5, " = ", "8+10*i", 23
"i", 5, " = ", "8+10*i", 43
"i", 5, " = ", "8+10*i", 53
"i", 5, " = ", "8+10*i", 73
"i", 5, " = ", "8+10*i", 83
"i", 5, " = ", "8+10*i", 103
"i", 5, " = ", "8+10*i", 113
"i", 5, " = ", "8+10*i", 163
"i", 5, " = ", "8+10*i", 173
"i", 5, " = ", "8+10*i", 193
"i", 5, " = ", "8+10*i", 223
"i", 5, " = ", "8+10*i", 233
"i", 5, " = ", "8+10*i", 263
"i", 5, " = ", "8+10*i", 283
"i", 5, " = ", "8+10*i", 293
"i", 5, " = ", "8+10*i", 313
```

Anmerkung:

1. In der *Folge 2* sind insgesamt 17 Primzahlen CF_p situiert.

2. Alle (diese) Primzahlen CF_p haben die Endziffer EZ3.

---

5. Die Bestimmung der Zwillinge mod 10, die in der *Folge 2* situiert sind:

```
for i from 0 to 30 do: if (isprime (8+10*i-5) and isprime (8+10*i+5))
```

```
then print (("i",-5,8+10*i-5, " = ", "8+10*i", 8+10*i+5)) end_if: end_for:
```

```
"i", -5, 3, " = ", "8+10*i", 13
"i", -5, 13, " = ", "8+10*i", 23
"i", -5, 43, " = ", "8+10*i", 53
"i", -5, 73, " = ", "8+10*i", 83
"i", -5, 103, " = ", "8+10*i", 113
"i", -5, 163, " = ", "8+10*i", 173
"i", -5, 223, " = ", "8+10*i", 233
"i", -5, 283, " = ", "8+10*i", 293
```

Anmerkung:

1. In der *Folge 2* sind insgesamt 8 Zwillinge mod 10 situiert.

2. Die Primzahlen (CM_p, CF_p), die diese Zwillinge mod 10 bilden, haben alle die Endziffer EZ3 und sind (jeweils) Elemente der (divergenten) Obergruppen: OG I und OG II.

---

7. Die Bestimmung der Primzahlen CM_p, die in der *Folge 3* situiert sind:

- ```
  for i from 0 to 30 do: if (isprime (12+10*i-5) and isprime (12+10*i-5))
    then print (("i",-5, " = ", "12+10*i", 12+10*i-5)) end_if: end_for:
  ```

```
"i", -5, " = ", "12+10*i", 7
"i", -5, " = ", "12+10*i", 17
"i", -5, " = ", "12+10*i", 37
"i", -5, " = ", "12+10*i", 47
"i", -5, " = ", "12+10*i", 67
"i", -5, " = ", "12+10*i", 97
"i", -5, " = ", "12+10*i", 107
"i", -5, " = ", "12+10*i", 127
"i", -5, " = ", "12+10*i", 137
"i", -5, " = ", "12+10*i", 157
"i", -5, " = ", "12+10*i", 167
"i", -5, " = ", "12+10*i", 197
"i", -5, " = ", "12+10*i", 227
"i", -5, " = ", "12+10*i", 257
"i", -5, " = ", "12+10*i", 277
"i", -5, " = ", "12+10*i", 307
```

Anmerkung:

1. In der *Folge 3* sind insgesamt 16 Primzahlen CM_p situiert, die alle die Endziffer EZ7 haben.

---

8. Die Bestimmung der Primzahlen CF_p, die in der *Folge 3* situiert sind:

- ```
  for i from 0 to 30 do: if (isprime (12+10*i+5) and isprime (12+10*i+5))
    then print (("i",+5, " = ", "12+10*i", 12+10*i+5)) end_if: end_for:
  ```

```
"i", 5, " = ", "12+10*i", 17
```

```
"i", 5, " = ", "12+10*i", 37
"i", 5, " = ", "12+10*i", 47
"i", 5, " = ", "12+10*i", 67
"i", 5, " = ", "12+10*i", 97
"i", 5, " = ", "12+10*i", 107
"i", 5, " = ", "12+10*i", 127
"i", 5, " = ", "12+10*i", 137
"i", 5, " = ", "12+10*i", 157
"i", 5, " = ", "12+10*i", 167
"i", 5, " = ", "12+10*i", 197
"i", 5, " = ", "12+10*i", 227
"i", 5, " = ", "12+10*i", 257
"i", 5, " = ", "12+10*i", 277
"i", 5, " = ", "12+10*i", 307
"i", 5, " = ", "12+10*i", 317
```

Anmerkung:

1. In der *Folge 3* sind insgesamt 16 Primzahlen CF_p situiert, die alle die Endziffer EZ7 haben.

---

9. Die Bestimmung der Zwillinge mod 10, die in der *Folge 3* situiert sind:

- ```
  for i from 0 to 30 do: if (isprime (12+10*i-5) and isprime (12+10*i+5))
  then print (("i",-5,12+10*i-5, " = ", "12+10*i", 12+10*i+5)) end_if: end_for:
  ```

```
"i", -5, 7, " = ", "12+10*i", 17
"i", -5, 37, " = ", "12+10*i", 47
"i", -5, 97, " = ", "12+10*i", 107
"i", -5, 127, " = ", "12+10*i", 137
"i", -5, 157, " = ", "12+10*i", 167
"i", -5, 307, " = ", "12+10*i", 317
```

Anmerkung:

1. In der *Folge 3* sind insgesamt 6 Zwillinge mod 10 situiert.

2. Die Primzahlen (CM_p, CF_p), die diese Zwillinge mod 10 bilden, haben alle die endziffer EZ7 und sind (jeweils) Elemente der (divergenten) Obergruppen: OG I und OG II.

---

10. Die Bestimmung der Primzahlen CM_p, die in *der Folge* 4 situiert sind:

- ```
  for i from 0 to 30 do: if (isprime (14+10*i-5) and isprime (14+10*i-5))
    then print (("i",-5, " = ", "14+10*i", 14+10*i-5)) end_if: end_for:
  ```

```
"i", -5, " = ", "14+10*i", 19
"i", -5, " = ", "14+10*i", 29
"i", -5, " = ", "14+10*i", 59
"i", -5, " = ", "14+10*i", 79
"i", -5, " = ", "14+10*i", 89
"i", -5, " = ", "14+10*i", 109
"i", -5, " = ", "14+10*i", 139
```

```
"i", -5, " = ", "14+10*i", 149
"i", -5, " = ", "14+10*i", 179
"i", -5, " = ", "14+10*i", 199
"i", -5, " = ", "14+10*i", 229
"i", -5, " = ", "14+10*i", 239
"i", -5, " = ", "14+10*i", 269
```

Anmerkung:

1. In der *Folge 4* sind insgesamt 13 Primzahlen CM_p situiert, die ale die Endziffer EZ9 haben.

---

11. Die Bestimmung der Primzahlen CF_p, die in der *Folge 4* situiert sind:

-
```
for i from 0 to 30 do: if (isprime (14+10*i+5) and isprime (14+10*i+5))
  then print (("i",+5, " = ", "14+10*i", 14+10*i+5)) end_if: end_for:
```

```
"i", 5, " = ", "14+10*i", 19
"i", 5, " = ", "14+10*i", 29
"i", 5, " = ", "14+10*i", 59
"i", 5, " = ", "14+10*i", 79
"i", 5, " = ", "14+10*i", 89
"i", 5, " = ", "14+10*i", 109
"i", 5, " = ", "14+10*i", 139
"i", 5, " = ", "14+10*i", 149
"i", 5, " = ", "14+10*i", 179
"i", 5, " = ", "14+10*i", 199
"i", 5, " = ", "14+10*i", 229
"i", 5, " = ", "14+10*i", 239
"i", 5, " = ", "14+10*i", 269
```

Anmerkung:

1. In der *Folge 4* sind insgesamt 13 Primzahlen CF_p situiert, die alle die Endziffer EZ9 haben.

---

12. Die Bestimmung der Zwillinge mod 10, die in der *Folge 4* situiert sind:

-
```
for i from 0 to 30 do: if (isprime (14+10*i-5) and isprime (14+10*i+5))
then print (("i",-5,14+10*i-5, " = ", "14+10*i", 14+10*i+5)) end_if: end_for:
```

```
"i", -5, 19, " = ", "14+10*i", 29
"i", -5, 79, " = ", "14+10*i", 89
"i", -5, 139, " = ", "14+10*i", 149
"i", -5, 229, " = ", "14+10*i", 239
```

Anmerkung:

1. In der *Folge 4* sind insgesamt 4 Zwillinge mod 10 situiert.

2. Die Primzahlen (CM_p, CF_p), die diese Zwillinge mod 10 bilden, haben allesamt die Endziffer EZ9 und sind Elemente der (divergenten) Obergruppen: OG I und OG II.

Die Bestimmung der Primzahlen CM_p und CF_p, sowie der Zwillinge mod 10, die in den vier Folgen: *Folge 1*, *Folge 2*, *Folge 3* und *Folge 4* im Intervall i:= 10^350 bis 10^350 +2000 situiert sind.

1. Die Bestimmung der Primzahlen CM_p, die in der *Folge 1* situiert sind:

- ```
  for i from 10^350 to 10^350+2000 do: if (isprime (6+10*i-5) and isprime (6+10*i-5))
      then print (("i",-5, " = ", "6+10*i", 6+10*i-5)) end_if: end_for:
  ```

```
"i", -5, " = ", "6+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000501

"i", -5, " = ", "6+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000008601

"i", -5, " = ", "6+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000009891

"i", -5, " = ", "6+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000011721

"i", -5, " = ", "6+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000011931

"i", -5, " = ", "6+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012001

"i", -5, " = ", "6+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012021

"i", -5, " = ", "6+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012301

"i", -5, " = ", "6+10*i",
```
```

```
1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012621
```

<u>Anmerkung:</u>

1. In der *Folge 1* sind insgesamt 9 Primzahlen CM_p situiert, die alle die Endziffer EZ1 haben

---

2. Die Bestimmung der in der *Folge 2* situierten Primzahlen CF_p:

- ```
  for i from 10^350 to 10^350+2000 do: if (isprime (6+10*i+5) and isprime (6+10*i+5))
      then print (("i",+5, " = ", "6+10*i", 6+10*i+5)) end_if: end_for:
  ```

```
"i", 5, " = ", "6+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000501

"i", 5, " = ", "6+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000008601

"i", 5, " = ", "6+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000009891

"i", 5, " = ", "6+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000011721

"i", 5, " = ", "6+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000011931

"i", 5, " = ", "6+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012001

"i", 5, " = ", "6+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012021

"i", 5, " = ", "6+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012301
```
```

```
"i", 5, " = ", "6+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012621
```

Anmerkung:

1. In der *Folge 1* sind insgesamt 10 Primzahlen CF_p, situiert die alle die Endziffer EZ1 haben

---

-

3. Die Bestimmung der Zwillinge mod 10, die in der *Folge 1* situiert sind:

Anmerkung:

1. In der *Folge 1* ist kein Zwilling mod 10 situiert.

-

4. Die Bestimmung der Primzahlen CM_p, die in der *Folge 2* situiert sind:

- ```
  for i from 10^350 to 10^350+2000 do: if (isprime (8+10*i-5) and isprime (8+10*i-5))
      then print (("i",-5," = ", "8+10*i", 8+10*i-5)) end_if: end_for:
  ```

```
"i", -5, " = ", "8+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000004953

"i", -5, " = ", "8+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000007413

"i", -5, " = ", "8+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000010183

"i", -5, " = ", "8+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000015903
```

Anmerkung:

1. In der *Folge 2* sind insgesamt 4 Primzahlen CM_p situiert, die alle die Endziffer EZ3 haben

---

5. Die Bestimmung der Primzahlen CF_p, die in der *Folge 2* situiert sind:

- ```
  for i from 10^350 to 10^350+2000 do: if (isprime (8+10*i+5) and isprime (8+10*i+5))
      then print (("i", +5, " = ", "8+10*i", 8+10*i+5)) end_if: end_for:
  ```

```
"i", 5, " = ", "8+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
```

```
000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000004953

"i", 5, " = ", "8+10*i",

100000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000007413

"i", 5, " = ", "8+10*i",

100000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000010183

"i", 5, " = ", "8+10*i",

100000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000015903
```

Anmerkung:

1. In der *Folge 2* sind insgesamt 4 Primzahlen CF_p situiert, die alle die Endziffer EZ3 haben.

---

6. Die Bestimmung der in der *Folge 2* situierten Zwillinge mod 10:

Anmerkung: Im analysierten Intervall sind in der *Folge 2* keine Zwillinge mod 10 situiert.

---

7. Die Bestimmung der Primzahlen CM_p, die in der *Folge 3* situiert sind:

- `for i from 10^350 to 10^350+2000 do: if (isprime (12+10*i-5) and isprime (12+10*i-5)) then print (("i",-5," = ", "12+10*i", 12+10*i-5)) end_if: end_for:`

```
"i", -5, " = ", "12+10*i",

100000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000207

"i", -5, " = ", "12+10*i",

100000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000001447

"i", -5, " = ", "12+10*i",

100000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000005977

"i", -5, " = ", "12+10*i",

100000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000006687

"i", -5, " = ", "12+10*i",

100000000000000000000000000000000000000000000000000000000000000000000000000000\
```

```
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000009237

"i", -5, " = ", "12+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000013507

"i", -5, " = ", "12+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000018067

"i", -5, " = ", "12+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000019777
```

Anmerkung:

1. In der *Folge 3* sind insgesamt 8 Primzahlen CM_p situiert, die alle die Endziffer EZ7 haben.

---

8. Die Bestimmung der in der *Folge 3* situierten Primzahlen CF_p:

- ```
  for i from 10^350 to 10^350+2000 do: if (isprime (12+10*i+5) and isprime (12+10*i+5))
      then print (("i", +5, " = ", "12+10*i", 12+10*i+5)) end_if: end_for:
  ```

```
"i", 5, " = ", "12+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000207

"i", 5, " = ", "12+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000001447

"i", 5, " = ", "12+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000005977

"i", 5, " = ", "12+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000006687

"i", 5, " = ", "12+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000009237

"i", 5, " = ", "12+10*i",
```
```

```
1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000013507

"i", 5, " = ", "12+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000018067

"i", 5, " = ", "12+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000019777
```

Anmerkung:

1. In der *Folge 3* sind insgesamt 8 Primzahlen CF_p situiert, die alle die Endziffer EZ7 haben.

---

9. Die Bestimmung der in der *Folge 3* situierten Zwillinge mod 10:

- ```
  for i from 10^350 to 10^350+2000 do: if (isprime (14+10*i-5) and isprime (14+10*i-5))
      then print (("i",-5," = ", "14+10*i", 14+10*i-5)) end_if: end_for:
  ```

```
"i", -5, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000249

"i", -5, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000004879

"i", -5, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000008059

"i", -5, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000010119

"i", -5, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000010939

"i", -5, " = ", "14+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012549
```
```

```
"i", -5, " = ", "14+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000014809
```

Anmerkung:

1. In der *Folge 4* sind insgesamt 8 Primzahlen CM_p situiert, die alle die Endziffer EZ9 haben.

---

11. Die Bestimmung der in der *Folge 4* situierten Primzahlen CF_p:

- ```
  for i from 10^350 to 10^350+2000 do: if (isprime (14+10*i+5) and isprime (14+10*i+5))
      then print (("i", +5, " = ", "14+10*i", 14+10*i+5)) end_if: end_for:
  ```

```
"i", 5, " = ", "14+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000249

"i", 5, " = ", "14+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000004879

"i", 5, " = ", "14+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000008059

"i", 5, " = ", "14+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000010119

"i", 5, " = ", "14+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000010939

"i", 5, " = ", "14+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000012549

"i", 5, " = ", "14+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000014809
```

Anmerkung:

1. In der *Folge 4* sind insgesamt 8 Primzahlen CF_p situiert, die alle die Endziffer EZ9 haben.

---

12. Die Bestimmung der in der *Folge 4* situierten Zwillinge mod 10:

Anmerkung: Auch im zehn mal größeren Intervall ist kein Zwilling mod 10 in der *Folge 4* situiert.

```

Anlage 9

Die Generierung von Primzahlen vom Typ SM_p, SF_p und von Zwillingen mod 20, die in den vier Folgen: *Folge 1, Folge 2, Folge 3, Folge 4* auftreten, mit dem MuPAD - Testprogramm.

1. Die Bestimmung der Primzahlen SM_p, die im Intervall i:= 0 bis 30 in der *Folge 1* situiert sind:

- ```
  for i from 0 to 30 do: if (isprime (11+10*i-10) and isprime (11+10*i-10))
     then print (("i", i, " = ", "11+10*i", 11+10*i-10)) end_if: end_for:
  ```

```
"i", 1, " = ", "11+10*i", 11
"i", 3, " = ", "11+10*i", 31
"i", 4, " = ", "11+10*i", 41
"i", 6, " = ", "11+10*i", 61
"i", 7, " = ", "11+10*i", 71
"i", 10, " = ", "11+10*i", 101
"i", 13, " = ", "11+10*i", 131
"i", 15, " = ", "11+10*i", 151
"i", 18, " = ", "11+10*i", 181
"i", 19, " = ", "11+10*i", 191
"i", 21, " = ", "11+10*i", 211
"i", 24, " = ", "11+10*i", 241
"i", 25, " = ", "11+10*i", 251
"i", 27, " = ", "11+10*i", 271
"i", 28, " = ", "11+10*i", 281
```

Anmerkung:

1. In der *Folge 1* sind insgesamt 15 Primzahlen SM_p, EZ1 situiert.

---

2. Die Bestimmung der Primzahlen SF_p, die in der *Folge 1* situiert sind:

- ```
  for i from 0 to 30 do: if (isprime (11+10*i+10) and isprime (11+10*i+10))
     then print (("i", i, " = ", "11+10*i", 11+10*i+10)) end_if: end_for:
  ```

```
"i", 1, " = ", "11+10*i", 31
"i", 2, " = ", "11+10*i", 41
"i", 4, " = ", "11+10*i", 61
"i", 5, " = ", "11+10*i", 71
"i", 8, " = ", "11+10*i", 101
"i", 11, " = ", "11+10*i", 131
"i", 13, " = ", "11+10*i", 151
"i", 16, " = ", "11+10*i", 181
"i", 17, " = ", "11+10*i", 191
"i", 19, " = ", "11+10*i", 211
"i", 22, " = ", "11+10*i", 241
```

```
"i", 23, " = ", "11+10*i", 251
"i", 25, " = ", "11+10*i", 271
"i", 26, " = ", "11+10*i", 281
"i", 29, " = ", "11+10*i", 311
```

Anmerkung:

1. In der *Folge 1* sind insgesamt 15 Primzahlen SF_p, EZ1 situiert.

---

3. Die Bestimmung der Zwillinge mod 20, die in der *Folge 1* situiert sind:

- ```
  for i from 0 to 30 do: if (isprime (11+10*i-10) and isprime (11+10*i+10))
  then print (("i", i, 11+10*i-10, " = ", "11+10*i", 11+10*i+10)) end_if: end_for:
  ```

```
"i", 1, 11, " = ", "11+10*i", 31    ... OG II, OG II
"i", 4, 41, " = ", "11+10*i", 61    ... OG I, OG I
"i", 13, 131, " = ", "11+10*i", 151 ... OG II, OG II
"i", 19, 191, " = ", "11+10*i", 211 ... OG II, OG II
"i", 25, 251, " = ", "11+10*i", 271 ... OG II, OG II
```

Anmerkung:

1. In der *Folge 1* sind insgesamt 5 Zwillinge mod 20 situiert.
2. Die Primzahlen (SM_p, SF_p) haben allesamt die Endziffer EZ1
3. Die Primzahlen SM_p und SF_p, die die Zwillinge mod 20 bilden, sind Elemente der gleichen Obergruppe

---

4. Die Bestimmung der Primzahlen SM_p, die in der *Folge 2* situiert sind:

- ```
  for i from 0 to 30 do: if (isprime (13+10*i-10) and isprime (13+10*i-10))
  then print (("i", i, " = ", "13+10*i", 13+10*i-10)) end_if: end_for:
  ```

```
"i", 0, " = ", "13+10*i", 3
"i", 1, " = ", "13+10*i", 13
"i", 2, " = ", "13+10*i", 23
"i", 4, " = ", "13+10*i", 43
"i", 5, " = ", "13+10*i", 53
"i", 7, " = ", "13+10*i", 73
"i", 8, " = ", "13+10*i", 83
"i", 10, " = ", "13+10*i", 103
"i", 11, " = ", "13+10*i", 113
"i", 16, " = ", "13+10*i", 163
"i", 17, " = ", "13+10*i", 173
"i", 19, " = ", "13+10*i", 193
"i", 22, " = ", "13+10*i", 223
"i", 23, " = ", "13+10*i", 233
"i", 26, " = ", "13+10*i", 263
"i", 28, " = ", "13+10*i", 283
"i", 29, " = ", "13+10*i", 293
```

Anmerkung:

1. In der *Folge 2* sind insgesamt 17 Primzahlen SM_p, EZ3 situiert.

---

5. Die Bestimmung der Primzahlen SF_p, die in der *Folge 2* situiert sind:

-
```
for i from 0 to 30 do: if (isprime (13+10*i+10) and isprime (13+10*i+10))
  then print (("i", i, " = ", "13+10*i", 13+10*i+10)) end_if: end_for:
```

```
"i", 0, " = ", "13+10*i", 23
"i", 2, " = ", "13+10*i", 43
"i", 3, " = ", "13+10*i", 53
"i", 5, " = ", "13+10*i", 73
"i", 6, " = ", "13+10*i", 83
"i", 8, " = ", "13+10*i", 103
"i", 9, " = ", "13+10*i", 113
"i", 14, " = ", "13+10*i", 163
"i", 15, " = ", "13+10*i", 173
"i", 17, " = ", "13+10*i", 193
"i", 20, " = ", "13+10*i", 223
"i", 21, " = ", "13+10*i", 233
"i", 24, " = ", "13+10*i", 263
"i", 26, " = ", "13+10*i", 283
"i", 27, " = ", "13+10*i", 293
"i", 29, " = ", "13+10*i", 313
```

Anmerkung:

1. In der *Folge 2* sind insgesamt 16 Primzahlen SF_p, EZ3 situiert.

-

6. Die Bestimmung der Zwillinge mod 20, die in der *Folge 2* situiert sind:

-
```
for i from 0 to 30 do: if (isprime (13+10*i-10) and isprime (13+10*i+10))
then print (("i", i, 13+10*i-10," = ", "13+10*i", 13+10*i+10)) end_if: end_for:
```

```
"i", 0, 3, " = ", "13+10*i", 23          ... OG I, OG I
"i", 2, 23, " = ", "13+10*i", 43         ... OG II, OG II
"i", 5, 53, " = ", "13+10*i", 73         ... OG I, OG I
"i", 8, 83, " = ", "13+10*i", 103        ... OG II, OG II
"i", 17, 173, " = ", "13+10*i", 193      ... OG I, OG I
"i", 26, 263, " = ", "13+10*i", 283      ... OG II, OG II
"i", 29, 293, " = ", "13+10*i", 313      ... OG I, OG I
```

Anmerkung:

1. In der *Folge 2* sind insgesamt 7 Zwillinge mod 20 situiert.
2. Die Primzahlen (SM_p, SF_p), die diese Zwillinge mod 20 bilden haben allesamt die Endziffer EZ3 und sind stets Elemente der gleichen Obergruppe.

---

7. Die Bestimmung der Primzahlen SM_p, die in der *Folge 3* situiert sind:

-
```
for i from 0 to 30 do: if (isprime (17+10*i-10) and isprime (17+10*i-10))
  then print (("i", i," = ", "17+10*i", 17+10*i-10)) end_if: end_for:
```

```
"i", 0, " = ", "17+10*i", 7
```

```
"i", 1, " = ", "17+10*i", 17
"i", 3, " = ", "17+10*i", 37
"i", 4, " = ", "17+10*i", 47
"i", 6, " = ", "17+10*i", 67
"i", 9, " = ", "17+10*i", 97
"i", 10, " = ", "17+10*i", 107
"i", 12, " = ", "17+10*i", 127
"i", 13, " = ", "17+10*i", 137
"i", 15, " = ", "17+10*i", 157
"i", 16, " = ", "17+10*i", 167
"i", 19, " = ", "17+10*i", 197
"i", 22, " = ", "17+10*i", 227
"i", 25, " = ", "17+10*i", 257
"i", 27, " = ", "17+10*i", 277
"i", 30, " = ", "17+10*i", 307
```

<u>Anmerkung:</u>

1. In der *Folge 3* sind insgesamt 16 Primzahlen SM_p, EZ7 situiert.

---

8. Die Bestimmung der Primzahlen SF_p, die in der *Folge 3* situiert sind:

- ```
  for i from 0 to 30 do: if (isprime (17+10*i+10) and isprime (17+10*i+10))
     then print (("i", i," = ", "17+10*i", 17+10*i+10)) end_if: end_for:
  ```

```
"i", 1, " = ", "17+10*i", 37
"i", 2, " = ", "17+10*i", 47
"i", 4, " = ", "17+10*i", 67
"i", 7, " = ", "17+10*i", 97
"i", 8, " = ", "17+10*i", 107
"i", 10, " = ", "17+10*i", 127
"i", 11, " = ", "17+10*i", 137
"i", 13, " = ", "17+10*i", 157
"i", 14, " = ", "17+10*i", 167
"i", 17, " = ", "17+10*i", 197
"i", 20, " = ", "17+10*i", 227
"i", 23, " = ", "17+10*i", 257
"i", 25, " = ", "17+10*i", 277
"i", 28, " = ", "17+10*i", 307
"i", 29, " = ", "17+10*i", 317
```

<u>Anmerkung:</u>

1. In der *Folge 3* sind insgesamt 15 Primzahlen SF_p, EZ7 situiert.

---

9. Die Bestimmung der Zwillinge mod 20, die in der *Folge 3* situiert sind:

- ```
  for i from 0 to 30 do: if (isprime (17+10*i-10) and isprime (17+10*i+10))
  then print (("i", i, 17+10*i-10, " = ", "17+10*i", 17+10*i+10)) end_if: end_for:
  ```

```
"i", 1, 17, " = ", "17+10*i", 37        ... OG I, OG I
"i", 4, 47, " = ", "17+10*i", 67        ... OG II, OG II
```

```
"i", 10, 107, " = ", "17+10*i", 127          ... OG II, OG II
"i", 13, 137, " = ", "17+10*i", 157          ... OG I, OG I
"i", 25, 257, " = ", "17+10*i", 277          ... OG I, OG I
```

Anmerkung:

1. In der *Folge 3* sind insgesamt 5 Zwillinge mod 20 situiert.
2. Die Primzahlen (SM_p, SF_p), die diese Zwillinge mod 20 bilden, sind stets Elemente der gleichen Obergruppen.

---

10. Die Bestimmung der Primzahlen SM_p, die in der *Folge 4* situiert sind:

- ```
  for i from 0 to 30 do: if (isprime (19+10*i-10) and isprime (19+10*i-10))
    then print (("i", i, " = ", "19+10*i", 19+10*i-10)) end_if: end_for:
  ```

```
"i", 1, " = ", "19+10*i", 19
"i", 2, " = ", "19+10*i", 29
"i", 5, " = ", "19+10*i", 59
"i", 7, " = ", "19+10*i", 79
"i", 8, " = ", "19+10*i", 89
"i", 10, " = ", "19+10*i", 109
"i", 13, " = ", "19+10*i", 139
"i", 14, " = ", "19+10*i", 149
"i", 17, " = ", "19+10*i", 179
"i", 19, " = ", "19+10*i", 199
"i", 22, " = ", "19+10*i", 229
"i", 23, " = ", "19+10*i", 239
"i", 26, " = ", "19+10*i", 269
```

Anmerkung:

1. In der *Folge 4* sind insgesamt 13 Primzahlen SM_p, EZ9 situiert.

---

11. Die Bestimmung der Primzahlen SF_p, die in der *Folge 4* situiert sind:

- ```
  for i from 0 to 30 do: if (isprime (19+10*i+10) and isprime (19+10*i+10))
    then print (("i", i, " = ", "19+10*i", 19+10*i+10)) end_if: end_for:
  ```

```
"i", 0, " = ", "19+10*i", 29
"i", 3, " = ", "19+10*i", 59
"i", 5, " = ", "19+10*i", 79
"i", 6, " = ", "19+10*i", 89
"i", 8, " = ", "19+10*i", 109
"i", 11, " = ", "19+10*i", 139
"i", 12, " = ", "19+10*i", 149
"i", 15, " = ", "19+10*i", 179
"i", 17, " = ", "19+10*i", 199
"i", 20, " = ", "19+10*i", 229
"i", 21, " = ", "19+10*i", 239
"i", 24, " = ", "19+10*i", 269
```

Anmerkung:
1. In der *Folge 4* sind insgesamt 12 Primzahlen SF_p, EZ9 situiert.

---

12. Die Bestimmung der Zwillinge mod 20, die in der *Folge 4* situiert sind:

-
```
for i from 0 to 30 do: if (isprime (19+10*i-10) and isprime (19+10*i+10))
then print (("i", i, 19+10*i-10, " = ", "19+10*i", 19+10*i+10)) end_if: end_for:
```

```
"i", 5, 59, " = ", "19+10*i", 79        ... OG II, OG II
"i", 8, 89, " = ", "19+10*i", 109       ... OG I, OG I
"i", 17, 179, " = ", "19+10*i", 199     ... OG II, OG II
```

Anmerkung:
1. In der *Folge 4* sind insgesamt 3 Zwillinge mod 20 situiert.
2. Die Primzahlen (SM_p, EZ9 und SF_p, EZ9), die diese Zwillinge bilden, sind stets Elemente der gleichen Obergruppen.

Zusammenfassung der Ergebnisse und Schlussfolgerungen:
1. In den vier Folgen sind 61 Primzahlen vom Typ SM_p und 48 Primzahlen vom Typ SF_p situiert.
2. In den vier Folgen sind 20 Zwillinge mod 20 situiert.
3. Die Ursache dafür, dass die Zahl der Zwillinge mod 20 geringer ist als die Zahl 48 der Primzahlen vom Typ SF_p liegt darin begründet, dass nicht alle Primzahlen SM_p und SF_p das "Und-Kriterium"

   (SM_p +10) < = > (SF_p -10) bzw. das Kriterium: SM_p +20 = SF_p erfüllen.

---

1. Die Bestimmung der Primzahlen SM_p, die in der *Folge 1* im Intervall i:= 10^350 bis 10^350 +2000 situiert sind:

-
```
for i from 10^350 to 10^350+2000 do: if (isprime (11+10*i-10) and isprime (11+10*i-10))
then print (("i", " = ", "11+10*i-10", 11+10*i-10)) end_if: end_for:
```

```
"i", " = ", "11+10*i-10",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000501

"i", " = ", "11+10*i-10",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000008601

"i", " = ", "11+10*i-10",

0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000009891

"i", " = ", "11+10*i-10",

0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000011721
```

```
"i", " = ", "11+10*i-10",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000011931

"i", " = ", "11+10*i-10",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012001

"i", " = ", "11+10*i-10",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012021

"i", " = ", "11+10*i-10",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012301

"i", " = ", "11+10*i-10",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012621
```

<u>Anmerkung:</u>

1. In der *Folge 1* sind in dem um Zahl 10^350 verschobenen und von Δ= 30 auf Δ= 2000 vergrößerten Intervall insgesamt 9 Primzahlen: SM_p, EZ1 situiert.

2. Die Abstände, in denen diese Primzahlen auftreten, ergibt sich aus ihrer "geordneten" Auflistung.

3. Die Zahl der Primzahlen SM_0, EZ1 erhöht sich offensichtlich durch eine "Intervall-Vergrößerung"

---

2. Die Bestimmung der in *der Folge* 1 situierten Primzahlen SF_p:

- for i from 10^350 to 10^350+2000 do: if (isprime (11+10*i+10) and isprime (11+10*i+10)) then print (("i", " = ", "11+10*i+10", 11+10*i+10)) end_if: end_for:

```
"i", " = ", "11+10*i+10",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000501

"i", " = ", "11+10*i+10",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000008601

"i", " = ", "11+10*i+10",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000009891
```

```
"i", " = ", "11+10*i+10",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000011721

"i", " = ", "11+10*i+10",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000011931

"i", " = ", "11+10*i+10",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012001

"i", " = ", "11+10*i+10",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012021

"i", " = ", "11+10*i+10",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012301

"i", " = ", "11+10*i+10",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012621
```

<u>Anmerkung:</u>

1. In der *Folge 1* sind insgesamt 9 Primzahlen SF_p, EZ1 situiert.

---

3. Die Bestimmung der in der *Folge 1* situierten Zwillinge mod 20:

- ```
  for i from 10^350 to 10^350+2000 do: if (isprime (11+10*i-10) and isprime (11+10*i+10))
      then print (("i",11+10*i-10, " = ", "11+10*i+10", 11+10*i+10)) end_if: end_for:
  ```

```
"i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012001                                    ... OG I

  , " = ", "11+10*i+10",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000012021                                    ... OG I
```

<u>Anmerkung:</u>

1. In der *Folge 1* ist 1 Zwilling mod 20 situiert: (SM_p, EZ1, SF_p, EZ1), dessen Primzahlen
```

Elemente der Obergruppe OG I sind.

---

4. Die Bestimmung der in der *Folge 2* situierten Primzahlen SM_p:

- `for i from 10^350 to 10^350+2000 do: if (isprime (13+10*i-10) and isprime (13+10*i-10)) then print (("i",-10," = ", "13+10*i", 13+10*i-10)) end_if: end_for:`

```
"i", -10, " = ", "13+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000004953

"i", -10, " = ", "13+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000007413

"i", -10, " = ", "13+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000010183

"i", -10, " = ", "13+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000015903
```

Anmerkung:

1. In der *Folge 2* sind 4 Primzahlen SM_p, EZ3 situiert.

---

5. Die Bestimmung der in der *Folge 2* situierten Primzahlen SF_p:

- `for i from 10^350 to 10^350+2000 do: if (isprime (13+10*i+10) and isprime (13+10*i+10)) then print (("i",+10," = ", "13+10*i", 13+10*i+10)) end_if: end_for:`

```
"i", 10, " = ", "13+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000004953

"i", 10, " = ", "13+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000007413

"i", 10, " = ", "13+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000010183

"i", 10, " = ", "13+10*i",

0000000000000000000000000000000000000000000000000000000000000000000000000000000\
```

```
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000015903
```

<u>Anmerkungen:</u>

1. In der *Folge 2* sind insgesamt 4 Primzahlen SF_p, EZ3 situiert.

-

6. Die Bestimmung der in der *Folge 2* situierten Zwillinge mod 20:

<u>Anmerkung:</u>

1. Im Intervall i:= 10^350 bis 10^350 +2000 ist kein Zwilling mod 2 in der *Folge 2* situiert, dagegen aber im Intervall i:= 10^350 +16000 bis 10^350 + 18000. Dies ist der folgende Zwilling mod 20:

- ```
  for i from 10^350+16000 to 10^350+18000 do: if (isprime (13+10*i-10) and isprime
  (13+10*i+10))then print (("i", 13+10*i-10, " = ", "13+10*i", 13+10*i+10)) end_if: end_for:
  ```

```
"i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000178813                                ...   OG I

, " = ", "13+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000178833                                ...   OG I
```

---

7. Die Bestimmung der in der *Folge 3* situierten Primzahlen SM_p:

- ```
  for i from 10^350 to 10^350+2000 do: if (isprime (17+10*i-10) and isprime (17+10*i-10))
     then print (("i",-10," = ", "17+10*i", 137+10*i+-10)) end_if: end_for:
  ```

```
"i", -10, " = ", "17+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000327

"i", -10, " = ", "17+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000001567

"i", -10, " = ", "17+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000006097

"i", -10, " = ", "17+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000006807

"i", -10, " = ", "17+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
```

```
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000009357

"i", -10, " = ", "17+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000013627

"i", -10, " = ", "17+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000018187

"i", -10, " = ", "17+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000019897
```

Anmerkung:

1. In der *Folge 3* sind insgesamt 8 Primzahlen SM_p, EZ7 situiert.

---

8. Die Bestimmung der in der *Folge 3* situierten Primzahlen SF_p:

-
```
for i from 10^350 to 10^350+2000 do: if (isprime (17+10*i+10) and isprime (17+10*i+10))
    then print (("i",+10," = ", "17+10*i", 137+10*i+10)) end_if: end_for:
```

```
"i", 10, " = ", "17+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000327

"i", 10, " = ", "17+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000001567

"i", 10, " = ", "17+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000006097

"i", 10, " = ", "17+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000006807

"i", 10, " = ", "17+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000009357

"i", 10, " = ", "17+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
```

```
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000013627

"i", 10, " = ", "17+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000018187

"i", 10, " = ", "17+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000019897
```

<u>Anmerkung:</u>

1. In der *Folge 3* sind insgesamt 8 Primzahlen SF_p, EZ7 situiert.

---

9. Die Bestimmung der in der *Folge 3* situierten Zwillinge mod 20

<u>Anmerkung</u>: Im Intervall i:= 10^350 bis 10^350+2000 ist in der *Folge 3* kein Zwilling mod 20 situiert.

---

10. Die Bestimmung der Primzahlen SM_p, die in der *Folge 4* situiert sind:

- `for i from 10^350 to 10^350+2000 do: if (isprime (19+10*i-10) and isprime (19+10*i-10))`
  `then print (("i", -10, " = " , "19+10*i", 19+10*i-10)) end_if: end_for:`

```
"i", -10, " = ", "19+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000249

"i", -10, " = ", "19+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000004879

"i", -10, " = ", "19+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000008059

"i", -10, " = ", "19+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000010119

"i", -10, " = ", "19+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000010939

"i", -10, " = ", "19+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
```

```
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000012549

"i", -10, " = ", "19+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000014809
```

Anmerkung:

1. In der *Folge 4* sind insgesamt 7 Primzahlen SM_p, EZ9 situiert.

---

11. Die Bestimmung der in der *Folge 4* situierten Primzahlen SF_p:

- `for i from 10^350 to 10^350+2000 do: if (isprime (19+10*i+10) and isprime (19+10*i+10)) then print (("i", +10, " = " , "19+10*i", 19+10*i+10)) end_if: end_for:`

```
"i", 10, " = ", "19+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000249

"i", 10, " = ", "19+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000004879

"i", 10, " = ", "19+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000008059

"i", 10, " = ", "19+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000010119

"i", 10, " = ", "19+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000010939

"i", 10, " = ", "19+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000012549

"i", 10, " = ", "19+10*i",

10000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000014809
```

Anmerkung:

1. In der *Folge 4* sind insgesamt 7 Primzahlen SF_p, EZ9 situiert.

---

12. Die Bestimmung der in der *Folge 4* situierten Zwillinge mod 20:

Anmerkung:

1. Im Intervall i:= 10^350 bis 10^350 + 20000 ist in der *Folge 4* kein Zwilling mod 20 situiert.

2. In *der Folge* 4 ist dagegen der nachfolgende Zwilling mod 20 im Intervall i:= 10^350+20000 bis 10^350 + 22000 situiert!

3. Die Primzahlen (SM_p, EZ9, SF_p, EZ9) sind beide Elemente der Obergruppe OG I.

- ```
  for i from 10^350+20000 to 10^350+22000 do: if (isprime (19+10*i-10) and isprime
  (19+10*i+10)) then print (("i", -10,19+10*i-10, " = " , "19+10*i", 19+10*i+10)) end_if:
  end_for:
  ```

```
"i", -10,

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000212989                                    ...  OG I

 , " = ", "19+10*i",

1000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000213009                                    ...  OG I
```

```

Anlage 11

Die erweiterten Beweisgleichungen des Hauptsatzes der Primzahl-Theorie führen zu drei Methoden für die Bestimmung von Primzahlen und von Zwillingen mod 2n, n ≥1:

Methode 1:
Nach dieser Methode wird eine gerade oder ungerade Zahl vorgegeben und nach Primzahlen gesucht, die in einem Intervall auftreten, in dem die (vorgegebene) Zahl situiert ist. Die Primzahl-Abstände in denen dann (fallweise) Primzahlen auftreten, werden aus Primzahlen (oder ihren Produkten) gebildet, die in der Faktor-Zerlegung der vorgegebenen Zahl nicht auftreten.

Methode 2:
In dieser Methode wird eine prime Zahl (oder ein Produkt primer Zahlen) vorgegeben, die in den Folgen: n_g ^i bzw. n_u^i in ihren Faktor-Zerlegungen nicht auftreten. Auf dieser Methode basiert der GIMPs-Test zur Auffindung der Mersenne-Primzahlen M_p.

Methode 3:
In dieser Methode werden Primzahlen gesucht, die definierte Zwillinge mod 2n, n ≥1 bilden.

Diese Methoden werden an den folgenden Beispielen mit dem MuPAD - Test demonstriert:

Beispiele für die Methode 1:

1. Die Bestimmung der Primzahlen, die im Intervall i:= -5000 bis +5000 in der Umgebung der geraden Zahl n_g = 2^1000 situiert sind.

-
```
for i from -5000 to +5000 do: if isprime (2^1000+i)
then print ("i", i, " = ", 2^1000+i) end_if: end_for:
```

```
"i", -3687, " = ",

10715086071862673209484250490600018105614048117055336074437503883703510511249361\
224931983788156958581275946729175531468251871452856923140435984577574698574803\
934567774824230985421074605062371141877954182153046474983581941267398767559165\
543946077062914571196477686542167660429831652624386837205668065689

"i", -3683, " = ",

10715086071862673209484250490600018105614048117055336074437503883703510511249361\
224931983788156958581275946729175531468251871452856923140435984577574698574803\
934567774824230985421074605062371141877954182153046474983581941267398767559165\
543946077062914571196477686542167660429831652624386837205668065693

"i", -3395, " = ",

10715086071862673209484250490600018105614048117055336074437503883703510511249361\
224931983788156958581275946729175531468251871452856923140435984577574698574803\
934567774824230985421074605062371141877954182153046474983581941267398767559165\
543946077062914571196477686542167660429831652624386837205668065981

"i", -2793, " = ",

10715086071862673209484250490600018105614048117055336074437503883703510511249361\
224931983788156958581275946729175531468251871452856923140435984577574698574803\
934567774824230985421074605062371141877954182153046474983581941267398767559165\
543946077062914571196477686542167660429831652624386837205668066583

"i", -2523, " = ",

10715086071862673209484250490600018105614048117055336074437503883703510511249361\
224931983788156958581275946729175531468251871452856923140435984577574698574803\
934567774824230985421074605062371141877954182153046474983581941267398767559165\
```

```
43946077062914571196477686542167660429831652624386837205668066853

"i", -1245, " = ",

1071508607186267320948425049060001810561404811705533607443750388370351051124936\
1224931983788156958581275946729175531468251871452856923140435984577574698574803\
9345677748242309854210746050623711418779541821530464749835819412673987675591655\
43946077062914571196477686542167660429831652624386837205668068131

"i", 297, " = ",

1071508607186267320948425049060001810561404811705533607443750388370351051124936\
1224931983788156958581275946729175531468251871452856923140435984577574698574803\
9345677748242309854210746050623711418779541821530464749835819412673987675591655\
43946077062914571196477686542167660429831652624386837205668069673

"i", 4081, " = ",

1071508607186267320948425049060001810561404811705533607443750388370351051124936\
1224931983788156958581275946729175531468251871452856923140435984577574698574803\
9345677748242309854210746050623711418779541821530464749835819412673987675591655\
43946077062914571196477686542167660429831652624386837205668073457
```

Anmerkung:

1. Im vorgegebenen Intervall sind 8 Primzahlen situiert.

2. Die Abstände, in denen diese Primzahlen auftreten, bestehen aus Faktoren ungerader Primzahlen, die nachfolgend angegeben werden, oder aus einer Primzahl::

$3687 = 3 \cdot 1229, \quad 3683 = 29 \cdot 127, \quad 3395 = 5 \cdot 7 \cdot 97, \quad 2793 = 3 \cdot 7^2 \cdot 19$

$2523 = 3 \cdot 29^2, \quad 1245 = 3 \cdot 5 \cdot 83, \quad 297 = 3^2 \cdot 11, \quad 4091 = 4091$

-

2. Die Bestimmung der Primzahlen im Intervall i:= -300 bis +300 in der Umgebung der Zahl $n_8 = 53!$

```
• for i from -300 to +300 do: if isprime (53!+i) then print
  ("i", i, " = ", 53!+i) end_if: end_for:

"i", -281, " = ",

4274883284060025564298013753389399649690343788366813724671999999999999719

"i", -229, " = ",

4274883284060025564298013753389399649690343788366813724671999999999999771

"i", -181, " = ",

4274883284060025564298013753389399649690343788366813724671999999999999819

'i", -151, " = ",

4274883284060025564298013753389399649690343788366813724671999999999999849

i", -59, " = ",

4274883284060025564298013753389399649690343788366813724671999999999999941

i", 71, " = ",

4274883284060025564298013753389399649690343788366813724672000000000000071

i", 79, " = ",

4274883284060025564298013753389399649690343788366813724672000000000000079
```

nmerkung:

Im vorgegebenen Intervall sind 7 Primzahlen situiert.

Alle diese Primzahlen treten in Primzahl-Abständen $p > 53$ auf.

---

3. Die Bestimmung der Primzahlen im Intervall i:= -300 bis +300 in der Umgebung der Zahl $n_g = 101!$

- ```
  for i from -300 to +300 do: if isprime (101!+i) then print
  ("i", i, " = ", 101!+i) end_if: end_for:
  ```

```
"i", 139, " = ",

942594775983835942085162312448293674956231279470254376832788935341697759931622\
476503087861591808346911623490003549599583369706302603264000000000000000000000001\
39

"i", 233, " = ",

942594775983835942085162312448293674956231279470254376832788935341697759931622\
476503087861591808346911623490003549599583369706302603264000000000000000000000002\
33
```

Anmerkung:

1. In diesem Intervall sind (nur) zwei Primzahlen situiert, die in primen Abständen $p > 101$ auftreten.
2. Ein Vergleich mit den 7 Primzahlen, die im gleichen Intervall in der Umgebung der Zahl $n_g = 53!$ auftreten, zeigt die Auswirkung des Asymptote-Gesetzes.

---

4. Die Bestimmung der im Intervall i:= -300 bis +300 um die Zahl $n_g = 101!\verb|^|2$ situierten Primzahlen:

- ```
  for i from -300 to +300 do: if isprime (101!^2-i) then print
  ("i", i, " = ", 101!^2+i) end_if: end_for:
  ```

```
"i", -191, " = ",

88848491171201786290128636241995700314676720288840510209136727802218719342731\
650728114234969043294549648657415540221586639211628605782186404497287125789810\
391004524288975261495029961462271783759841448119431843989644342256109220993313\
187069627181678624002095033834536959999999999999999999999999999999999999999999\
9809
```

Anmerkung:

1. Infolge des Asymptote-Gesetzes ist hier im gleich großen Intervall nur eine Primzahl situiert, die im Primzahl-Abstand $p = 191 > 101$ auftritt.

Beispiele für die Methode 2:

l. Die Bestimmung der Primzahlen 2^i -5 , die in der Folge der Exponenten i:= 10 bis 1000 auftreten:

```
for i from 10 to 1000 do: if (isprime (2^i-5) and isprime (2^i -5))
 then print (("i", i, " = ", 2^i-5)) end_if: end_for:

"i", 10, " = ", 1019
"i", 12, " = ", 4091
"i", 18, " = ", 262139
"i", 20, " = ", 1048571
"i", 26, " = ", 67108859
"i", 32, " = ", 4294967291
"i", 36, " = ", 68719476731
"i", 56, " = ", 72057594037927931
```

```
"i", 66, " = ", 73786976294838206459

"i", 118, " = ", 332306998946228968225951765070086139

"i", 130, " = ", 1361129467683753853853498429727072845819

"i", 150, " = ", 1427247692705959881058285969449495136382746619

"i", 166, " = ", 93536104789177786765035829293842113257979682750459

"i", 206, " = ", 102844034832575377634685573909834406561420991602098741459288059

"i", 226, " = ",

107839786668602559178668060348078522694548577690162289924414440996859

"i", 550, " = ",

368551018048978647679839314549635633878605587931293010583613896508361734608 6082\
863365358130056307390177215209990980317284932211552660930305235775636164742 2301\
26362619

"i", 706, " = ",

336648697699095904463423352504328234595221747380683127007889977796398857875 0131\
726152746583217656608785269100067954059336332436640117634472401801453217203 7439\
71343149522204542842664805045966539053627681110900080 59

"i", 810, " = ",

682804677926897077665776823369856798427623252205148995624576799924637413152 5255\
241237946573692471593720760228447675638216590132893818229842292076597586769 9672\
014413614223509078661921647736975613224601765231478804292513259903758963621 5181\
0433019
```

Anmerkung:

1. Im Exponenten-Intervall i:= 10 bis 1000 sind 18 Primzahlen 2^i -5 situiert.

---

2. Die Bestimmung der Primzahlen 2^i -3 in der Exponenten-Folge i:= 1000 bis 3000:

-
```
for i from 1000 to 3000 do: if isprime (2^i-3)
then print ("i", i, " = ", 2^i-3) end_if: end_for:
```

```
"i", 1736, " = ",

387322273736962452994368291567341234951493599483280178852314556259314267226 1417\
519398456218547242273534524437025619137650161277833770996636763023198929825 4814\
399869627019245750086484117926509068042826311149816702153054493621720003300 2962\
929005595995829408109584013092439494004738650821651864095664603758025337703 8440\
880569218953826149700308242340599431626053471830468600503629408257907183468 4490\
207680412889699071097187304845470330614536721317220811871474259006621638179 3007\
529459922883136643240445571571682438840848 0628733

"i", 2321, " = ",

490478456129733713247338466317215001375660942757406424347083858734583153227 0898\
597682117935292650637723829964693565462392669052134557803176158680063712372 9942\
309238844979834869479493240845611404187770780758649939383976533841270267234 4522\
485649173285437907304343243175371730985872642564205569499444443268468650883 6848\
664087629534553644227206429984970907622746021306057769426213516663486194201 3294\
307055539173170755322938036795885753189540873551038978919537776561588534585 3350\
010100644160488698736417355050551871432459938148887113219000371731702876258 4717\
332938926715722192746653833959452902014238660785401301083462763045074955485 2678\
752000476951087518271174956306970994042229793910010099598334571314 9
```

Anmerkung:

1. In dem drei mal größeren Exponenten-Intervall als im 1. Beispiel, sind hier nur 2 Primzahlen vom Typ p = 2^i - 3 situiert.

2. Die Ursachen für diesen Sachverhalt sind auf das Asymptote-Gesetz zurückzuführen.

---

3. Die Bestimmung der Primzahlen p = 3^i -3 im Intervall i:= 10 bis 1000:

- ```
  for i from 100 to 1000 do: if isprime (3^i-2)
  then print ("i", i, " = ", 3^i-2) end_if: end_for:
  ```

```
"i", 102, " = ", 4638397686588101979328150167890591454318967698007

"i", 105, " = ", 125236737537878753441860054533045969266612127846241

"i", 317, " = ",

1767818791893718942542512976226534840430462418663846347592909659799155506964877\
0683672843412379273268176113845487788768161966012911937393135864623398161

"i", 520, " = ",

1267804982496475720068841448851704713044890168618794386177607513832290843232323\
3825711770059882613859415333016250177953007331217321384542514056724358093860403\
8739769427385468006506866897314769910412165745223337576718744062067606378331982\
456148394399

"i", 541, " = ",

1326168790943636873463383702999509006710931275809481594345135568419247032683280\
4768010205770069260168834737042384420000006022058158963387968160292916287523165\
0298028321323305617751812999082122553158792100321382117098017267978611718212818\
2482511664415807616401

"i", 561, " = ",

4624064653355303120600537140296304915258679488675529071059228510194498781725599\
5400176409287877689046303587466322871501453417618450588319497492394008102134602\
7520652151846317705732690254668572236324369571402622252657721998016460140700262\
2662503126785584774690972129601

"i", 648, " = ",

1494765475221008241645283557110864397393812827996524345650616731378808040818315\
7009249398937882019326429587913485807796030583036458623082206914234681554285074\
7072648261237189289192113376244283886017003350833174247899723851336982872865238\
0513166152898823769632161530244241015606671172212868060723071835178515359

"i", 780, " = ",

1427508429116232068654784609302249106798879954505198077794702832678487479931968\
6049497096339220428456407440491557197591921914261306909162962431222668681667893\
6483091023375678370433086519315754438306372648397974591088603923156803059715122\
0443733280015473273500173909985231407430800908445815801892176220466645947134059\
212337189541617821947255374674878585650999364069530351599

"i", 786, " = ",

1040653644825733178049337980181339598856383486834289398712338365022617372870405\
1130083383231291692344721024118345197044511075496492736779799612361325468935894\
4696173356040869532045720072581184985525345660682123476903592259981309430532323\
9703481561131280016381626780379233696017053862256999719579396464720184895460729\
165793811175839392199549168137986488939578536406687626317127

"i", 957, " = ",

4027548386785007571117173292795657383359941164158052317894419985866105130353285\
5005585948717494491600126593662167970061216275506091264645530297111568669513274\
7453879691658758515523562337864765346465031722639562539176303797085856959058416\
6150099107891445876651308516534118747449004540517470731222098020245058472977775\
7505019030876954114584461550192892878866267114205001734295703633158413710567740\
68276407607294805856129475397488521110313652022778742898508561
```

<u>Anmerkung:</u>

1. In diesem Exponenten-Intervall sind 10 Primzahlen p = 3^i -2 situiert.

```

4. Die Bestimmung der Primzahlen p = 3^i +2 im Exponenten-Intervall i:= 100 bis 1000:

```
for i from 100 to 1000 do: if isprime (3^i+2)
  then print ("i", i, " = ", 3^i+2) end_if: end_for:
```

```
"i", 110, " = ", 30432527221704537086371993251530170531786747066637051

"i", 123, " = ", 48519278097689642681155855396759336072749841943521979872829

"i", 126, " = ", 1310020508637620352391208095712502073964245732475093456566331

"i", 139, " = ",

2088595827392656793085408064780643444068898148936888424953199350269

"i", 235, " = ",

1328907826336853501035071949182453462840482737459712697538869371101828487939695\
7772210874852435860862184279107709

"i", 243, " = ",

8718964248596095820291107058586077169696407240473175008552521943799096709372343\
9943475549906831683116791055225665629

"i", 315, " = ",

1964243102104132158380569973585038711589402687404273719547677399776839452183196\
742630315934708808140908457093943087640906885112545770821459540513710909

"i", 363, " = ",

1566806855931287351164694142635079311519424524552371870753453068361209465085661\
8485704651360089843444750887289934688702552379488463219583816563323032694515603\
7679213625378429

"i", 386, " = ",

1475041780253088107250209473988537166277012240990633971553468634424811985015644\
7628362069360986269022056164766032427956749652735167596646914112360848363277501\
476652958394736238608965131

"i", 391, " = ",

3584351526015004100618009021792145314053139745607240550874928781652293123588016\
7736919828547196633723596480381458799934901656146457259852001293036861522764328\
58826668889920905981978526349

"i", 494, " = ",

4987694347855958414593315100074529371470971882853446233243626314482935554401083\
0568173536711905883104675143624875725453707621305235957566725641194672924589547\
354967418172350411694571792271348710524815324540675690984922340178518237829371
```

<u>Anmerkung:</u>

1. In diesem Exponenten-Intervall sind 11 Primzahlen p = 3^1 +2 situiert.

<u>Beispiele für die Methode 3:</u>

1. Die Bestimmung von Zwillingen mod 2 im Intervall i:= 1 bis 50:

```
for i from 1 to 50 do: if (isprime (30*i-1)and isprime (30*i-1))
  then print (("i", i,30*i-1, " = ", 30*i+1)) end_if: end_for:
```

```
"i", 1, 29, " = ", 31

"i", 2, 59, " = ", 61

"i", 3, 89, " = ", 91

"i", 5, 149, " = ", 151

"i", 6, 179, " = ", 181
```

```
"i", 8, 239, " = ", 241
"i", 9, 269, " = ", 271
"i", 12, 359, " = ", 361
"i", 13, 389, " = ", 391
"i", 14, 419, " = ", 421
"i", 15, 449, " = ", 451
"i", 16, 479, " = ", 481
"i", 17, 509, " = ", 511
"i", 19, 569, " = ", 571
"i", 20, 599, " = ", 601
"i", 22, 659, " = ", 661
"i", 24, 719, " = ", 721
"i", 27, 809, " = ", 811
"i", 28, 839, " = ", 841
"i", 31, 929, " = ", 931
"i", 34, 1019, " = ", 1021
"i", 35, 1049, " = ", 1051
"i", 37, 1109, " = ", 1111
"i", 41, 1229, " = ", 1231
"i", 42, 1259, " = ", 1261
"i", 43, 1289, " = ", 1291
"i", 44, 1319, " = ", 1321
"i", 47, 1409, " = ", 1411
"i", 48, 1439, " = ", 1441
"i", 50, 1499, " = ", 1501
```

Anmerkung:

1. In diesem Intervall sind Zwillinge mod 2 situiert.

2. Diese Methode unterscheidet die Zwillinge mod 2 (jedoch) nicht nach der Zugehörigkeit zu den Folgen.

---

2. Die Bestimmung der Zwillinge mod 2 im Intervall i;= 1000 bis 1050:

- ```
  for i from 1000 to 1050 do: if (isprime (30*i-1) and isprime (30*i+1))
    then print (("i", i, 30*i-1, " = ", 30*i+1)) end_if: end_for:
  ```

```
"i", 1003, 30089, " = ", 30091
"i", 1009, 30269, " = ", 30271
"i", 1013, 30389, " = ", 30391
"i", 1028, 30839, " = ", 30841
"i", 1029, 30869, " = ", 30871
"i", 1036, 31079, " = ", 31081
"i", 1044, 31319, " = ", 31321
```

Anmerkung:

1. In diesem gleichen, aber im Zahlenbereich verschobenen Intervall sind nur 7 Zwillinge mod 2 situiert.
```

---

3. Die Bestimmung der Zwillinge mod 2 im Intervall i:= 3000 bis 3050:

- ```
  for i from 3000 to 3050 do: if (isprime (30*i-1) and isprime (30*i+1))
     then print (("i", i, 30*i-1, " = ", 30*i+1)) end_if: end_for:
  ```

```
"i", 3036, 91079, " = ", 91081
"i", 3038, 91139, " = ", 91141
```

Anmerkung:

1. Im gleichen, jedoch noch weiter im Zahlenbereich verschobenen Intervall sind nur 2 Zwillinge mod 2 situiert.
2. Dieser Sachverhalt entspricht dem Asymptote-Gesetz.

---

4. Die Bestimmung der im Intervall i:= 1 bis 50 situierten Zwillinge mod 6:

- ```
  for i from 1 to 50 do: if (isprime (10*i-3)and isprime (10*i+3))
     then print (("i", i,10*i-3, " = ", 10*i+3)) end_if: end_for:
  ```

```
"i", 1, 7, " = ", 13
"i", 2, 17, " = ", 23
"i", 4, 37, " = ", 43
"i", 5, 47, " = ", 53
"i", 7, 67, " = ", 73
"i", 10, 97, " = ", 103
"i", 11, 107, " = ", 113
"i", 16, 157, " = ", 163
"i", 17, 167, " = ", 173
"i", 23, 227, " = ", 233
"i", 26, 257, " = ", 263
"i", 28, 277, " = ", 283
"i", 31, 307, " = ", 313
"i", 35, 347, " = ", 353
"i", 37, 367, " = ", 373
"i", 46, 457, " = ", 463
```

Anmerkung:

1. In diesem Intervall sind 16 Zwillinge mod 6 situiert.

---

5. Die Bestimmung der Zwillinge mod 6, die im verschobenen Intervall situiert sind:

- ```
  for i from 3000 to 3050 do: if (isprime (10*i-3)and isprime (10*i+3))
     then print (("i", i,10*i-3, " = ", 10*i+3)) end_if: end_for:
  ```

```
"i", 3010, 30097, " = ", 30103
"i", 3020, 30197, " = ", 30203
"i", 3031, 30307, " = ", 30313
```

Anmerkung:

1. Im Verschobenen (sonst gleichen) Intervall sind nur (noch) 3 Zwillinge mod 6 situiert.

2. Auch hier manifestiert sich die Wirkung des Asymptote-Gesetzes.
```

---

6. Die Bestimmung der Zwillinge mod 6 im Intervall i:= 1 bis 100 in der Umgebung der Zahl 3*i: :

- ```
  for i from 1 to 100 do: if (isprime (3*i-2) and isprime (3*i+4))
      then print (("i", i, 3*i-2, " = ", 3*i+4)) end_if: end_for:
  ```

```
"i", 3, 7, " = ", 13
"i", 5, 13, " = ", 19
"i", 11, 31, " = ", 37
"i", 13, 37, " = ", 43
"i", 21, 61, " = ", 67
"i", 23, 67, " = ", 73
"i", 25, 73, " = ", 79
"i", 33, 97, " = ", 103
"i", 35, 103, " = ", 109
"i", 51, 151, " = ", 157
"i", 53, 157, " = ", 163
"i", 65, 193, " = ", 199
"i", 75, 223, " = ", 229
"i", 91, 271, " = ", 277
"i", 93, 277, " = ", 283
```

Anmerkung:

1. In diesem Zahlen-Intervall sind 15 Zwillinge mod 6 situiert.

---

7. Die Bestimmung der Zwillinge mod 6 im gleichen (im Zahlenbereich verschobenen) Intervall:

- ```
  for i from 1000 to 1100 do: if (isprime (3*i-2) and isprime (3*i+4))
      then print (("i", i, 3*i-2, " = ", 3*i+4)) end_if: end_for:
  ```

```
"i", 1021, 3061, " = ", 3067
"i", 1055, 3163, " = ", 3169
"i", 1061, 3181, " = ", 3187
"i", 1085, 3253, " = ", 3259
```

Anmerkung:

1. Im gleichen (aber im Zahlenbereich verschobenen) Intervall sind nur 3. Zwillinge mod 6 situiert.
2. Auch hier zeigt sich die Wirkung des Asymptote-Gesetzes.

---

8. Die Bestimmung der in der Folge 2*i im Intervall i:= 1 bis 100 situierten Zwillinge mod 6:

- ```
  for i from 1 to 100 do: if (isprime (2*i-3) and isprime (2*i+3))
   then print (("i", i, 2*i-3, " = ", 2*i+3)) end_if: end_for:
  ```

```
"i", 4, 5, " = ", 11
"i", 5, 7, " = ", 13
"i", 7, 11, " = ", 17
"i", 8, 13, " = ", 19
"i", 10, 17, " = ", 23
"i", 13, 23, " = ", 29
```
```

```
"i", 17, 31, " = ", 37
"i", 20, 37, " = ", 43
"i", 22, 41, " = ", 47
"i", 25, 47, " = ", 53
"i", 28, 53, " = ", 59
"i", 32, 61, " = ", 67
"i", 35, 67, " = ", 73
"i", 38, 73, " = ", 79
"i", 43, 83, " = ", 89
"i", 50, 97, " = ", 103
"i", 52, 101, " = ", 107
"i", 53, 103, " = ", 109
"i", 55, 107, " = ", 113
"i", 67, 131, " = ", 137
"i", 77, 151, " = ", 157
"i", 80, 157, " = ", 163
"i", 85, 167, " = ", 173
"i", 88, 173, " = ", 179
"i", 97, 191, " = ", 197
"i", 98, 193, " = ", 199
```

<u>Anmerkung:</u>

1. In diesem Intervall sind 26 Zwillinge mod 6 situiert.

---

9. Die Bestimmung der im Intervall i:= 1000 bis 1100 situierten Zwillinge mod 6:

- ```
  for i from 1000 to 1100 do: if (isprime (2*i-3) and isprime (2*i+3))
    then print (("i", i, 2*i-3, " = ", 2*i+3)) end_if: end_for:
  ```

```
"i", 1000, 1997, " = ", 2003
"i", 1007, 2011, " = ", 2017
"i", 1033, 2063, " = ", 2069
"i", 1042, 2081, " = ", 2087
"i", 1043, 2083, " = ", 2089
"i", 1067, 2131, " = ", 2137
"i", 1070, 2137, " = ", 2143
```

<u>Anmerkung:</u>

1. In diesem (gleichen, jedoch im Zahlenbereich verschobenen) Intervall sind 7 Zwillinge mod 6 situiert.

---

10. Die Bestimmung der im Intervall i:= 10^9 bis 10^9+100 situierten Zwillinge mod 6:

- ```
  for i from 10^9 to 10^9+100 do: if (isprime (2*i-3) and isprime (2*i+3))
    then print (("i", i, 2*i-3, " = ", 2*i+3)) end_if: end_for:
  ```

```
"i", 1000000070, 2000000137, " = ", 2000000143
```

<u>Anmerkung:</u>

1. In diesem gleichen (im Zahlenbereich verschobenen) Intervall ist nur ein Zwilling mod 6 situiert.
2. Dieser Sachverhalt bestätigt das Asymptote-Gesetz.

---

11. Die Bestimmung der Zahl der Zwillinge mod 6 im Intervall $i := 10^9$ bis $10^9 + 2 \times 10^3$:

```
• for i from 10^9 to 10^9+2*10^3 do: if (isprime (2*i-3) and isprime (2*i+3))
    then print (("i", i, 2*i-3, " = ", 2*i+3)) end_if: end_for:

              "i", 1000000070, 2000000137, " = ", 2000000143
              "i", 1000000138, 2000000273, " = ", 2000000279
              "i", 1000000205, 2000000407, " = ", 2000000413
              "i", 1000000303, 2000000603, " = ", 2000000609
              "i", 1000000385, 2000000767, " = ", 2000000773
              "i", 1000000420, 2000000837, " = ", 2000000843
              "i", 1000000523, 2000001043, " = ", 2000001049
              "i", 1000000550, 2000001097, " = ", 2000001103
              "i", 1000000553, 2000001103, " = ", 2000001109
              "i", 1000000585, 2000001167, " = ", 2000001173
              "i", 1000000757, 2000001511, " = ", 2000001517
              "i", 1000000777, 2000001551, " = ", 2000001557
              "i", 1000000813, 2000001623, " = ", 2000001629
              "i", 1000001087, 2000002171, " = ", 2000002177
              "i", 1000001090, 2000002177, " = ", 2000002183
              "i", 1000001132, 2000002261, " = ", 2000002267
              "i", 1000001135, 2000002267, " = ", 2000002273
              "i", 1000001408, 2000002813, " = ", 2000002819
              "i", 1000001495, 2000002987, " = ", 2000002993
              "i", 1000001678, 2000003353, " = ", 2000003359
              "i", 1000001705, 2000003407, " = ", 2000003413
              "i", 1000001932, 2000003861, " = ", 2000003867
```

<u>Anmerkung:</u>

1. Im erweiterten Intervall sind 22 Zwillinge mod 6 situiert.

---

12. Die Bestimmung der im Intervall $i :=$ 1 bis 100 in der Zahlenmenge $2*i$ lokalisierten Zwillinge mod 2:

```
for i from 1 to 100 do: if (isprime (2*i-1) and isprime (2*i+1))
then print (("i", i, 2*i-1, " = ", 2*i+1)) end_if: end_for:

              "i", 2, 3, " = ", 5
              "i", 3, 5, " = ", 7
              "i", 6, 11, " = ", 13
              "i", 9, 17, " = ", 19
              "i", 15, 29, " = ", 31
              "i", 21, 41, " = ", 43
              "i", 30, 59, " = ", 61
```

"i", 36, 71, " = ", 73

"i", 51, 101, " = ", 103

"i", 54, 107, " = ", 109

"i", 69, 137, " = ", 139

"i", 75, 149, " = ", 151

"i", 90, 179, " = ", 181

"i", 96, 191, " = ", 193

"i", 99, 197, " = ", 199

Anmerkung:

1. In diesem Intervall sind 15 Zwillinge mod 2 situiert.
2. Die Zahlen: 2*i, i := 1 bis 100 umfassen alle geraden Zahlen von $n_g = 2$ bis 200.

---

13. Bestimmung der im Intervall i:= 1000 bis 1460 situierten Zwillinge mod 2:

- for i from 1000 to 1460 do: if (isprime (2*i-1) and isprime (2*i+1))
  then print (("i", 2*i-1, " = ", 2*i+1)) end_if: end_for:

"i", 2027, " = ", 2029

"i", 2081, " = ", 2083

"i", 2087, " = ", 2089

"i", 2111, " = ", 2113

"i", 2129, " = ", 2131

"i", 2141, " = ", 2143

"i", 2237, " = ", 2239

"i", 2267, " = ", 2269

"i", 2309, " = ", 2311

"i", 2339, " = ", 2341

"i", 2381, " = ", 2383

"i", 2549, " = ", 2551

"i", 2591, " = ", 2593

"i", 2657, " = ", 2659

"i", 2687, " = ", 2689

"i", 2711, " = ", 2713

"i", 2729, " = ", 2731

"i", 2789, " = ", 2791

"i", 2801, " = ", 2803

Anmerkung:
1. In diesem (erweiterten) Intervall sind 19 Zwillinge mod 2 situiert.

---

14. Bestimmung der im Intervall i:= 10000 bis 10720 situierten Zwillinge mod 2:

- for i from 10000 to 10720 do; if (isprime (2*i-1) and isprime (2*i+1))
  then print (("i", 2*i-1, " = ", 2*i+1)) end_if: end_for:

"i", 20021, " = ", 20023

```
"i", 20147, " = ", 20149
"i", 20231, " = ", 20233
"i", 20357, " = ", 20359
"i", 20441, " = ", 20443
"i", 20477, " = ", 20479
"i", 20507, " = ", 20509
"i", 20549, " = ", 20551
"i", 20639, " = ", 20641
"i", 20717, " = ", 20719
"i", 20747, " = ", 20749
"i", 20771, " = ", 20773
"i", 20807, " = ", 20809
"i", 20897, " = ", 20899
"i", 20981, " = ", 20983
"i", 21011, " = ", 21013
"i", 21017, " = ", 21019
"i", 21059, " = ", 21061
"i", 21191, " = ", 21193
"i", 21317, " = ", 21319
"i", 21377, " = ", 21379
```

Anmerkung:

1. In diesem (nochmals erweiterten) Intervall sind 21 Zwillinge mod 2 situiert.

-

15. Die Bestimmung der im Intervall i:= 10000 bis 10720 situierten Zwillinge mod 2:

- ```
  for i from 10000 to 10720 do; if (isprime (2*i-1) and isprime (2*i+1))
     then print (("i", 2*i-1, " = ", 2*i+1)) end_if: end_for:
  ```

```
"i", 20021, " = ", 20023
"i", 20147, " = ", 20149
"i", 20231, " = ", 20233
"i", 20357, " = ", 20359
"i", 20441, " = ", 20443
"i", 20477, " = ", 20479
"i", 20507, " = ", 20509
"i", 20549, " = ", 20551
"i", 20639, " = ", 20641
"i", 20717, " = ", 20719
"i", 20747, " = ", 20749
"i", 20771, " = ", 20773
"i", 20807, " = ", 20809
"i", 20897, " = ", 20899
"i", 20981, " = ", 20983
```
```

```
"i", 21011, " = ", 21013
"i", 21017, " = ", 21019
"i", 21059, " = ", 21061
"i", 21191, " = ", 21193
"i", 21317, " = ", 21319
"i", 21377, " = ", 21379
```

Anmerkung:

1. In diesen (nochmals erweiterten) Intervall sind 21 Zwillinge mod 2 situiert.

---

16. Die im Intervall i:= 10^{10} bis 10^{10} + 5000 situierten Zwillinge mod 2:

- ```
  for i from 10^10 to 10^10+5000 do; if (isprime (2*i-1) and isprime (2*i+1))
      then print (("i", 2*i-1, " = ", 2*i+1)) end_if: end_for:
  ```

```
"i", 20000001221, " = ", 20000001223
"i", 20000001311, " = ", 20000001313
"i", 20000001809, " = ", 20000001811
"i", 20000001821, " = ", 20000001823
"i", 20000001839, " = ", 20000001841
"i", 20000002157, " = ", 20000002159
"i", 20000003711, " = ", 20000003713
"i", 20000004287, " = ", 20000004289
"i", 20000005031, " = ", 20000005033
"i", 20000005301, " = ", 20000005303
"i", 20000005391, " = ", 20000005393
"i", 20000005577, " = ", 20000005579
"i", 20000007281, " = ", 20000007283
"i", 20000007299, " = ", 20000007301
"i", 20000007539, " = ", 20000007541
"i", 20000007569, " = ", 20000007571
"i", 20000007689, " = ", 20000007691
"i", 20000008907, " = ", 20000008909
"i", 20000009171, " = ", 20000009173
"i", 20000009177, " = ", 20000009179
"i", 20000009747, " = ", 20000009749
```

Anmerkungen und Schlussfolgerungen:

1. In diesem (wesentlich erweiterten) Intervall sind (ebenfalls) 21 Zwillinge mod 2 situiert.

2. Ein Vergleich der in den Beispielen: **11.** bis **16.** bestimmten Zwillinge mod 2, bestätigt das Asymptote-Gesetz in dem Sinne, dass in zunehmend höheren Zahlenbereichen in Intervallen, die in einem logarithmischen Maßstab vergrößert werden, etwa gleich viele Zwillinge mod 2 situiert sind.

3. Aus den Beispielen: **11.** bis **16.** ist zu entnehmen, dass alle Zwillinge mod 2 nur in den drei Folgen: *Folge 1, Folge 2, Folge 3* auftreten, die in dieser Monographie deduziert wurden.

4. In den Beispielen: **11.** bis **16.** wurde die Bestimmung der Zwillinge mod 2 auf die (fortlaufende)
```

Folge aller geraden Zahlen von $n_g := 2, 4, 6, 8, 10, \ldots, 2*i$ bezogen. In dieser Folge bestehen die geraden Zahlen aus divergierenden, Primzahl - Faktoren. und dies ist der tiefere Grund dafür, ob in gewissen Abständen von diesen geraden Zahlen Primzahlen situiert sein können.

In den folgenden Beispielen für die Methode 3 werden wiederum alle ungeraden Zahlen (n_u) einbezogen, die in der Zahlenfolge: $n_u = 1, 3, 5, 7, 9, 11, 13, 15, \ldots$ auftreten. Für alle diese ungeraden Zahlen gilt offensichtlich die wichtige "Folge-Gleichung":

$$n_u = 2*i - i \text{ für } i := 1 \text{ bis } \infty$$

Diese Folge-Gleichung ermöglicht die Bestimmung von Zwillingen mod 2n, n > 1 in diversen Zahlen-Intervallen mit dem MuPAD - Test, was an den nachfolgenden Beispielen demonstriert wird.

---

17. Die Bestimmung der im Intervall i: = 1 bis 100 situierten Zwillinge mod 4:

- ```
  for i from 1 to 100 do: if (isprime (2*i-i-2) and isprime (2*i-i+2))
    then print (("i", 2*i-i-2, " = ", 2*i-i+2)) end_if: end_for:
  ```

```
"i", 3, " = ", 7
"i", 7, " = ", 11
"i", 13, " = ", 17
"i", 19, " = ", 23
"i", 37, " = ", 41
"i", 43, " = ", 47
"i", 67, " = ", 71
"i", 79, " = ", 83
"i", 97, " = ", 101
```

Anmerkung:

1. In diesem Intervall sind 9 Zwillinge mod 4 situiert.

---

18. Die Bestimmung der im Intervall i:= 1000 bis 1460 situierten Zwillinge mod 4:

- ```
  for i from 1000 to 1460 do: if (isprime (2*i-i-2) and isprime (2*i-i+2))
    then print (("i", 2*i-i-2, " = ", 2*i-i+2)) end_if: end_for:
  ```

```
"i", 1009, " = ", 1013
"i", 1087, " = ", 1091
"i", 1093, " = ", 1097
"i", 1213, " = ", 1217
"i", 1279, " = ", 1283
"i", 1297, " = ", 1301
"i", 1303, " = ", 1307
"i", 1423, " = ", 1427
"i", 1429, " = ", 1433
"i", 1447, " = ", 1451
```

Anmerkung:

1. In diesem (logarithmisch erweiterten) Intervall sind 10 Zwillinge mod 4 situiert..

---

19. Die Bestimmung der im Intervall i:= 10^10 bis 10^10 +3500 situierten Zwillinge mod 4:

- for i from 10^10 to 10^10+3500 do: if (isprime (2*i-i-2) and isprime (2*i-i+2)) then print (("i", 2*i-i-2, " = ", 2*i-i+2)) end_if: end_for:

```
"i", 10000000597, " = ", 10000000601
"i", 10000001047, " = ", 10000001051
"i", 10000001113, " = ", 10000001117
"i", 10000001437, " = ", 10000001441
"i", 10000001467, " = ", 10000001471
"i", 10000001743, " = ", 10000001747
"i", 10000002403, " = ", 10000002407
"i", 10000002703, " = ", 10000002707
"i", 10000002937, " = ", 10000002941
"i", 10000003099, " = ", 10000003103
```

<u>Anmerkung:</u>
1. In diesem erweiterten Intervall sind ebenfall 10 Zwillinge mod 4 situiert.

---

20. Die Bestimmung der im Intervall i:= 10^20 bis 10^20+3500 situierten Zwillinge mod 4:

- for i from 10^10 to 10^10+3500 do: if (isprime (2*i-i) and isprime (2*i-i+4)) then print (("i", 2*i-i, " = ", 2*i-i+4)) end_if: end_for:

```
"i", 10000000597, " = ", 10000000601
"i", 10000001047, " = ", 10000001051
"i", 10000001113, " = ", 10000001117
"i", 10000001437, " = ", 10000001441
"i", 10000001467, " = ", 10000001471
"i", 10000001743, " = ", 10000001747
"i", 10000002403, " = ", 10000002407
"i", 10000002703, " = ", 10000002707
"i", 10000002937, " = ", 10000002941
"i", 10000003099, " = ", 10000003103
```

<u>Anmerkung:</u>
1. In dem erweiterten Intervall sind ebenfalls 10 Zwillinge mod 4 situiert.

---

21. Die Bestimmung der im Intervall i:= 1 bis 100 situierten Zwillinge mod 6:

- for i from 1 to 100 do: if (isprime (2*i-i) and isprime (2*i-i+6)) then print (("i", 2*i-i, " = ", 2*i-i+6)) end_if: end_for:

```
"i", 5, " = ", 11
"i", 7, " = ", 13
"i", 11, " = ", 17
"i", 13, " = ", 19
"i", 17, " = ", 23
"i", 23, " = ", 29
```

```
"i", 31, " = ", 37
"i", 37, " = ", 43
"i", 41, " = ", 47
"i", 47, " = ", 53
"i", 53, " = ", 59
"i", 61, " = ", 67
"i", 67, " = ", 73
"i", 73, " = ", 79
"i", 83, " = ", 89
"i", 97, " = ", 103
```

Anmerkung:

1. In diesem Intervall sind 16 Zwillinge mod 6 situiert.

---

22. Die Bestimmung der im Intervall i::= 10^20 bis 10^20 +14800 situierten Zwillinge mod 6:

- ```
  for i from 10^20 to 10^20+14800 do: if (isprime (2*i-i) and isprime (2*i-i+6))
    then print (("i", 2*i-i, " = ", 2*i-i+6)) end_if: end_for:
  ```

```
"i", 100000000000000000757, " = ", 100000000000000000763
"i", 100000000000000001261, " = ", 100000000000000001267
"i", 100000000000000001821, " = ", 100000000000000001827
"i", 100000000000000001977, " = ", 100000000000000001983
"i", 100000000000000002691, " = ", 100000000000000002697
"i", 100000000000000003097, " = ", 100000000000000003103
"i", 100000000000000003307, " = ", 100000000000000003313
"i", 100000000000000004533, " = ", 100000000000000004539
"i", 100000000000000012271, " = ", 100000000000000012277
"i", 100000000000000012391, " = ", 100000000000000012397
"i", 100000000000000012651, " = ", 100000000000000012657
"i", 100000000000000012937, " = ", 100000000000000012943
"i", 100000000000000013363, " = ", 100000000000000013369
"i", 100000000000000013497, " = ", 100000000000000013503
"i", 100000000000000013717, " = ", 100000000000000013723
"i", 100000000000000014611, " = ", 100000000000000014617
```

Anmerkung:

1. In diesem erweiterten Intervall sind ebenfalls 16 Zwillinge mod 6 situiert.

2. Aus dieser Aufstellung kann entnommen werden, in welchen Abständen die Zwillinge mod 6 auftreten.

---

23. Die Bestimmung der im Intervall i:= 1 bis 100 situierten Zwillinge mod 8:

- ```
  for i from 1 to 100 do: if (isprime (2*i-i) and isprime (2*i-i+8))
    then print (("i", 2*i-i, " = ", 2*i-i+8)) end_if: end_for:
  ```

```
"i", 3, " = ", 11
"i", 5, " = ", 13
```

```
"i", 11, " = ", 19
"i", 23, " = ", 31
"i", 29, " = ", 37
"i", 53, " = ", 61
"i", 59, " = ", 67
"i", 71, " = ", 79
"i", 89, " = ", 97
```

Anmerkung:

1. In diesem Intervall sind 9 Zwillinge mod 8 situiert.

---

24. Die Bestimmung der im Intervall i:= 10^20 bis 10^20 + 20400 situierten Zwillinge mod 8:

- `for i from 10^20 to 10^20+20400 do: if (isprime (2*i-i) and isprime (2*i-i+8)) then print (("i", 2*i-i, " = ", 2*i-i+8)) end_if: end_for:`

```
"i", 100000000000000002659, " = ", 100000000000000002667
"i", 100000000000000003121, " = ", 100000000000000003129
"i", 100000000000000009073, " = ", 100000000000000009081
"i", 100000000000000009823, " = ", 100000000000000009831
"i", 100000000000000011269, " = ", 100000000000000011277
"i", 100000000000000012391, " = ", 100000000000000012399
"i", 100000000000000013951, " = ", 100000000000000013959
"i", 100000000000000015379, " = ", 100000000000000015387
"i", 100000000000000020359, " = ", 100000000000000020367
```

Anmerkung:

1. In diesem erweiterten Intervall sind (ebenfalls) 9 Zwillinge mod 8 situiert.

---

25. Die Bestimmung der im Intervall i:= 1 bis 100 situierten Zwillinge mod 10:

- `for i from 1 to 100 do: if (isprime (2*i-i) and isprime (2*i-i+10)) then print (("i", 2*i-i, " = ", 2*i-i+10)) end_if: end_for:`

```
"i", 3, " = ", 13 ... (3,5) —(5,7) —(7,11) ... (11,13).
"i", 7, " = ", 17 ... (7,11) ... (11,13) ... (13,17)
"i", 13, " = ", 23 ... (13,17) ... (17,19) ...(19,23)
"i", 19, " = ", 29 ... (19,23) ... (23,29)
"i", 31, " = ", 41 ... (31,37) ... (37, 41)
"i", 37, " = ", 47 ... (37, 41) ... (41,43) ... (43,47)
"i", 43, " = ", 53 ... (43, 47) ...)47,53)
"i", 61, " = ", 71 ... (61, 67) ... (67,71)
"i", 73, " = ", 83 ... (73, 79) ... (79,83)
"i", 79, " = ", 89 ... (79, 83) ... (83, 89)
"i", 97, " = ", 107 ... (97, 101) .. (101,103) ... (103, 107)
```

Anmerkung:

1. In diesem Intervall sind 11 Zwillinge mod 10 situiert.

2. In der Tabelle wurden die Zwillinge angegeben, aus denen die (jeweiligen) Zwillinge mod 10 "zusammengesetzt" sind.

---

26. Die Bestimmung der im Intervall i:= 10^20 bis 10^20+21500 situierten Zwillinge mod 10:

-
```
for i from 10^20 to 10^20+21600 do: if (isprime (2*i-i) and isprime (2*i-i+10))
    then print (("i", 2*i-i, " = ", 2*i-i+10)) end_if: end_for:
```

```
"i", 100000000000000000753, " = ", 100000000000000000763
"i", 100000000000000003087, " = ", 100000000000000003097
"i", 100000000000000004239, " = ", 100000000000000004249
"i", 100000000000000005601, " = ", 100000000000000005611
"i", 100000000000000007659, " = ", 100000000000000007669
"i", 100000000000000007827, " = ", 100000000000000007837
"i", 100000000000000011277, " = ", 100000000000000011287
"i", 100000000000000012657, " = ", 100000000000000012667
"i", 100000000000000017079, " = ", 100000000000000017089
"i", 100000000000000020517, " = ", 100000000000000020527
"i", 100000000000000021003, " = ", 100000000000000021013
```

Anmerkung:

1. Im erweiterten Intervall sind ebenfalls 11 Zwillinge mod 10 situiert.

---

27. Der Nachweis, dass mit Hilfe der "Und-Kondition":

" for i from i_1 to $i_2 > i_i$ do: if (isprime (2*i - i) and isprime (2*i -i + n))
then print (("i", 2*i -i, " = ", 2*i -i +n)) end_if: end_for: ... **Tab. 2**

alle Zwillinge mod 2n, n ≥1 mit dem MuPAD - Testprogramm bestimmt werden, die in beliebigen vorgegebenen Intervallen im Zahlenbereich (n → ∞) situiert sind.

Hier die Bestimmung von Zwillingen mod 2, die im Intervall i:= 1 bis 100 und nachfolgend im Intervall i: = 10^10 bis 10^!0 +10000 situiert sind:

-
```
for i from 1 to 100 do: if (isprime (2*i-i) and isprime (2*i-i+2))
    then print (("i", 2*i-i, " = ", 2*i-i+2)) end_if: end_for:
```

```
"i", 3, " = ", 5
"i", 5, " = ", 7
"i", 11, " = ", 13
"i", 17, " = ", 19
"i", 29, " = ", 31
"i", 41, " = ", 43
"i", 59, " = ", 61
"i", 71, " = ", 73:
```

-
```
for i from 10^10 to 10^10+10000 do: if (isprime (2*i-i) and isprime (2*i-i+2))
    then print (("i", 2*i-i, " = ", 2*i-i+2)) end_if: end_for:
```

```
"i", 10000000277, " = ", 10000000279
"i", 10000000991, " = ", 10000000993
```

```
"i", 10000003001, " = ", 10000003003
"i", 10000003019, " = ", 10000003021
"i", 10000003229, " = ", 10000003231
"i", 10000003337, " = ", 10000003339
"i", 10000005161, " = ", 10000005163
"i", 10000005551, " = ", 10000005553
"i", 10000005809, " = ", 10000005811
"i", 10000006349, " = ", 10000006351
"i", 10000007441, " = ", 10000007443
"i", 10000008911, " = ", 10000008913
"i", 10000009049, " = ", 10000009051
"i", 10000009247, " = ", 10000009249
```

28. Mit dieser "universellen" Und-Kondition können auch alle Primzahlen bestimmt werden, die in einem vorgegebenen Intervall im Zahlenbereich (i → ∞) situiert sind - hier ein Beispiel:

- for i from 10^10 to 10^10+500 do: if (isprime (2*i-i) and isprime (2*i-i)) then print (("i", 2*i-i, " = ", 2*i-i)) end_if: end_for:

```
                                                   Zwilling mod (   ):
"i", 10000000019, " = ", 10000000019
                                                   (14)
"i", 10000000033, " = ", 10000000033
                                                   (28)
"i", 10000000061, " = ", 10000000061
                                                   (8)
"i", 10000000069, " = ", 10000000069
                                                   (28)
"i", 10000000097, " = ", 10000000097
                                                   (6)
"i", 10000000103, " = ", 10000000103
                                                   (18)
"i", 10000000121, " = ", 10000000121
                                                   (20)
"i", 10000000141, " = ", 10000000141
                                                   (6)
"i", 10000000147, " = ", 10000000147
                                                   (60)
"i", 10000000207, " = ", 10000000207
                                                   (52)
"i", 10000000259, " = ", 10000000259
                                                   (18)
"i", 10000000277, " = ", 10000000277
                                                   (2)
"i", 10000000279, " = ", 10000000279
                                                   (40)
"i", 10000000319, " = ", 10000000319
                                                   (24)
"i", 10000000343, " = ", 10000000343
                                                   (48)
"i", 10000000391, " = ", 10000000391
                                                   (12)
"i", 10000000403, " = ", 10000000403
                                                   (66)
"i", 10000000469, " = ", 10000000469
```

<u>Anmerkung:</u>

1. Im vorgegebenen Intervall sind insgesamt 18 Primzahlen situiert, die paarweise Zwillinge mod 2n bilden.

2. Diese Zwillinge wurden hier (in Klammern vermerkt) angegeben.

Schluss-Bemerkungen:

1. Mit Hilfe schneller PC-Rechner und bei Einsatz des in Tab. 2 angeführten Programms. können in beliebigen Zahlen-Intervallen die dort situierten Primzahlen ermittelt werden.

2. Mit Hilfe des in Tab. 2 angegebenen Programms können die Zwillinge beliebiger Modularität (mod 2n, n ≥1), die in beliebigen Intervallen im Zahlenbereich ($n \to \infty$) auftreten, bestimmt werden.

3. Die Ordnungs-Folgen, in denen die Zwillinge mod 2n, ≥1 (in definierten Intervallen) auftreten, können die Basis für eine extrem sichere (noch unbekannte) Daten-Verschlüsselung bilden.

Anlage 10

Die Bestimmung der Zwillinge mod 2 und mod 6 in den Potenz-Folgen: 6^n, 14^n und 18^n

Vorworte:

In den Anlagen: **Anlage 4**, **Anlage 5**, **Anlage 6**, **Anlage 7** und **Anlage 8** wurden die Primzahlen vom Typ: CM_p, CF_p bzw. SM_p und SF_p in den jeweiligen Klassen-Folgen für unterschiedliche Intervalle, die in ansteigenden Zahlenbereichen situiert waren, bestimmt. Dabei wurden die Potenzen der geraden Zahl $n_g = 10^n = (2 \cdot 5)^n$ zugrunde gelegt.

In **Anlage 9** soll der Nachweis erbracht werden, dass Zwillinge mod 2 (und allgemein mod 2n, n > 1) auch in den Umgebungen der Potenzen der geraden Zahlen $n_g = 2 \cdot 3 = 6$, $2 \cdot 7 = 14$, $2 \cdot 3^2 = 18$ usw. auftreten. In **Anlage 9** wird dieser Sachverhalt an einigen Beispielen demonstriert.

Hieraus ergeben sich zwei Schlussfolgerungen:

Schlussfolgerung 1:

Die von Euklid zum Beweis des Hauptsatzes der Primzahl - Theorie stammende Gleichung 1 ist um drei weitere Gleichungen: Gleichung 2, Gleichung 3 und Gleichung 4 zu erweitern, die bereits im Vorwort dieser Monographie eingeführt wurden - siehe die nachfolgende Tabelle:

Tab.1

Gleichung 1:	$n_g + 1, 3, 5, \ldots = \Pi p_i + 1, 3, 5, .$	Gleichung 2:	$n_g - 1, 3, 5, \ldots = \Pi p_i - 1, 3, 5, \ldots$
Gleichung 3:	$n_u + 2, 4, 6, \ldots = \Pi p_i + 2. 4, 6, \ldots$	Gleichung 3	$n_u - 2, 4, 6, \ldots = \Pi p_i - 2, 4, 6, \ldots$

Die aus diesen Gleichungen resultierenden ungeraden Zahlen können fallweise Primzahlen sein, die in den Fakultäten: Πp_i nicht auftreten. Des weiteren können die Gleichungen 1 und 2 bzw. 2 und 3 gleichzeitig erfüllt sein und damit Primzahl-Zwillinge mod 2 bzw. mod 4 ergeben.

Schlussfolgerung 2:

Nachdem die geraden Zahlen $n_g = 2^n$ in der Folge der natürlichen Zahlen ($n \to \infty$) spärlich und in immer größeren Abständen auftreten, die geraden Zahlen $n_g \neq 2^n$ hingegen in einer weit größeren Zahl auftreten als die geraden Zahlen $n_g = 2^n$, ist es einsichtig, dass die Gleichungen 1 und 2 in einem viel höheren Maße von diesen (composed equal numbers) erfüllt werden.
Daraus ist es auch leicht einsichtig, warum bisher nur 42 Mersennesche Primzahlen (M_p) bestimmt werden konnten und nur 5 Fermatsche Primzahlen (F_p) bekannt sind.

---

1. Die Bestimmung der Zwillinge mod 2, die im Intervall i:= 6^10 bis 6^10+1000 situiert sind:

```
• for i from 6^10 to 6^10+1000 do: if (isprime (i-1) and isprime (i+1))
    then print (("i", i-1, " = ", i+1)) end_if: end_for:

                    "i", 60466667, " = ", 60466669
```

"i", 60466697, " = ", 60466699

"i", 60466829, " = ", 60466831

"i", 60466961, " = ", 60466963

"i", 60466979, " = ", 60466981

Anmerkung:

1. Im Intervall i:= 6^10 bis 6^10+1000 sind 5 Zwillinge mod 2 aituiert.

---

2. Die Bestimmung der Zwillinge mod 2, die im Intervall i:= 6^100 bis 6^1000+500000 situiert sind:

- ```
  for i from 6^100 to 6^100+50000 do: if (isprime (i-1) and isprime (i+1))
      then print (("i", i-1, " = ", i+1)) end_if: end_for:
  ```

"i",

653318623500070906096690267158057820537143710472954871543071966369497141496167

, " = ",

653318623500070906096690267158057820537143710472954871543071966369497141496169

"i",

653318623500070906096690267158057820537143710472954871543071966369497141498489

, " = ",

653318623500070906096690267158057820537143710472954871543071966369497141498491

"i",

653318623500070906096690267158057820537143710472954871543071966369497141503277

, " = ",

653318623500070906096690267158057820537143710472954871543071966369497141503279

"i",

653318623500070906096690267158057820537143710472954871543071966369497141514917

, " = ",

653318623500070906096690267158057820537143710472954871543071966369497141514919

Anmerkung:

1. In diesem Intervall sind 4 Zwillinge mod 2 situiert.

---

3. Die Bestimmung der Zwillinge mod 2, die im Intervall i:= 18^10 bis 18^10+1000 aituiert sind:

- ```
  for i from 18^10 to 18^10+1000 do: if (isprime (i-1) and isprime (i+1))
      then print (("i", i-1, " = ", i+1)) end_if: end_for:
  ```

"i", 3570467226749, " = ", 3570467226751

"i", 3570467226839, " = ", 3570467226841

"i", 3570467227319, " = ", 3570467227321

"i", 3570467227331, " = ", 3570467227333

"i", 3570467227529, " = ", 3570467227531

Anmerkung:

1. In diesem Intervall sind 5 Zwillinge mod 2 situiert.

---

4. Die Bestimmung der Zwillinge mod 2, die im Intervall i:= 18^100 +50000 bis 18^100 +100000 situiert sind:

- ```
  for i from 18^100+50000 to 18^100+100000 do: if (isprime (i-1) and isprime (i+1))
      then print (("i", i-1, " = ", i+1)) end_if: end_for:
  ```

```
"i",

336705732427516898587460627772004697542605295316077247452351004740460372771448 5\
6303114350735475700982267461869740537956383347 1

, " = ",

3367057324275168985874606277720046975426052953160772474523510047404603727714485\
63031143507354757009822674618697405379563833473

"i",

3367057324275168985874606277720046975426052953160772474523510047404603727714485\
63031143507354757009822674618697405379563836939

, " = ",

3367057324275168985874606277720046975426052953160772474523510047404603727714485\
63031143507354757009822674618697405379563836941

"i",

3367057324275168985874606277720046975426052953160772474523510047404603727714485\
63031143507354757009822674618697405379563839291

, " = ",

3367057324275168985874606277720046975426052953160772474523510047404603727714485\
63031143507354757009822674618697405379563839293

"i",

3367057324275168985874606277720046975426052953160772474523510047404603727714485\
63031143507354757009822674618697405379563839417

, " = ",

3367057324275168985874606277720046975426052953160772474523510047404603727714485\
63031143507354757009822674618697405379563839419
```

Anmerkung:

1. In diesem Intervall sind 4 Zwillinge mod 2 situiert.

2. Im "vorangehenden und gleich großen" Intervall i:= 18^100 bis 18^100+50000 ist dagegen kein Zwilling mod 2 situiert.

---

5. Die Bestimmung der im Intervall i:= 14^10+1000 bis 14^10 +5000 situierten Zwillinge mod 2:

- ```
  for i from 14^10+1000 to 14^10+5000 do: if (isprime (i-1) and isprime (i+1))
      then print (("i", i-1, " = ", i+1)) end_if: end_for:
  ```

```
"i", 289254657071, " = ", 289254657073

"i", 289254657869, " = ", 289254657871

"i", 289254659789, " = ", 289254659791
```

Anmerkung:

1. In diesem Intervall sind 3 Zwillinge mod 2 situiert.

---

6. Die Bestimmung der im Intervall i:= 14^10 bis 14^10+1000 situierten Zwillinge mod 4:

- ```
  for i from 18^10 to 18^10+1000 do: if (isprime (i-2) and isprime (i+2))
      then print (("i", i-2, " = ", i+2)) end_if: end_for:
  ```
  ```

"i", 3570467226919, " = ", 3570467226923

"i", 3570467227339, " = ", 3570467227343

Anmerkung:

1. In diesem Intervall sind 2 Zwillinge mod 4 lokalisiert.

---

7. Die Bestimmung der im Intervall i:= 14^100 + 20000 bis 14^100+30000 lokalisierten Zwillinge mod 6:

- for i from 14^100+20000 to 14^100+30000 do: if (isprime (i-3) and isprime (i+3)) then print (("i", i-3, " = ", i+3)) end_if: end_for:

"i",

41001860888499328805296416524670972545801067523792027322197126356748926146602644\
83061479032219018658198141395378784l

, " = ",

41001860888499328805296416524670972545801067523792027322197126356748926146602644\
830614790322190186581981413953787847

"i",

41001860888499328805296416524670972545801067523792027322197126356748926146602644\
830614790322190186581981413953789727

, " = ",

41001860888499328805296416524670972545801067523792027322197126356748926146602644\
830614790322190186581981413953789733

Anmerkung:

1. In diesem Intervall sind 2 Zwillinge mod 6 lokalisiert.

---

8. Die Bestimmung der im Intervall i:= 18^10 bis 18^10+1000 situierten Zwillinge mod 2:

- for i from 18^10 to 18^10+1000 do: if (isprime (i-1) and isprime (i+1)) then print (("i", i-1, " = ", i+1)) end_if: end_for:

"i", 3570467226749, " = ", 3570467226751 ... *Folge 3*

"i", 3570467226839, " = ", 3570467226841 ... *Folge 3*

"i", 3570467227319, " = ", 3570467227321 ... *Folge 3*

"i", 3570467227331, " = ", 3570467227333 ... *Folge 1*

"i", 3570467227529, " = ", 3570467227531 ... *Folge 3*

Anmerkung:

1. In diesem Intervall sind 5 Zwillinge mod 2 lokalisiert.

---

9. Die Bestimmung der im Intervall i:= 18^100'500000 bis 18^100+100000 situierten Zwillinge mod 2:

- for i from 18^100+50000 to 18^100+100000 do: if (isprime (i-1) and isprime (i+1)) then print (("i", i-1, " = ", i+1)) end_if: end_for:

"i",

33670573242751689858746062777200469754260529531607724745235100474040460372771448 5\
630311435073547570098226746186974053795638334 71

, " = ",

```
3367057324275168985874606277720046975426052953160772474523510047404603727714485\
630311435073547570098226746186974053795638 33473

"i",

3367057324275168985874606277720046975426052953160772474523510047404603727714485\
630311435073547570098226746186974053795638 36939

    , " = ",

3367057324275168985874606277720046975426052953160772474523510047404603727714485\
630311435073547570098226746186974053795638 36941

"i",

3367057324275168985874606277720046975426052953160772474523510047404603727714485\
630311435073547570098226746186974053795638 39291

    , " = ",

3367057324275168985874606277720046975426052953160772474523510047404603727714485\
630311435073547570098226746186974053795638 39293

"i",

3367057324275168985874606277720046975426052953160772474523510047404603727714485\
630311435073547570098226746186974053795638 39417

    , " = ",

3367057324275168985874606277720046975426052953160772474523510047404603727714485\
630311435073547570098226746186974053795638 39419
```

Anmerkung:

1. In diesem Intervall sind 4 Zwillinge mod 2 situiert.

---

10. Die Bestimmung der im Intervall i:= 18^10 bis 18^10+1000 situierten Zwillinge mod 4:

- `for i from 18^10 to 18^10+1000 do: if (isprime (i-2)and isprime (i+2)) then print (("i", i-2, " = ", i+2)) end_if: end_for:`

```
"i", 3570467226919, " = ", 3570467226923

"i", 3570467227339, " = ", 3570467227343
```

Anmerkung:

1. In diesem Intervall sind 2 Zwilinge mod 4 situiert.

---

11. Die Bestimmung der im Intervall i:= 18^10 bis 18^10+1000 situierten Zwillinge mod 6:

- `for i from 18^10 to 18^10+1000 do: if (isprime (i-3) and isprime (i+3)) then print (("i", i-3, " = ", i+3)) end_if: end_for:`

```
"i", 3570467226841, " = ", 3570467226847

"i", 3570467226961, " = ", 3570467226967

"i", 3570467226967, " = ", 3570467226973

"i", 3570467227333, " = ", 3570467227339
```

Anmerkung:

1 In diesem Intervall sind 4 Zwillinge mod 6 situiert.

Zusammenfassung, die Schlussfolgerungen und die verborgenen Ursachen für die Genese von Primzahlen:

1. Die Folge der in den getesteten Intervallen situierten Zahlen durchläuft alle geraden Zahlen (n_g) und

alle ungeraden Zahlen (n_u).

2. Die hier gewonnen Resultate geben den Anlass, die Gesetze: Gesetz 1 , Gesetz 2 , Gesetz 3 und Gesetz 4 , die den Hauptsatz der Primzahl-Theorie prägen, wie bereits in der Tafel 1 formuliert wurde zu erweitern.

3. Am Beispiel der Zwillinge mod 6, die im Intervall i:= 18^10 bis 18^10+1000 im **11.** Rechenbeispiel ermittelt wurden, soll nun (im **12.** Rechenbeispiel der Nachweis erbracht werden, das hierfür die zwei Gesetze: Gesetz 1 und Gesetz 2 der Faktor-Zerlegung der nicht primen Zahlen ursächlich sind.

12- Die Faktor-Zerlegung der geraden Zahlen, die inmitten der Zwillinge mod 6 situiert sind:

- `ifactor(3570467226844)...` siehe die Faktor-Zerlegung dieser geraden Zahl (n_g) in der p=3 nicht auftritt.

  $2^2 \cdot 7 \cdot 17 \cdot 127 \cdot 59062847$

- `ifactor(3570467226844-3)...` bei der Subtraktion von p=3 erfolgt die "Verschmelzung" zu der folgenden Primzahl

  3570467226841

- `ifactor(3570467226844+3)...` analog erfolgt bei der Addition von p=3 die Verschmelzung zu der folgenden Primzahl.

  3570467226847

- `ifactor(3570467226964) ...` in der Faktor-Zerlegung dieser geraden Zahl ist p=3 nicht enthalten.

  $2^2 \cdot 11 \cdot 2381 \cdot 34081051$

- `ifactor(3570467226964-3) ...` analoge Gründe für die "Verschmelzung" zu der nachfolgenden Primzahl - usw,

  3570467226961

- `ifactor(3570467226964+3)`

  3570467226967

- `ifactor(3570467226970)`

  $2 \cdot 5 \cdot 7 \cdot 166603 \cdot 306157$

- `ifactor(3570467226970-3)`

  3570467226967

`ifactor(3570467226970+3)`

3570467226973

`ifactor(3570467227336)`

$2^3 \cdot 97 \cdot 4601117561$

`ifactor(3570467227336-3)`

3570467227333

- `ifactor(3570467227336+3)`

  3570467227339

Anmerkungen und Schlussfolgerungen:

1- Die hier dokumentierten Gründe für die "Verschmelzung" von geraden (aber auch ungeraden) Zahlen zu Primzahlen infolge der Addition bzw. Subtraktion von Primzahlen, die in den Faktor-Zerlegungen dieser Zahlen (n_g) bzw. (n_u) nicht auftreten, gelten (fallweise) generell - q.e.d.

2. Nachdem in der Faktor-Zerlegung der Zahl $n_g = 3570467226844 = 2^2 \cdot 7 \cdot 17 \cdot 127 \cdot 59062847$ die Primzahlen $p = 3$ und auch $p = 5$ nicht auftreten, wird nachfolgend getestet, ob bei der Addition bzw. der Subtraktion der Primzahl $p = 5$ diese gerade Zahl zu einer Primzahl "mutiert":

`ifactor(3570467226844+5)` Anmerkung: In der Faktor-Zerlegung von $n_g + 5$ tritt die Zahl $p = 5$ nicht auf.

$396718580761 \cdot 3^2$

`ifactor(3570467226844-5)`. Anmerkung: Die Zahl $n_g - 5$ ist eine Primzahl und bildet mit $n_g - 3$ einen Zwilling mod 2

3570467226839

3. Die Primzahlen: (3570467226839, 3570467226841), (3570467226841, 3570467226847) bilden einen Zwilling mod 8, der aus einer "Verkettung" eines Zwillings mod 2 mit einem Zwilling mod 6 besteht.

4. Diese Beispiele erklären die Genese von Primzahlen durch die Addition bzw. Subtraktion von Primzahlen (p) zu nicht primen Zahlen, in deren Faktor-Zerlegungen diese Primzahlen (p) nicht auftreten.

5. Diese Beispiele bestätigen die Erweiterung der Gleichungen, die zum Beweis des Hauptsatzes der Primzahl-Theorie gereichen.

Anlage 12

Die Bestimmung von Primzahlen mit definierten Endziffern: EZ1, EZ3, EZ7, EZ9 und von Zwillingen mod 2n, n>1 in den Umgebungen der geraden Zahlen $n_g = 2^n$ mit dem MuPD - Testprogramm:

In **Tab. 28** (Seite 54/55) wurden die bisher entdeckten Mersenne - Primzahlen: M_{p1} bis M_{p42} in der chronologischen Zeitfolge ihrer Entdeckung aufgelistet. Diese Primzahlen treten (bis auf die Primzahl $M_{p1} = 2^2 - 1 = 3$) allesamt nur in den Endziffern: EZ1 und EZ7 auf.
In **Kapitel 13** (siehe die Tabellen: Tab. 41 bis Tab. 46) wurden zahlreiche Primzahlen vom Typ: $2^n \pm n_u > 1$ ermittelt, die in allen vier Endziffern auftreten, wie auch zahlreiche Zwillinge mod 2n, n > 1 bestimmt, die in den Umgebungen der Zahlen $n_g = 2^n$ situiert sind.

In Anlage 12 wird der Einsatz des MuPAD - Testprogramms für die Lösung dieser Aufgabe in den vier Folgen:

Folge 1 := 2^{1+4n}, EZ2 *Folge 2* := 2^{2+4n}, EZ4 *Folge 3* := 2^{3+4n}, EZ8 *Folge 4* := 2^{4+4n}, EZ6

demonstriert.

1. Die Bestimmung von Primzahlen, die im Umfeld der Zahlen $n_g = 2^{1+4n}$ situiert sind:
<u>Anmerkung:</u>
Die ungeraden Zahlen n_g -7, -17, ... und n_g = +3, +13, ... mit der Endziffer, EZ5 scheiden aus.

1.1 Die Bestimmung der Primzahlen $M_p = 2^{(1+4 \cdot i)} - 1$ im Intervall i := 0 bis 1000

```
• for i from 0 to 1000 do: if (isprime (2^(1+4*i)-1) and isprime (2^(1+4*i) -1))
    then print (("i", i, " = ", 2^(1+4*i)-1)) end_if: end_for:

                    "i", 1, " = ", 31                                  2^5-1

                    "i", 3, " = ", 8191                                2^13-1

                   "i", 4, " = ", 131071

              "i", 15, " = ", 2305843009213693951                      2^61-1

           "i", 22, " = ", 618970019642690137449562111

"i", 130, " = ",

686479766013060971498190079908139321726943530014330540939446345918554318339765 6\
0521225596406614545549772963113914808580371219879997166438125740282911150571 51

"i", 570, " = ",                                                       2^2281-1

446087557183758429571151706402101809886208632412859901111991219963404685792820 4\
733691125452690039890261532459311243167023957587056936793647909034974611470710 6\
525419335393812497822630794731241079887486904007027932842881031175484410809487 8\
252494866760969586998128982645877596028979171536962503068429617331702184750324 5\
830091718321049160501576288866063721455017022259251252240768296054271735739648 1\
299525056941248072073847685529368166671284483119087762060678666386219024011857 0\
736831901886479225810414714078935386562497968178729127629594924411960961386713 9\
462798992750069549171397587960612238033935373810346664944029510520590479686932 5\
53886479304409251041868170096401717641331724181328363 51

"i", 804, " = ",                                                       2^3217-1

259117086013202627776246767922441530941818887553125427303974923161874019266586 3\
620862012095168004834065506952417331941774416895092388070174103777095975120423 1\
```

```
30666240829163535179523111861548622656045476911275958487756105687579311910177 11\
40882625215384903583040118507211642474746182303147139834022928807454567790794 10\
37288235820705892351068433882986888616658650280927692080339605869308790500409 50\
37098759021190183719916209940025689351131365488297391126567973032419865172501 16\
41270350970542777347797234982167644344666838311932254009964899405179024162405 65\
19054483690809616061625743042361721863339415852426431208737266591962061753535 74\
88928945996291951830826218608534009379328394202618665861425032514507730962742 35\
37682293864940712770084607712421182308080413929808705750471382526457144837937 11\
25032081826126566649084251699453951887789613650248405739378594599444335231188 28\
01236604062624686092121503499375847822922371443396288584859382157388212323936 87\
046160677362909315071
```

1.2 Die Bestimmung der Primzahlen $SM_p = 2^{(1+4*i)} - 3$ im Intervall i := 0 bis 1000:

- ```
  for i from 0 to 1000 do: if (isprime (2^(1+4*i)-3) and isprime (2^(1+4*i) -3))
      then print (("i", i, " = ", 2^(1+4*i)-3)) end_if: end_for:
  ```

```
                              "i", 1, " = ", 29                              2^5-3

                              "i", 2, " = ", 509                             2^9-3

                           "i", 7, " = ", 536870909                          2^29-3

"i", 53, " = ",

   131640364585696483372397534604588040398618869250686389067888721 89        2^213-3

"i", 55, " = ",

   336999333339382997433337688587745383420464305281757156013795128 1149     2^221-3

"i", 58, " = ",

   138034926935811275748695117245540509049022179443407731103250484 47598589

"i", 136, " = ",                                                             2^545-3

11517219314030582739994978579676113558706424622852906580737934265886304206519 00\
89480167441564259605943037975312218134915154131611020654072038617988630148194 69\
1448829

"i", 172, " = ",                                                             2^689-3

25684257331779167515825145912500628249757518568472528610831449722015293722153 71\
49517268873841679123595067527776180577036768526965341006048004287830598236369 62\
6719905816818583896077331927945445179262365807410 9

"i", 580, " = ",                                                             2^2321-3

49047845612973371324733846631721500137566094275740642434708385873458315322708 98\
59768211793529265063772382996469356546239266905213455780317615868006371237299 42\
30923884497983486947949324084561140418777078075864993938397653384127026723445 22\
48564917328543790730434324317537173098587264256420556949944444326846865088368 48\
66408762953455364422720642998497090762274602130605776942621351666348619420132 94\
30705553917317075532293803679588575318954087355103897891953777656158853458533 50\
01010064416048869873641735505055187143245993814888711321900037173170287625847 17\
33293892671572219274665383395945290201423866078540130108346276304507495548526 78\
752000476951087518271174956306970994042229793910010099598334571314 9

"i", 809, " = ",                                                             2^3237-3

27170395758337995862310573092104205074884868183490604806069280902938521162648 00\
61210900519470288583688507301813795625817805097018839591307088104217218920787 28\
04661484143680983064403626543335208550185541997718099447257346277459164565525 87\
75822139617847440659489875303817955219599452854704815298600426597205480875799 71\
81115549163932501777913934127286859718101460876974035650834182564016334307757 39\
57620868339403518084294859754072377237051922698256925197932138890442692543120 58\
06756699539287863300243833468661419836353374649772675180752954358685000440118 66\
88524074290574383971435275136387484864572991316873897531132888053933210865275 50\
14311158477007829682960433003662157418938650279405070174309614893932458501988 52\
63448748971724072833604237616659753659877728116863093441006280404062327105584 74\
56801640232912490750630168310006627054690881922922872296574649210706943259378 48\
20189473341570742843732397653361529006768648558630706699157511505065502125784 42\
788114978426489997960937469
```

<u>Anmerkung</u>: In diesem Potenz - Intervall sind 10 Primzahlen SM_p, EZ9 situiert.
```

1.3 Die Bestimmung der Zwillinge mod 2 (SM_p, M_p = 2^(1+4*i -3), 2^(1+4*i -1) im gleichen Intervall:

```
• for i from 0 to 1000 do: if (isprime (2^(1+4*i)-3) and isprime (2^(1+4*i)-1))
     then print (("i", i,2^(1+4*i)-3, " = ", 2^(1+4*i)-1)) end_if: end_for:

                    "i", 1, 29, " = ", 31
```

Anmerkung: In diesem Intervall ist nur ein Zwilling mod 2 situiert.

1.4 Die Bestimmung der Primzahlen SF_p = 2^(1+4*i)+5 im Intervall i:= 0 bis 1000:

```
• for i from 0 to 1000 do: if (isprime (2^(1+4*i)+5) and isprime (2^(1+4*i)+5))
     then print (("i", i, " = ", 2^(1+4*i)+5)) end_if: end_for:

                    "i", 0, " = ", 7

                    "i", 1, " = ", 37

               "i", 13, " = ", 9007199254740997

        "i", 35, " = ", 2787593149816327892691964784081045188247557

"i", 68, " = ",

1517710072051350836655829614705874145814380343009484000977978445108518972816569\
1397

"i", 85, " = ",

4479489484355608421114884561136888556243290994469299069799978201927583742360321\
8907617549865432142315557
```

Anmerkung: In diesem Potenz-Intervall sind 6 Primzahlen SF_p situiert.

1.5 Die Bestimmung der Primzahlen SM_p = 2^(1+4*i) -9 im Intervall i:= 0 bis 1000:

```
• for i from 0 to 1000 do: if (isprime (2^(1+4*i)-9) and isprime (2^(1+4*i)-9))
     then print (("i", i, " = ", 2^(1+4*i)-9)) end_if: end_for:

                    "i", 1, " = ", 23

                    "i", 2, " = ", 503

                  "i", 4, " = ", 131063

                  "i", 5, " = ", 2097143

                "i", 8, " = ", 8589934583

        "i", 31, " = ", 42535295865117307932921825928971026423

      "i", 35, " = ", 2787593149816327892691964784081045188247543

"i", 71, " = ",

6216540455122333026942278101835260501255701884966846468005799711164493712656667\
1941623

"i", 80, " = ",

4271974071841820164790043412339104229205409044713305539894083215644439451561281\
100045924173873143

"i", 134, " = ",

4498913794543196382810538507685981858869697118301916633100755572611837580671487\
8703190406861038908571499209106335208951232082660554942996890085151808651638551\
3463

"i", 182, " = ",

2824013958708217496949108842204627863351353911851577524683401930862693830361198\
```

```
4999058739209952299969708978654982839965781232968658783909476265530884869461064\
30796091482716120572632072492703527723757359478834530365734903

"i", 710, " = ",

168351767899193495124165367999589642387398361820831325074602093259626203764520 5\
77603440238596771860290033270103031398685284845383060598467153605398524792790 98\
68613670050709713667183958929072622585315773137462924246986710207833268598226 57\
86081030051653511083204043185632037724171410892031421268841513278330271944393 83\
26011581406220001680780451820124300040917611583293307422368363412435850247409 18\
89072026980461637398051325653590023413733408046676357962594920208824317585058 65\
38190138566650616552733566674307484700363040101772605441604319576195682059174 74\
78876147124704755178819917177456824637563863198241582743466094795028790667459 27\
19008117121266997020250625288959157178415183486897843961681666457358837484734 12\
53925891830548195483200293817193509244883720077428420588463164917633143687531 12\
4723997114513110687276800591643371770275763415585129902093390315 43

"i", 841, " = ",

924560657882587998110083685081946262051770737167350096224003917614383393094311 0\
528357026680758175853534173918153199183648394688316293520472840047515930328623 9\
285252174466835583455110237418535066609502374681831351623775398747441654642702 9\
308943990867510857854286033297921604207916768104584354174900286824780030794826 7\
734887814953659736968186325535149557430250802637100848890794888607502068895837 6\
002279923152221791966958294821251291275798148785230241388977885088526687030301 1\
108359316460694748071219737829413475153477506818658938567020691165821035121342 9\
514163658920756596598158084031804812446318639715494448795489984377635074745060 0\
344613564576695994004532306335629024613376651092580146289928909056724176855107 8\
926817090104757654894191316911061740888653136705682720881075209256671254031567 1\
229377815881542478561362597788916430719482197907496801379611623782102221834983 3\
163116602772047690067095256259612361363359264010655176271889834492841576970838 6\
67605142083361306318165248993151948906255350031040563238490079223
```

<u>Anmerkung:</u> In diesem Intervall sind 12 Primzahlen SM_p, EZ3 situiert.

1.6 Die Bestimmung der Primzahlen $SF_p = 2^\wedge(1+4*i)+9$ im Intervall i:= 0 bis 1000:

- `for i from 0 to 1000 do: if (isprime (2^(1+4*i)+9) and isprime (2^(1+4*i)+9)) then print (("i", i, " = ", 2^(1+4*i)+9)) end_if: end_for:`

```
"i", 0, " = ", 11

"i", 1, " = ", 41

"i", 2, " = ", 521

"i", 9, " = ", 137438953481

"i", 14, " = ", 144115188075855881

"i", 79, " = ",

26699837949011376029937771327119401432533806529458159624338020097777746572258 00\
68752870260867081

"i", 174, " = ",

65751698769354668840512373536001608319379247535289673243728511288359151928713 51\
02764208317034698556403372871107022277214127429031272975482890976846331485106 24\
440295889105557477395796973554033965891165646697 2681

"i", 429, " = ",

73875860927002420996545465768306967726038665672927890558684264423239568181255 67\
47321788066586922125536827933697818591623337035719637107207634548797403302284 51\
53783727077340269105240653596212209328236829977000561171160601353019714984950 31\
22140044402280694600979616754997156907031755604105351275570793868641917744416 06\
29381030836835126819669388263816725087366766325086326695180780078488766378106 88\
41491777971210302562177144021123949168116897834247743963522769738506629596576 83\
42868790222766235969628443064056861656350 81

"i", 807, " = ",

10613435843100779633715067614103205107376901634176017502370812852710359829159 37\
73910508015418081478003323164771013916335080116022984215354331290709851140932 53\
14320892243625384009532666618490315839916227342858632596584900889632486158408 54\
59305523288221656507613232540553888757656036271369068476015791639533390967109 26\
48873261392161133506997630518471429577383383155067982676107102564068880588967 73\
```

```
27195651695079499251677679591434522358223407304006611405442241754079176774656 47\
68264335757534321601657747448695867123575536972567451242481622796361328296921 35\
50204716519755618738841904350151361275223824733153866223098784396067660494248 24\
27465296280081183469906419142055530241772910265392605536839693317942366602339 26\
81034667567079715950626655319007716273389737545649645875393078282836846525619 04\
12813140715981441699464909496096338693238625751141746990849472347932399710694 71\
95386513024051071423332967833344347268269003343215119804358402931666211767884 54\
214107413447847655453 4921
```

```
"i", 813, " = ",
```

```
17806390564184388968323857181641411837876547212732402765705563932549789229152 99\
72915175764440048326206100145316689101335956748382266714559013259979796591927 15\
21262950248402769061087560691440162275449596803624533653754574456395638089663 03\
91322797419952498710603284679110135132716697422859347754090775574744583946764 10\
32615886300074804365173675869658716384854973400333704004130689885153744851931 88\
68866412274911489611723479288428873146074348059529658497716806543240522985059 50\
37516070610067694092447798702021948103952547610475020406458256168507801888436 17\
08191137327070828279519821913382902080846555589426477486003249515025669072666 95\
26178960819491851261024949373280031486075633847110906789435549216887576003863 20\
06253772126109088372230873164454136158537467898667396917497875925606286691916 01\
89289522943041529938332987103645943106562216377006733588283162106728902334466 28\
19759373249131802030068464126107011649875901519384219942359866739959727473154 08\
265619032261584485063679981650 01
```

```
"i", 937, " = ",
```

```
36429544770832288502074468050256885503805075636030111714170018382468838747661 19\
55793544713191027330450048488538313576618496296238190724391740481559056211677 92\
29402934766340125251060807157397514656328358527382303632709407066219246190020 40\
41723330533404715047709286302548477638245421342812429822352828986948797257923 38\
44726426060774973712521607223993127380516477034765768351519981930883953914305 23\
03201910246860195310076274295805492451471375375386182938963292427679902450074 46\
89939762727503257810027678765192172704092097102232363873718993734076948821670 45\
23163953366636361464520379636843209281242957394439401560957342563196485394609 58\
69535561386590239372849672150205945899537164873446947679115056282496875704248 85\
07307739721445768213439632397766831994618801156025328331257860978483076658586 46\
24528207241575618975611425145393142738584492528976384251382333433064162558804 87\
05029127961786605815687585450251190490383054580121455061147118427031894816198 63\
80488038418180232773085373965122519100518688261805621055145178786014906363219 40\
95640240728467313710889901148072012919397926754212736523116130116415365908663 38\
908731764294758296453 21
```

<u>Anmerkung:</u> In diesem Intervall sind 11 Primzahlen SF_p situiert,

---

1.7 Die Bestimmung der Primzahlen $SF_p = 2^{(1+4*i)}+11$ im Exponenten-Intervall i: = 0 bis 1000:

```
• for i from 0 to 1000 do: if (isprime (2^(1+4*i)+11) and isprime (2^(1+4*i)+11))
     then print (("i", i, " = ", 2^(1+4*i)+11)) end_if: end_for:
```

```
"i", 0, " = ", 13

"i", 1, " = ", 43

"i", 2, " = ", 523

"i", 7, " = ", 536870923

"i", 19, " = ", 151115727451828646838283

"i", 74, " = ",

25462949704181076078355571105117227013143354920824203132951755616929766247041 70\
88272924683

"i", 143, " = ",

30916300184138066757562815127455898754391418637651479989257882011612153138316 48\
33962895501326553806236997089282520174174189206292883439012459432693877366459 89\
575846518587403

"i", 325, " = ",

43654031635809667726915363206024493430956148904759729031769174207747254899120 21\
37913939279394156542577813083450948966481034312574496081021580708428358188091 85\
21572809774924445315784601621897853208293558944085756592445163683467117510561 62\
87406754932894654157859056143107781994836250352595673965477299408294262365632 65\
952109990436535476413460079235532427513796766241535593516662802341308871476 3

"i", 415, " = ",
```

```
1025233522064550609150599760459418137798474425128157507795593545208417521081507\
4998385422003029406400565314484677173373685335973632364137237340373473616670981\
1591487109779522861403031808117006358885792554849766283364140303416552277084726\
6439132100566698693757944582036172832914974134611926105336715457203604597641533\
9206225680280181273002901135901986444212235593047099028940216393058892463104143\
7017406031273623925356391129186716520528778432339685799439304571802159393675524\
7218184236884328474759659 63
```

<u>Anmerkung:</u> In diesem Intervall sind 9 Primzahlen SF_p, EZ3 situiert.

---

2. Die Bestimmung von Primzahlen, die im Umfeld von $n_g = 2^{2+4n}$ situiert sind:

<u>Anmerkung:</u>
Die ungeraden Zahlen $n_u = n_g$ +1, 11, ... und $n_u = n_g$ -1, -11, ... scheiden aus, da sie die Endziffer, EZ5 haben oder aber durch 3 teilbar sind. Des weiteren scheiden die ungeraden Zahlen $n_u = n_g$ -7 ,17, .. aus und die ungeraden Zahlen $n_u = n_g$ +5, 15, ... , da diese Zahlen (stets9 durch 3 teilbar sind.

2.1 Die Bestimmung der Primzahlen SM_p = 2^(2+4*i)-3 im Intervall i:= 0 bis 1000:

-
```
for i from 0 to 1000 do: if (isprime (2^(2+4*i)-3) and isprime (2^(2+4*i)-3))
    then print (("i", i, " = ", 2^(2+4*i)-3)) end_if: end_for:
```

```
"i", 1, " = ", 61
"i", 2, " = ", 1021
"i", 3, " = ", 16381
"i", 5, " = ", 4194301
"i", 23, " = ", 19807040628566084398385987581
"i", 30, " = ", 5316911983139663491615228241121378301
"i", 37, " = ", 1427247692705959881058285969449495136382746621
"i", 43, " = ", 23945242826029513411849172299223580994042798784118781
"i", 66, " = ",

1185710993790117841137366886488964176417484642976159375764045660241030447512944\
61

"i", 173, " = ",

8218962346169333605064046692000201039922405941911209155466063911044893991089188\
7845526039629337319550421608888377784651765928628909121935361372105791435638280\
55036986138194684674474621694254245736395705 8371581

"i", 212, " = ",

7507516828804700229971157695509256861311759593549503536677899390762631562619231\
7079474101985803313808485540191847054626191826906663022432617614609066399051600\
3972692259090257733662834988914541231997976791790262615433033904468461711926461\
3887239597666074621

"i", 988, " = ",

1873280685673170040820266698114823161608937825083775079936008652114419601435932\
9966339021386790082648788229500497545846881233185419888203116629288235390311056\
3133430508555775252389290738676577490715921966913264135553620567825075687204901\
1190728522385897895017415362120980467028428950809775760667333287818293402985149\
1895345350924486937584993722921190833392519023192505422068344877371825080380559\
2870485628985545596762485076488292421768786265136558930078307499930877565922638\
6678508460762379168139456295222525146824470795952069244571656232154630980995650\
8351113518865354483497406593299987711551779705234850231767568440697637098873250\
0860509440579270480356911075829995837292303161382836390582279294807761319369715\
2649187482002393848668276657937535756433081371771545400577082813096740234313699\
2576480860824970831719003851753344727723021917009641756562764346708029742794770\
3982362817990533538701141090185946028073084154626832621439286101731602078522329\
0098550026681188138399943304550167805470348361023583353988476707759064192886549\
1850443985922031805005510498665700790114179888145661523285685820711791705604158\
3478616661967445741425111629716717450786395481366441550782221167694609042390577\
577981
```

<u>Anmerkung:</u> In diesem Intervall sind 12 Primzahlen SM_p, EZ1 situiert.

2.2 Die Bestimmung der Primzahlen $SF_p = 2^{(2+4*i)}+3$ im Intervall i:= 0 bis 1000:

```
• for i from 0 to 1000 do: if (isprime (2^(2+4*i)+3) and isprime (2^(2+4*i)+3))
      then print (("i", i, " = ", 2^(2+4*i)+3)) end_if: end_for:

                              "i", 0, " = ", 7

                              "i", 1, " = ", 67

                            "i", 4, " = ", 262147

                          "i", 7, " = ", 1073741827

"i", 97, " = ",

252172839656924666958585856640919128352510331330978858674869077787172619337582l\
479130513040312634601011624191379636227

"i", 277, " = ",

139089769374657110966000786263988918128997769925810641894248414180128060698034 3\
653089832149320454204651374414126473972067843965495627256829046328150302784381l\
211021823392143699745652883624633068305241252378852532749105598230661075510635 8\
128969759543099143779658958837607371612201667749100381674244476760662706843125 2\
3561526010025345027

"i", 592, " = ",

276114824032390615084035037184030932036361749728800100103097329416210820274966 7\
323357031030600103189306067331084936586541351762535284048666716596834162683197 6\
235551628701362361977386042927998876530915796124013370902650761940787927825665 5\
921394460732999949996632801043650547184727862520625299616061059761045530679268 8\
682527314218722473268146446037989958582420277457536591642269547126171138974313 9\
710045588397020318890149384902773825509969853304928180801457116383619066868869 2\
019910873754872360003241124056129957338023474912251468850325865417868147414698 6\
803118346575461417860240065205222035187754280473347524183893204495550946695955 2\
621816858945955126654861167405651637860039105938759790730221940882140523095549 5\
427

"i", 1000, " = ",

527281637372377240041555917694636545273607664437309107637121380096702771245133 7\
820431900849268848813256376302592286729215378727077696232512692424981004873615 4\
178697777547555825315882571087975720146346211696998057955328733557663329502480 0\
371396904357844441543150163116530617636743325023442017265891384145459632954000 3\
130724668987560084275641795235794138876861083528047906002450377943359104749867 7\
205114980934019186347978056217740868215214930812961133603263704677393459197890 8\
643783075760289726746502755464122633299473224245000705734258789296290114982688 7\
093477929694670129336702272735688781877528182428808101553748490730737943454477 6\
851500558266349781560272058991903255886992459081756995306754668516951816514223 3\
674978426629185219706344007173530903164400835249150694448024139158924806755759 9\
029741491061599587689237119302280666427189079562494768651505909135166210434585 7\
556646282138467826814979362011901116376350349195873694066126504363593557405796 0\
802274048843161958668755157732369282879143025595088344849481637605076510704426 3\
256431751503217361444570181897672230860342081974865384361005093020504215855688 5\
828023908989066705376072622590038061586845405950184249917778371116222221685450 8\
90018302827643797507
```

<u>Anmerkung:</u> In diesem Intervall sind 8 Primzahlen SF_p, EZ7 situiert.

2.3 Die Bestimmung der Zwillinge mod 6 (SM_p, SF_p), (2^(2+4*i)-3, 2^(2+4*i(+3): im Intervall i:= 0 bis 1000:

```
• for i from 0 to 1000 do: if (isprime (2^(2+4*i)-3) and isprime (2^(2+4*i)+3))
      then print (("i", i, 2^(2+4*i)-3, " = ", 2^(2+4*i)+3)) end_if: end_for:

                         "i", 1, 61, " = ", 67
```

<u>Anmerkung:</u> In diesem Intervall ist nur 1 Zwilling mod 6 situiert.

2.4 Die Bestimmung der Primzahlen $SM_p = 2^{(2+4*i)}-5$ im Intervall i:= 0 bis 1000:

```
• for i from 0 to 1000 do: if (isprime (2^(2+4*i)-5) and isprime (2^(2+4*i)-5))
     then print (("i", i, " = ", 2^(2+4*i)-5)) end_if: end_for:

"i", 1, " = ", 59

"i", 2, " = ", 1019

"i", 4, " = ", 262139

"i", 6, " = ", 67108859

"i", 16, " = ", 73786976294838206459

"i", 29, " = ", 332306998946228968225951765070086139

"i", 32, " = ", 1361129467683753853853498429727072845819

"i", 37, " = ", 1427247692705959881058285969449495136382746619

"i", 41, " = ", 93536104789177786765035829293842113257979682750459

"i", 51, " = ", 102844034832575377634685573909834406561420991602098741459288059

"i", 56, " = ",

107839786668602559178668060348078522694548577690162289924414440996859

"i", 137, " = ",

3685510180489786476798393145496356338786055879312930105836138965083617346086082\
8633653581300563073901772152099909803172849322115526609303052357756361647422301\
26362619

"i", 176, " = ",

3366486976990959044634233525043282345952217473806831270078899777963988578750131\
7261527465832176566087852691000679540593363324366401176344724018014532172037439\
7134314952220454284266480504596653905362768111090008059

"i", 202, " = ",

6828046779268970776657768233698567984276232522051489956245767999246374131525255\
2412379465736924715937207602284476756382165901328938182298422920765975867699672\
0144136142235090786619216477369756132246017652314788042925132599037589636215181\
0433019

"i", 454, " = ",

1872975588929836374042938036008377612079131825235270688301208488874753216671196\
8705062759812346995025534904787823738859732730378163170781114161803332934965988\
5629861135313063054246966257213626422691849579748334352927800840206147982519339\
3720520921159817672963247548345349958197716540800790095573618613899162713499295\
0000397378813600763731015622871590329821682095354994194066348461338042106517981\
0609662698567301845507920030146162104039858193030681609613958578384326129514739\
82374192367024383986850253624125958272304624247571630150111022242169094139
```

<u>Anmerkung:</u> In diesem Intervall sind 15 Primzahlen SM_p, EZ9 situiert.

2.5 Die Bestimmung der Primzahlen $SF_p = 2^{(2+4*i)}+7$ im Intervall i:= 0 bis 1000:

```
• for i from 0 to 1000 do: if (isprime (2^(2+4*i)-15) and isprime (2^(2+4*i)-15))
     then print (("i", i, " = ", 2^(2+4*i)-15)) end_if: end_for:

"i", 2, " = ", 1009

"i", 3, " = ", 16369

"i", 39, " = ", 365375409332725729550921208179070754913983135729
```

```
"i", 84, " = ",

5599361855444510526393605701421110695304113743086623837249972752409479677950402\
3634521937331790177892 9
```

```
"i", 411, " = ",

3128764410597383450776976808042657891230695877466300988145732254664360110722373\
9619096136483854389650162702895132975383561205974219861258658875651469777438296\
9944723845762704044809057031607075069841896224517109019055603953297583853407979\
2599890443623958416009352362170937600448529463537372147634019339610609734013470\
2167436768433170388802798876654011365393785379172055142029468972958045846875438\
5429095554423901139393283475301258912746516211974138792234205846564207134019545\
6598462636976100081649
```

```
"i", 462, " = ",

8044368900659986865145642164410452225898445753458275111961100256994644905673592\
5440020023021533440638547100970176777414996408274996531056548099387767339478615\
7503353937191036539576593981948049571022966230770439956298226457886727483056942\
2904889120465211294699971630095456984134159644617643111449406307073754896262089\
7442269929008538560225335041098550074093978111258453573784843895586811248165677\
0927975111937667481416961038461100336765740419544432637880591279259333245264067\
8929203665078255805806793336492646736420902369288875891213441129924485145042511\
4609
```

<u>Anmerkung:</u> In diesem Intervall sind 6 Primzahlen SM_p, EZ9 situiert

---

2.6 Die Bestimmung der Primzahlen $SF_p = 2^{(2+4*i)}+15$ im Intervall i:= 0 bis 1000:

```
• for i from 0 to 1000 do: if (isprime (2^(2+4*i)+15) and isprime (2^(2+4*i)+15))
     then print (("i", i, " = ", 2^(2+4*i)+15)) end_if: end_for:

"i", 0, " = ", 19

"i", 1, " = ", 79

"i", 2, " = ", 1039

"i", 5, " = ", 4194319

"i", 6, " = ", 67108879

"i", 7, " = ", 1073741839

"i", 10, " = ", 4398046511119

"i", 11, " = ", 70368744177679

"i", 49, " = ", 40173451106474756888549052308529065063055074844569820882535 9

"i", 77, " = ",

2085924839766513752338883849312032369167036351139187206514078201388864509576 56\
787131798913039

"i", 85, " = ",

8958978968711216842229769122273777112486581988938598139599956403855167484720643\
7815235099730864284631 19

"i", 127, " = ",

3351951982485649274893506249551461531869841455148098344430890360930441007518386\
744200468574541725856922507964546621512713438470702986642486608412251521039

"i", 136, " = ",

2303443862806116547998995715935222711741284924570581316147586853177260841303801\
7896033488312851921188607595062443626983030826322204130814407723597726029638938\
2897679

"i", 970, " = ",

3966826150550754530112319010338652467900795504600459281388196156157964949024761\
1731020043288568357340406875963375015036805290672721899883869212304786840639881\
4493191869464488465475122474436607036866992620957550707805214632390475902484319\
2361107835184997506096858773470977481699563875646518797811235622797201830892169\
1308599898804607912040368409212351718812904077861717584173096704291837139324259\
```

```
0021223927619364330376178592241086561602124822329677798932953784524362576675429\
9624891694899628489007795710852044240024078190585147188811452558342285915285463\
6936672126834379837192418659786058192985677843346165970295292192029546600990000\
3222040278118144261673577141724657073514937618333525178294077422049401995833459\
2609985165530267888899904068897479038133780666725035802678802134923662320552190\
3060714863073023213879056976045771040647059856964277194402601096666827490450292\
5946601291093495027083647447740755074326992397801546260260316636563131546305668\
8699341980374190699920848738614682498317212561052471501476584267358030054758203\
7929033812761470523810427617104767414337154224828542939165586503886876574044705\
233822125437769268306849264315624956237814799293507363043016719
```

Anmerkung: In diesem Intervall sind 14 Primzahlen SF_p, EZ9 situiert.

---

3. Die Bestimmung von Primzahlen, die im Umfeld der Zahlen $n_g = 2^{3+4n}$ auftreten:

Anmerkung:

Hier werden nur vier Beispiele angeführt.

3.1 Die Bestimmung der Primzahlen $SM_p = 2^(3+4*i)-9$ im Intervall i:= 0 bis 1000 situiert sind:

- ```
  for i from 0 to 1000 do: if (isprime (2^(3+4*i)-9) and isprime (2^(3+4*i) -9))
      then print (("i", i, " = " , 2^(3+4*i)-9)) end_if: end_for:
  ```

```
"i", 2, " = ", 2039

"i", 60, " = ",

  14134776518227074636666380005943348126619871175004951664972849610340958199

"i", 62, " = ",

  3618502788666131106986593281521497120414687020801267626233049500247285301239

"i", 140, " = ",

3019169939857233081793243664790615112733536976333152342700965040196499329913719\
0816689013801421270140331747000246110759198164677039398341060491474011461568349\
195162615799

"i", 174, " = ",

2630067950774186753620494941440064332775169901411586929749140451534366077148540\
4110568332681387942256134914844280891088565097161250919019315639073853259404249\
77611835564222299095831878942161358635646625867890679
```

Anmerkung: In diesem Intervall sind 5 Primzahlen SM_p, EZ9 situiert.

3.2 Die Bestimmung der Primzahlen $SF_p = 2^(3+4*i)+3$ im Intervall i:= 0 bis 1000:

- ```
  for i from 0 to 1000 do: if (isprime (2^(3+4*i)+3) and isprime (2^(3+4*i)+3))
      then print (("i", i, " = " , 2^(3+4*i)+3)) end_if: end_for:
  ```

```
"i", 0, " = ", 11

"i", 1, " = ", 131

"i", 3, " = ", 32771

"i", 13, " = ", 36028797018963971

"i", 16, " = ", 147573952589676412931

"i", 534, " = ",

8001303153100924247104732196883416239048548130052405150351687183009917783813377\
6785665937791152914510697265992113349037722912781794949200587025555109507232617\
4031203650701758789407605796506997065317112741503966831320317843640665309857857\
9140924246992991312855168098168737332743646321305569312397620289153742965270001\
9753526386022144853194998410236025450252527573627379481326670557066923467148515\
7310467205083604059283262963454595425336884037197494061422348253014606418486463\
7228894206111718605030402756580427391807398515375648801820275288851187818639040\
9292095213882488664762289538400072746520519914934169714369457189054633839067516\
771802021891
```

```
"i", 547, " = ",

3603466589878369714562740778167435879668130237629809021356863321480884516303903\
6850034358814682355697821008484655675792902374629852545390425157797005133533703\
5390918240583534399999046384055231020343489996406594144283080905327839709130518\
0259566738845028741800269066138261319450438496608387409579499741788649769069416\
0004148081054259208650462517813389133071859672161224172500889657427208487539866\
9076830299285315060281844393910524238389117108597050256860192544152141743019634\
3546577386890195341773144947448472869694640483664119962911340509038673143194793\
8058480224037074528081778172914433902282175238903846500159243407931394385096203\
218014298858652819767741645l
```

```
"i", 591, " = ",

3451435300404882688550437964800386650454521871610001251288716617702635253437084\
1541962887882501289866325841638561707331766897031691050608333957460427033539970\
2944395358767029524717325536599985956636447451550167136283134524259849097820819\
9017430759162499374957910013045631839809098281507816245200763247013069133490860\
8531591427734030915851830575474874482280253468219207395528369339077139237178924\
6375569854962753986126867311284672818874623166311602260018213954795238335860865\
0248885921935904500040514050701624466725293436403143360629073317723351842683733\
5038979332193267723253000815065275439846928505916844052298665056194386833699440\
7772710736824439083185764592570645473250488824234497384127774261026756538694369\
31
```

<u>Anmerkung:</u> In diesem Intervall sind 8 Primzahlen SF_p, EZ1 situiert.

3.3 Die Bestimmung der Primzahlen SM_p = 2^83+4*i)-15 im Intervall i:= 0 bis 1000:

- `for i from 0 to 1000 do: if (isprime (2^(3+4*i)-15) and isprime (2^(3+4*i)-15)) then print (("i", i, " = ", 2^(3+4*i)-15)) end_if: end_for:`

```
"i", 1, " = ", 113

"i", 5, " = ", 8388593

"i", 23, " = ", 39614081257132168796771975153

"i", 58, " = ",

552139707743245102994780468982162036196088717773630924413001937903943 53

"i", 472, " = ",

1768975428879058759761412883100751228457176510890668569929915598232022789990455\
5431586836913287845443221358367872979473207346867126982420547707572296619290500\
3181589000732075010141311018984309108660760024374857138986626655202581289584330\
8156625515054299485003069249709455211777587535363878889091967402227189302985551 6\
0625188832926510838831244308078647730240103056301819264150539646566942363637581\
9828869973876440116388277012474707154666672205576635258806894095235155936828145\
5863816501290723633493109978873172023163336077665517400543466670522723873156884\
64694851244392433
```

<u>Anmerkung:</u> In diesem Intervall sind 5 Primzahlen SM_p, EZ3 situiert.

3.4 Die Bestimmung der Primzahlen SF_p = 2^(3+4*i)+15 im Intervall i:= 0 bis 1000:

- `for i from 0 to 1000 do: if (isprime (2^(3+4*i)+15) and isprime (2^(3+4*i)+15)) then print (("i", i, " = ", 2^(3+4*i)+15)) end_if: end_for:`

```
"i", 0, " = ", 23

"i", 2, " = ", 2063

"i", 3, " = ", 32783

"i", 5, " = ", 8388623

"i", 38, " = ", 456719261665907161938651510223838443642478919 83

"i", 50, " = ", 1285550435407192220433569673872930082017762395026234268241102 3

"i", 128, " = ",

1072624634395407767965921999856467690198349265647391470217884915497741122405883\
75814414994385335227421520254865491888406830031062495572559571469192048672783
```

```
"i", 390, " = ",

3235066577115070294531819968361914239053966194634110384461980945616500908346581\
2901693267177451321489560325805462379782508001558348248348423210712427399128587\
8310628167094662616223691281230181914258943006826185063003544899878166133222730\
6832566545489799103403438264672442883395824354914066336610914156847116134415555\
4190591913984440137682316858261680920452433412935512205296745189603690746166513\
974465459105260118553503660956933507823228510045189460103981792760137829582 3
```

<u>Anmerkung:</u> In diesem Intervall sind 8 Primzahlen SF_p, EZ3 situiert.

---

4. Die Bestimmung von Primzahlen, die im Umfeld der geraden Zahlen $n_g = 2^{4+4n}$ auftreten:

4.1 Die Bestimmung der Primzahlen $SM_p = 2^(4+4*i)-3$ im Intervall i:= 0 bis 1000:

- ```
  for i from 0 to 1000 do: if (isprime (2^(4+4*i)-3) and isprime (2^(4+4*i)-3))
      then print (("i", i, " = ", 2^(4+4*i)-3)) end_if: end_for:
  ```

```
"i", 0, " = ", 13
"i", 2, " = ", 4093
"i", 4, " = ", 1048573
"i", 5, " = ", 16777213
"i", 28, " = ", 830767497365572420564879412675215 33

"i", 83, " = ",

1399840463861127631598401425355277673826028435771655959312493188102369919487600\
59086304843329475444733

"i", 112, " = ",

1162941958872971024878918092620807254965826177099708896450384318689022860981436\
6773219056811420217048972200345700258846936553626057834493

"i", 433, " = ",

3873222737369624529943682915673412349514935994832801788523145562593142672261417\
5193984562185472422735345244370256191376501612778337709966367630231989298254814\
3998696270192457500864841179265090680428263111498167021530544936217200033002962\
9290055959958294081095840130924394940047386508216518640956646037580253377038440\
8805692189538261497003082423405994316260534718304686005036294082579071834684490\
2076804128896990710971873048454703306145367213172208118714742590066216381793007\
52945992288313664324044557157168243884084806287 33
```

<u>Anmerkung:</u> In diesem Intervall sind 8 Primzahlen SM_p, EZ3 situiert.

4.3 Die Bestimmung der Zwillinge mod 6 (SM_p, SF_p), (2^(4+4*i)-3, 2^(4+4*i)+3) im Intervall i:= 0 bis 1000:

- ```
  for i from 0 to 1000 do: if (isprime (2^(4+4*i)-3) and isprime (2^(4+4*i)+3))
      then print (("i", i, 2^(4+4*i)-3, " = ", 2^(4+4*i)+3)) end_if: end_for:
  ```

```
"i", 0, 13, " = ", 19
"i", 2, 4093, " = ", 4099
```

<u>Anmerkung:</u> In diesem Intervall sind in der *Folge 4* zwei Zwillinge mod 6 situiert.

4.3 Die Bestimmung der Primzahlen $SF_p = 2^(4+4*i)+3$ im Intervall i: 0 bis 1000:

- ```
  for i from 0 to 1000 do: if (isprime (2^(4+4*i)+3) and isprime (2^(4+4*i)+3))
      then print (("i", i, " = ", 2^(4+4*i)+3)) end_if: end_for:
  ```

```
"i", 0, " = ", 19
```
```

"i", 2, " = ", 4099

"i", 3, " = ", 65539

"i", 6, " = ", 268435459

"i", 20, " = ", 19342813113834066795298819

"i", 56, " = ",

431359146674410236714672241392314090778194310760649159697657763987459

"i", 195, " = ",

1017458256970192607739235197558785674613152820177598291076089143640752752352543\
9562258044740099417557896316391896718201363966066977110847595769281085709884713\
8903161308502419410142185759152435680068435915159402496058513611411689167650819

"i", 425, " = ",

9018049429565725219304866426795284146244954305777330390464387746977486350250936\
8569554207347036392277444816125221996047770634243119764063431951757495520298402\
7561490121745638318510821972193873209013284909301826314838940594851039426873817\
4089360888950280102658644623412738880252899854016766513131200423418202849660163\
9328739115824600441854173372047940074141450967417764489157810068174898239515000\
5064914303126257636984514651016107076186144755157195308047213102845047558175883\
091660036899001903926128455371787862019

"i", 501, " = ",

2939214579902091582036052995014865879097133317347059713222765406273961629164468\
0034730482849702560509912216694758079047000246245398094216484503842717866321546\
0172772211999436801763274619494514870858053094562524786640935586934754211705131\
5866635938661655167911888957409508982517903956778228125804082440516642410724070\
0021377434209148110825999078639302784109824695476896212613634081852488010690884\
5781292048893428214830405175756437514347929224149123944676950789355316620691925\
9895604202498098104745742918537738894943385997525728932337460595428231060067395\
2044911495373010647749329399156163119321894151520259

Anmerkung: In diesem Intervall sind 9 Primzahlen SF_p, EZ9 situiert.

4.4 Die Bestimmung der Primzahlen $SM_p = 2^{(4+4*i)}-5$ im Intervall i:= 0 bis 1000:

- ```
  for i from 0 to 1000 do: if (isprime (2^(4+4*i)-5) and isprime (2^(4+4*i)-5))
      then print (("i", i, " = ", 2^(4+4*i)-5)) end_if: end_for:
  ```

"i", 0, " = ", 11

"i", 1, " = ", 251

"i", 2, " = ", 4091

"i", 4, " = ", 1048571

"i", 7, " = ", 4294967291

"i", 8, " = ", 68719476731

"i", 13, " = ", 72057594037927931

"i", 283, " = ",

9334156416755229106450255389283100404226045798254516456633819209429885527968133\
2888468725491574053638177252970693220591040739453666742173233541285538041115853\
9802138847055390214687853958373327630799145693082091096315273713300074746536989\
3212326053290542578425579035014591186920702461547563636124964893682474143407160\
77187244639253515768692731

"i", 306, " = ",

4622050384501541079063198370564118606036557611490203003743065735166521106191233\
1915789662070059070341631819962354836873925128265520372150146410501950443897647\
0878297914755898724212073143639381033096898778631081370868724051861263951355707\
2749900038873486151696084416697258142197059285474066136970301829657013867090765\
986365597716572168322172422029426139705149135001747451

"i", 591, " = ",
  ```

```
69028706000809765377100875929600773300909043743220002502577433235405270506874168\
30839257757650025797326516832771234146635337940633821012166679149208540670799940\
58887907175340590494346510731999719132728949031003342725662690485196981956416 39\
80348615183249987499158200260912636796181965630156324904015264940261382669817 21\
70631828554680618317036611509497489645605069364384147910567386781542784743578 49\
27511397099255079722537346225693456377492463326232045200364279095904766717217 30\
04977718438718090000810281014032489334505868728062867212581466354467036853674 67\
00779586643865354465060016301305508796938570118336881045973301123887736673988 81\
55454214736488781663715291851412909465009776484689947682555485220535130773887 38\
51
```

```
"i", 599, " = ",
```

```
29647603478997813412081369410588919663714309947763332481608231453689106133057 89\
60717457630202115461307371403034595170935724489293025075877350504677578646496 93\
50468585944797157383454673648565181435625631932061024707340075556129140962071 05\
26917848889094774330129329765083844215272776004651744683029590200319803225860 61\
13584103529903220339673080606794891142197240305184142245671365903090895689843 73\
72375118120856983337017065617798243006253276047264601304275834600352742058095 75\
85763334590299037671706477045583925877317151050709913211002807772816026679655 94\
45408422233980009645488034469133404238749146357925975400309760081625787339024 17\
76637361631858057511654164835691373754442584632201316999652679872760973429292 94\
919389413371
```

```
"i", 781, " = ",
```

```
41862623033468068226816457831765168742078132358509177215056788790893519829231 03\
00547648924165244600051187981067738572030345792068230907723301473677302284414 16\
84438264329132328665029475062947773269568808522956439204280692699340717567302 56\
24436842666590541602063689295220896308109577907607998937285242011991647569103 54\
83972909698820522583021973558150422331188102522746408530531402224774301164280 93\
04371717846888586921182427544360950745291433594016941970369701016371691323646 09\
46868424543063459295856346160002883738548165793011817614001836610093748507879 53\
69824900320107254622958344701250944607028381870039436598725818885723304391290 35\
65081530118964035238238597741717978789972392246098513875469383334602900031905 38\
72347331355664163980831139021249020261046053691597168155607852710645872319386 25\
53711094028744803444585414025072800326119124164194957694078110948831513689328 05\
960287029826564911101234001422088364261210760347496556221880373 7401491451
```

<u>Anmerkung:</u> In diesem Intervall sind 12 Primzahlen SM_p, EZ1 situiert.

4.5 Die Bestimmung der Primzahlen $SF_p = 2^{(4+4*i)}+7$ im Intervalll i:= 0 bis 1000:

- ```
  for i from 0 to 1000 do: if (isprime (2^(4+4*i)+7) and isprime (2^(4+4*i)+7))
      then print (("i", i, " = ", 2^(4+4*i)+7)) end_if: end_for:
  ```

```
"i", 0, " = ", 23

"i", 1, " = ", 263

"i", 3, " = ", 65543

"i", 4, " = ", 1048583

"i", 6, " = ", 268435463

"i", 10, " = ", 17592186044423

"i", 21, " = ", 30948500982134506872478106 3

"i", 39, " = ", 1461501637330902918203684832716283019655932542983

"i", 50, " = ", 2571100870814384440867139347745860164035524790052468536482 2023

"i", 146, " = ",
```

```
10130653244338361715118183260964748903838980059185636962880022777565070340363 54\
52792961597874685151227739206216096210673398319118052045295602706905129735441 57\
864213387210716610 63
```

```
"i", 471, " = ",
```

```
22112192860988234497017661038759390355714706386133357124123944977900284874880 69\
42894835461416098068040266979598412243415091835839087280256846344653707741131 25\
39769862509150937626766387737303863858259500304685714237332833190032266119804 13\
51957818938178743562538365621368190147219844192048486136495925278398662873193 95\
07814860411581385485390553850983096628001288203772740801881745582086779545469 77\
47860874673455501454853462655933839433333402569707940735086176190439449210351 81\
98297706266134045418663874735914650289541700970818967506793333381534048414461 05\
```
```

```
8086856405549063

"i", 661, " = ",

134099919832526597753105769650547407575091747469349475928044506252864596063622 8\
124781143819374817599752079000535478071096806897770604462878266395179780670028 1\
952051510755107848656421768532751711957832557419632412455512529392301553137322 7\
578087975738650857505379895637533426807185407417749377696450580307784776240802 8\
275131124210574642886112628493449376977121653483228471754150611744607622773655 7\
214665302684276992157753611609230381337651117939117746448468617428997029119700 6\
803027700731748111462478822683681852820202506137468559310155048665716430373773 7\
371888904521357058095521883338254455179341003420995801953884751621994013844086 4\
258402622589469101997187528496584828552989207480064590835842106708925817549682 0\
270191535858105066214773784035111692689096865461430777153004056440684444870363 2\
70086663

"i", 965, " = ",

151322408697157078938000450528665636745483226951616641288307043310469243966093 4\
895745088321249708455673480070624352075073444010647655482630508892241929650874 3\
839004206446246660822873019196953088259391503179838207542618356032961879824993 8\
673443139464759731525301314295615290897352747941838027870606827651870797382056 0\
123771663620170895082106338852399891617313540567845061652110927745507703373880 7\
297562558274054120268866676034587347473225586789309608418768073445295813624398 4\
208102863117203845558463886674958963013613822577863585998972036679927288638514 0\
874354252885222619521805521384661033362795938237997664272128760987455238380050 6\
714707957386708994357054792088495524349782471402493483669055077440239028771723 1\
468581587430201259189602053409480248952246882122994835002090535542437069723212 8\
260067553065224579386865881959753839136011499670573318267921489588425731294643 1\
196083118098964501460405252370481684659080215370237207804119744741940748073794 1\
310857466902701976773103665871226596768082144205187664088309641546555711927726 8\
902932503233393498375336746868315407346235436432973592344878635554766715013302 0\
49019703881750841839097948620438574075234024020901007196 23
```

Anmerkung: In diesem Intervall sind 13 Primzahlen SF_p, EZ3 situiert.

Schlussfolgerungen:

1. Mit den heute verfügbaren Test-Programmen und unter Beachtung der vier Folgen, in denen die geraden Zahlen $n_g = 2^n$ (zyklisch) auftreten, können Primzahlen vom Typ SM_p und SF_p mit vorgegebenen Endziffern relativ schnell bestimmt werden.

2. Die hier angewandte (und demonstrierte) Methode kann als eine " vertikale Primzahl-Suche" bezeichnet werden.

3. Sofern bestimmte Zahlen $n_g = 2^n$ der vier Folgen vorgegeben werden, können die in definierten Intervallen in der Umgebung dieser geraden Zahlen auftretenden Primzahlen bestimmt werden. Dieses "Primzahl - Bestimmungs-Verfahren" kann als "horizontale Primzahl-Suche" bezeichnet werden.

4. Nachfolgend werden vier Beispiele für die "horizontale Primzahl-Suche" vorgestellt.

5. Die "horizontale" Bestimmung von Primzahlen im Umfeld gerader Zahlen $n_g = 2^n$:

5.1 Die Bestimmung der Primzahlen im Intervall i:= -500 bis +1000, die im Umfeld der geraden Zahl $n_g = 2$^2000 auftreten:

- ```
  for i from -500 to +1000 do: if isprime (2^2000+i)
     then print ("i", i, " = ", 2^2000+i) end_if: end_for:
  ```

```
"i", 841, " = ",

114813069527425452423283320117768198402231770208869520047764273682576626139237 0\
313856659486316506269918445964638987462773447118960863055331425931356166653185 3\
912998914531228000068877914824004487142892699006348624478161546364638836394731 7\
026040466353970904996558162398808944629605623311649536164221970332681344168908 9\
844585056023794848079140589009347765004290027167066258305220081322362812917612 6\
788331720659899539641812702177985840404215985318325154088943390209192055495778 3\
589672039160081957216630582755380425583726015528348786419432054508915275783882 6\
25175435528800822842770817965453762184851149030217
```
```

"i", 925, " = ", (84)

```
11481306952742545242328332011776819840223177020886952004776427368257662613923 70\
313856659486316506269918445964638987462773447118960863055331425931356166653185 3\
912998914531228000068877914824004487142892699006348624478161546364638836394731 7\
026040466353970904996558162398808944629605623311649536164221970332681344168908 9\
844585056023794848079140589009347765004290027167066258305220081322362812917612 6\
788331720659899539641812702177985840404215985318325154088943390209192055495778 3\
589672039160081957216630582755380425583726015528348786419432054508915275783882 6\
2517543552880082284277081796545376218485114903030 1
```

<u>Anmerkung:</u>

1. Im Umfeld der Zahl n_g = 2^2000 sind im Intervall i:= -500 bis +1000 zwei P4rimzahlen mit den Endziffern: EZ1 und EZ7 situiert.

5.2

- `for i from -500 to +1000 do: if isprime (2^2001+i)`
  `then print ("i", i, " = ", 2^2001+i) end_if: end_for:`

"i", 137, " = ",

```
229626139054850904846566640235536396804463540417739040095528547365153252278474 0\
627713318972633012539836891929277974925546894237921726110662851862712333306370 7\
825997829062456000137755829648008974285785398012697248956323092729277672789463 4\
052080932707941809993116324797617889259211246623299072328443940665362688337817 9\
689170112047589696158281178018695530008580054334132516610440162644725625835225 3\
576663441319799079283625404355971680808431970636650308177886780418384110991556 7\
179344078320163914433261165510760851167452031056697572838864109017830551567765 2\
503508710576016456855416359309075243697022980588 89
```

"i", 641, " = ", (504)

```
229626139054850904846566640235536396804463540417739040095528547365153252278474 0\
627713318972633012539836891929277974925546894237921726110662851862712333306370 7\
825997829062456000137755829648008974285785398012697248956323092729277672789463 4\
052080932707941809993116324797617889259211246623299072328443940665362688337817 9\
689170112047589696158281178018695530008580054334132516610440162644725625835225 3\
576663441319799079283625404355971680808431970636650308177886780418384110991556 7\
179344078320163914433261165510760851167452031056697572838864109017830551567765 2\
503508710576016456855416359309075243697022980593 93
```

<u>Anmerkung:</u>

1. Im Umfeld der geraden Zahl n_g = 2^2001 sind im Intervall i:= -500 bis +1000 zwei Primzahlen mit den Endziffern: EZ3 und EZ9 situiert.

5.3 Die Bestimmung der Primzahlen im Intervall i:= -500 bis +1000, die im Umfeld der geraden Zahl n_g = 2^2002: auftreten.

- `for i from -500 to +1000 do: if isprime (2^2002+i)`
  `then print ("i", i, " = ", 2^2002+i) end_if: end_for:`

"i", -297, " = ",

```
459252278109701809693133280471072793608927080835478080191057094730306504556948 1\
255426637945266025079673783858555949851093788475843452221325703725424666612741 5\
651995658124912000275511659296017948571570796025394497912646185458555345578926 8\
104161865415883619986232649595235778518422493246598144656887881330725376675635 9\
378340224095179392316562356037391060017160108668265033220880325289451251670450 7\
153326882639598158567250808711943361616863941273300616355773560836768221983113 4\
358688156640327828866522331021521702334904062113395145677728218035661103135530 5\
007017421152032913710832718618150487394045961172 07
```

Anmerkung: Im Umfeld der Zahl n_g = 2^2002 ist im Intervall i:= -500 bis +1000 eine Primzahl mit der Endziffer EZ7 situiert.

5.4 Die Bestimmung der im Umfeld der geraden Zahl n_g = 2^2003 situierten Primzahlen:

- ```
  for i from -500 to +1000 do: if isprime (2^2003+i)
      then print ("i", i, " = ", 2^2003+i) end_if: end_for:
  ```

```
"i", 533, " = ",

918504556219403619386266560942145587217854161670956160382114189460613009l138962\
510853275890532050159347567717111899702187576951686904442651407450849333225483l\
303991316249824000551023318592035897143141592050788995825292370917110691157853 6\
208323730831767239972465299190471557036844986493196289313775762661450753351271 8\
756680448190358784633124712074782120034320217336530066441760650578902503340901 4\
306653765279196317134501617423886723233727882546601232711547121673536443966226 8\
717376313280655657733044662043043404669808124226790291355456436071322206271061 0\
01403484230406582742166543723630097478809192235541
```

Anmerkung:

1. Im Umfeld der geraden Zahl $n_g = 2\text{^}2003$ ist im Intervall $i := -500$ bis $+1000$ nur eine Primzahl mit der Endziffer EZ1 situiert.

5.5 Die Bestimmung der Primzahlen im Intervall $i := -500$ bis $+1000$, die im Umfeld der geraden Zahl $n_g = 2\text{^}2004$ auftreten:

- ```
  for i from -500 to +1000 do: if isprime (2^2004+i)
      then print ("i", i, " = ", 2^2004+i) end_if: end_for:
  ```

```
"i", -213, " = ",

18370091124388072387725331218842911744357083233419123207642283789212260182277 92\
50217065517810641003186951354342237994043751539033738088853028149016986664509 66\
26079826324996480011020466371840717942862831841015779916505847418342213823157 07\
24166474616635344799449305983809431140736899729863925786275515253229015067025 43\
75133608963807175692662494241495642400686404346730601328835213011578050066818 02\
86133075305583926342690032348477734464674557650932024654230942433470728879324 53\
74347526265613113154660893240860868093396162484535805827109128721426444125421 22\
0028069684608131654843330874472601949576183844698O3

"i", 823, " = ",                                                        (1036)

18370091124388072387725331218842911744357083233419123207642283789212260182277 92\
50217065517810641003186951354342237994043751539033738088853028149016986664509 66\
26079826324996480011020466371840717942862831841015779916505847418342213823157 07\
24166474616635344799449305983809431140736899729863925786275515253229015067025 43\
75133608963807175692662494241495642400686404346730601328835213011578050066818 02\
86133075305583926342690032348477734464674557650932024654230942433470728879324 53\
74347526265613113154660893240860868093396162484535805827109128721426444125421 22\
0028069684608131654843330874472601949576183844708 39

"i", 967, " = ",                                                        (144)

18370091124388072387725331218842911744357083233419123207642283789212260182277 92\
50217065517810641003186951354342237994043751539033738088853028149016986664509 66\
26079826324996480011020466371840717942862831841015779916505847418342213823157 07\
24166474616635344799449305983809431140736899729863925786275515253229015067025 43\
75133608963807175692662494241495642400686404346730601328835213011578050066818 02\
86133075305583926342690032348477734464674557650932024654230942433470728879324 53\
74347526265613113154660893240860868093396162484535805827109128721426444125421 22\
0028069684608131654843330874472601949576183844709 83
```

Anmerkung:

1. Im Intervall $i := -500$ bis $+1000$ sind im Umfeld der geraden Zahl $n_g = 2\text{^}2004$ drei Primzahlen mit den Endziffern: EZ3, EZ9 situiert.

2. Diese drei Primzahlen bilden Zwillinge mod 2n, n> 1, die in der Tabelle in Klammern vermerkt wurden,

---

5.6

- ```
  for i from -500 to +1000 do: if isprime (2^2005+i)
      then print ("i", i, " = ", 2^2005+i) end_if: end_for:
  ```
  ```

```
"i", -235, " = ",

3674018224877614477545066243768582348871416646683824641528456757842452036455585\
0043413103562128200637390270868447598808750307806747617770605629803397332901932\
5215965264999296002204093274368143588572566368203155983301169483668442764631414\
4833294923327068959889861196761886228147379945972785157255103050645803013405087\
5026721792761435138532498848299128480137280869346120265767042602315610013363605\
7226615061116785268538006469695546892934911530186404930846188486694145775864907\
4869505253122622630932178648172173618679232496907161165421825744285288825084244\
005613936921626330968666174894520389915236768939797

"i", 27, " = ",                                                     (262)

3674018224877614477545066243768582348871416646683824641528456757842452036455585\
0043413103562128200637390270868447598808750307806747617770605629803397332901932\
5215965264999296002204093274368143588572566368203155983301169483668442764631414\
4833294923327068959889861196761886228147379945972785157255103050645803013405087\
5026721792761435138532498848299128480137280869346120265767042602315610013363605\
7226615061116785268538006469695546892934911530186404930846188486694145775864907\
4869505253122622630932178648172173618679232496907161165421825744285288825084244\
005613936921626330968666174894520389915236768940059

"i", 251, " = ",                                                    (224)

3674018224877614477545066243768582348871416646683824641528456757842452036455585\
0043413103562128200637390270868447598808750307806747617770605629803397332901932\
5215965264999296002204093274368143588572566368203155983301169483668442764631414\
4833294923327068959889861196761886228147379945972785157255103050645803013405087\
5026721792761435138532498848299128480137280869346120265767042602315610013363605\
7226615061116785268538006469695546892934911530186404930846188486694145775864907\
4869505253122622630932178648172173618679232496907161165421825744285288825084244\
005613936921626330968666174894520389915236768940283

"i", 279, " = ",                                                    (28)

3674018224877614477545066243768582348871416646683824641528456757842452036455585\
0043413103562128200637390270868447598808750307806747617770605629803397332901932\
5215965264999296002204093274368143588572566368203155983301169483668442764631414\
4833294923327068959889861196761886228147379945972785157255103050645803013405087\
5026721792761435138532498848299128480137280869346120265767042602315610013363605\
7226615061116785268538006469695546892934911530186404930846188486694145775864907\
4869505253122622630932178648172173618679232496907161165421825744285288825084244\
005613936921626330968666174894520389915236768940311
```

<u>Anmerkung:</u>

1. Im Umfeld der geraden Zahl $n_g = 2\hat{}2005$ sind im Intervall i:= -500 bis +1000 vier Primzahlen mit den Endziffern: EZ1, EZ3, EZ7 und EZ9 situiert.

-

5.7 Die Bestimmung der Primzahlen im Intervall i:= -500 bis +1000, die im Umfeld der geraden Zahl $n_g = 2\hat{}2006$ auftreten:

```
• for i from -500 to +1000 do: if isprime (2^2006+i)
     then print ("i", i, " = ", 2^2006+i) end_if: end_for:

"i", -57, " = ",

7348036449755228955090132487537164697742833293367649283056913515684904072911170\
0086826207124256401274780541736895197617500615613495235541211259606794665803865\
0431930529998592004408186548736287177145132736406311966602338967336885529262828\
9666589846654137919779722393523772456294759891945570314510206101291606026810175\
0053443585522870277064997696598256960274561738692240531534085204631220026727211\
4453230122233570537076012939391093785869823060372809861692376973388291551729814\
9739010506245245261864357296344347237358464993814322330843651488570577650168488\
011227873843252661937332349789040779830473537880007

"i", 943, " = ",                                                    (1000)

7348036449755228955090132487537164697742833293367649283056913515684904072911170\
0086826207124256401274780541736895197617500615613495235541211259606794665803865\
0431930529998592004408186548736287177145132736406311966602338967336885529262828\
9666589846654137919779722393523772456294759891945570314510206101291606026810175\
0053443585522870277064997696598256960274561738692240531534085204631220026727211\
4453230122233570537076012939391093785869823060372809861692376973388291551729814\
9739010506245245261864357296344347237358464993814322330843651488570577650168488\
```

```
011227873843252661937332349789040779830473537881007
```

Anmerkung:

1. Im Intervall i:= -500 bis +1000 sind im Umfeld der geraden Zahl n_g = 2^2006 zwei Primzahlen mit den Endziffern: EZ7 situiert, die einen Zwilling mod 1000 bilden.

5.8 Die Bestimmung der Primzahlen im Intervall i:= -500 bis +1000, die im Umfeld der geraden Zahl n_g = 2^2007 auftreten:

Anmerkung:

In diesem Intervall ist im Umfeld von n_g = 2^2007 keine Primzahl situiert. Das Intervall wurde deshalb zu i:= -2500 bis +2500 erweitert.

```
• for i from -2500 to +2500 do: if isprime (2^2007+i)
    then print ("i", i, " = ", 2^2007+i) end_if: end_for:

"i", -2499, " = ",

1469607289951045791018026497507432939548566658673529856611382703136980814582234\
0017365241424851280254956108347379039523500123122699047108242251921358933160773\
0086386105999718400881637309747257435429026547281262393320467793467377105852565\
7933317969330827583955944478704754491258951978389114062902041220258321205362035\
0010688717104574055412999539319651392054912347738448106306817040926244005345442\
2890646024446714107415202587878218757173964612074561972338475394677658310345962\
9947802101249049052372871459268869447471692998762864466168730297714115530033697\
6022455747686505323874664699578081559660947075757629

"i", -2265, " = ",                                                          (234)

1469607289951045791018026497507432939548566658673529856611382703136980814582234\
0017365241424851280254956108347379039523500123122699047108242251921358933160773\
0086386105999718400881637309747257435429026547281262393320467793467377105852565\
7933317969330827583955944478704754491258951978389114062902041220258321205362035\
0010688717104574055412999539319651392054912347738448106306817040926244005345442\
2890646024446714107415202587878218757173964612074561972338475394677658310345962\
9947802101249049052372871459268869447471692998762864466168730297714115530033697\
6022455747686505323874664699578081559660947075757863

"i", 1883, " = ",                                                          (4148)

1469607289951045791018026497507432939548566658673529856611382703136980814582234\
0017365241424851280254956108347379039523500123122699047108242251921358933160773\
0086386105999718400881637309747257435429026547281262393320467793467377105852565\
7933317969330827583955944478704754491258951978389114062902041220258321205362035\
0010688717104574055412999539319651392054912347738448106306817040926244005345442\
2890646024446714107415202587878218757173964612074561972338475394677658310345962\
9947802101249049052372871459268869447471692998762864466168730297714115530033697\
6022455747686505323874664699578081559660947075762011
```

Anmerkung:

1. Im erweiterten Intervall i:= -2500 bis +2500 sind in der Umgebung der Zahl n_g = 2^2007 drei Primzahlen: p_1, p_2, p_3 mit den Endziffern EZ9, EZ3, EZ1 situiert.

2. Die Primzahlen (p_2, p_3) bilden einen Zwilling mod 4148.

5.9 Die Bestimmung der im Intervall i:= -500 bis +1000 in Umfeld der geraden Zahl n_g = 2^2008 situierten Primzahlen:

```
• for i from -2500 to +2500 do: if isprime (2^2008+i)
    then print ("i", i, " = ", 2^2008+i) end_if: end_for:

"i", 3, " = ",

2939214579902091582036052995014865879097133317347059713222765406273961629164468\
0034730482849702560509912216694758079047000246245398094216484503842717866321546\
0172772211999436801763274619494514870858053094562524786640935586934754211705131\
5866635938661655167911888957409508982517903956778228125804082440516642410724070\
0021377434209148110825999078639302784109824695476896212613634081852488010690884\
5781292048893428214830405175756437514347929224149123944676950789355316620691925\
9895604202498098104745742918537738894943385997525728932337460595428231060067395\
2044911495373010647749329399156163119321894151520259

"i", 81, " = ",                                                          (78)
```

293921457990209158203605299501486587909713331734705971322276540627396162916446 8\
003473048284970256050991221669475807904700024624539809421648450384271786632154 6\
017277221199943680176327461949451487085805309456252478664093558693475421170513 1\
586663593866165516791188895740950898251790395677822812580408244051664241072407 0\
002137743420914811082599907863930278410982469547689621261363408185248801069088 4\
578129204889342821483040517575643751434792922414912394467695078935531662069192 5\
989560420249809810474574291853773889494338599752572893233746059542823106006739 5\
20449114953730106477493293991561631193218941515203 37

<u>Anmerkung:</u>

1. Im Intervall i:= -500 bis +1000 sind im Umfeld der geraden Zahl $n_g = 2\char`^2008$ zwei Primzahlen mit den Endziffern: EZ7 und EZ9 situiert.

<u>Die Schlussfolgerungen für die Bestimmung sehr großer Primzahlen im Umfeld von $n_g = 2^n$ $(n \to \infty)$:</u>

1. Durch den "Rückgriff" auf die erweiterten Beweis-Gleichungen des Hauptsatzes der Primzahl-Theorie, die in **Tab. 1** angegeben wurden, und durch die in dieser Monographie eingeführte Gliederung der Zahlen $n_g = 2^n$ in die vier Folgen: *Folge* 1 (EZ2), *Folge* 2 (EZ4), *Folge* 3 (EZ8) *und Folge* 4 (EZ6), sind neue Grundlagen für die Bestimmung sehr großer Primzahlen vom Typ: $SM_p = 2^n - n_u$ und vom Typ: $SF_p = 2^n + n_u$ mit vorbestimmten Endziffern: EZ1, EZ3, EZ7 und EZ9 geschaffen worden.

2. Für die Bestimmung sehr großer Primzahlen, die in den Umgebungen der Zahlen $n_g = 2^n$, $n \to \infty$, auftreten, bieten sich zwei Methoden an - die "vertikale Methode" und die "horizontale Methode".

3. Die "vertikale Methode" bietet die Möglichkeit sehr große Primzahlen vom Typ: $SM_p = 2^n - n_u$ und $SF_p = 2^n + n_u$ mit "vor-definierten" Endziffern zu bestimmen, und dabei die subtrahierten bzw. die addierten ungeraden Zahlen (n_u) auszuschließen, die zu nicht primen Zahlen mit der Endziffer EZ5 führen bzw. zu ungeraden Zahlen (n_u), die durch 3 teilbar sind. Das hier mögliche Vorgehen kann an zwei Beispielen erläutert werden:

<u>Beispiel 1:</u> Es soll hier definiert werden, welche ungeraden Zahlen (n_u) bei der Addition bzw. bei der Subtraktion zu den geraden Zahlen der *Folge 1* := 2^(1+4*n) niemals Primzahlen ergeben können. Dabei genügt der Nachweis für die Zahl $n_g = 2^5 = 32$.

1. $n_u = 32 - 7 = 25$ ... Die Subtraktion der Zahl 7 führt in der *Folge 1* stets zu nicht primen Zahlen.
   $= 32 - 17 = 15$
   usw.

2. $n_u = 32 + 3 = 35$ ... Die Addition der Zahl 3 führt in der *Folge 1* stets zu nicht primen Zahlen
   $= 32 + 13 = 45.$
   usw.

3. $n_u = 32 + 7 = 39$ QS = 12 ; dagegen ist $n_u = 32 + 19 = 41$ eine Primzahl
4. $n_u = 32 + 19 = 51$ QS = 6 ; dagegen ist $n_u = 32 + 29 = 61$ eine Primzahl

---

<u>Beispiel 2:</u> Es wird hier definiert, welche ungeraden Zahlen (n_u) bei der Addition bzw. bei der Subtraktion zu den geraden Zahlen der *Folge 2* := 2^(2+4*i) niemals Primzahlen ergeben können. Dabei genügt der Nachweis für die Zahl $n_g = 2^6 = 64$.

1. $n_u = 64 + 1 = 65$
   $64 + 11 = 75$
   usw.

2. $n_u = 64 - 9 = 55$
   $64 - 19 = 45$
   usw.

Für die Zahlen der *Folge 3* := 2^(3+4*i), EZ8 ergeben sich die Zahlen := -3, -13, usw und die Zahlen := +7, +17, usw., die bei Zufügung zu den Zahlen dieser *Folge* keine Primzahlen bilden können.

Für die Zahlen der *Folge 4* := 2^(4+4*i), EZ6 ergeben sich die Zahlen := -1, -11, usw. und die Zahlen := +9, +19, usw., die bei Zufügung zu den Zahlen dieser *Folge* keine Primzahlen bilden können.

---

6. Die Modifikation des Verfahrens zur Bestimmun sehr großer Primzahlen mit definierten Endziffern mit dem MuPAD - Testprogramm:

Unter Beachtung der (im "vertikalen Testverfahren" bereits erkannten und einzuhaltenden Restriktionen bei der Addition bzw. Subtraktion von ungeraden Zahlen zu den geraden Zahlen der *Folgen* := 2^(1+4*i), 2^(2+4*i), 2^(3+4*i) und 2^(4+4*i), lassen sich Primzahlen vom Typ: SM_p und SF_p mit definierten Endziffern auch im "horizontalen Testverfahren" bestimmen. Diese Methode hat den Vorteil, dass mehrere Rechner parallel an der Bestimmung großer Primzahlen mit vorgegebener Endziffer beteiligt sein können.

Das derart modifizierte "horizontale Testverfahren" wird nachfolgend an einigen Beispielen demonstriert:

6.1 Die Bestimmung von Primzahlen mit der Endziffer EZ1 im Umfeld der Zahl n_g = 2^2000::

- ```
  for i from 0 to 1000 do: if isprime (2^2000+5+10*i) then print
      ("i", i, 5+10*i," = ", 2^2000+5+10*i) end_if: end_for:
  ```

```
"i", 92, 925, " = ",

114813069527425452423283320117768198402231770208869520047764273682576626139237O\
313856659486316506269918445964638987462773447118960863055331425931356166653185З\
9129989145312280000688779148240044871428926990063486244781615463646388363947317\
0260404663539709049965581623988089446296056233116495361642219703326813441689089\
8445850560237948480791405890093477650042900271670662583052200813223628129176126\
7883317206598995396418127021779858404042159853183251540889433902091920554957783\
5896720391600819572166305827553804255837260155283487864194320545089152757838826\
25175435528800822842770817965453762184851149030301
```

```
"i", 785, 7855, " = ",                                                (6930)

1148130695274254524232833201177681984022317702088695200477642736825766261392370\
3138566594863165062699184459646389874627734471189608630553314259313561666531853\
9129989145312280000688779148240044871428926990063486244781615463646388363947317\
0260404663539709049965581623988089446296056233116495361642219703326813441689089\
8445850560237948480791405890093477650042900271670662583052200813223628129176126\
7883317206598995396418127021779858404042159853183251540889433902091920554957783\
5896720391600819572166305827553804255837260155283487864194320545089152757838826\
25175435528800822842770817965453762184851149037231
```

<u>Anmerkung:</u>
In diesem Intervall sind 2 660-stellige Primzahlen mit der Endziffer EZ1 in einem Zwilling mod 6930 situiert.

6.2 Die Bestimmung von Primzahlen mit der Endziffer EZ3 im Umfeld der Zahl n_g = 2^2000:

- ```
  for i from 501 to 1501 do: if isprime (2^2000+7+10*i)
      then print ("i",i, 7+10*i, " = ", 2^2000 +7 +10*i) end_if: end_for:
  ```

```
"i", 501, 5017, " = ",

1148130695274254524232833201177681984022317702088695200477642736825766261392370\
3138566594863165062699184459646389874627734471189608630553314259313561666531853\
9129989145312280000688779148240044871428926990063486244781615463646388363947317\
0260404663539709049965581623988089446296056233116495361642219703326813441689089\
8445850560237948480791405890093477650042900271670662583052200813223628129176126\
7883317206598995396418127021779858404042159853183251540889433902091920554957783\
5896720391600819572166305827553804255837260155283487864194320545089152757838826\
25175435528800822842770817965453762184851149034393

"i", 1238, 12387, " = ",                                              (7370)

1148130695274254524232833201177681984022317702088695200477642736825766261392370\
3138566594863165062699184459646389874627734471189608630553314259313561666531853\
9129989145312280000688779148240044871428926990063486244781615463646388363947317\
0260404663539709049965581623988089446296056233116495361642219703326813441689089\
```

```
84458505602379484807914058900934776500429002716706625830522008132236281291761 26\
78833172065989953964181270217798584040421598531832515408894339020919205549577 83\
58967203916008195721663058275538042558372601552834878641943205450891527578388 26\
2517543552880082284277081796545376218485114904 1763
```

"i", 1334, 13347, " = ", (960)

```
11481306952742545242328332011776819840223177020886952004776427368257766261392370\
3138566594863165062699184459646389874627734471189608630553314259313561666531853\
91299891453122800000688779148240044871428926990063486244781615463646388363947317\
02604046635397090499655816239880894462960562331164953616422197033268134416890 89\
84458505602379484807914058900934776500429002716706625830522008132236281291761 26\
78833172065989953964181270217798584040421598531832515408894339020919205549577 83\
58967203916008195721663058275538042558372601552834878641943205450891527578388 26\
251754355288008228427708179654537621848511490427 23
```

"i", 1389, 13897, " = ", (550)

```
11481306952742545242328332011776819840223177020886952004776427368257766261392370\
3138566594863165062699184459646389874627734471189608630553314259313561666531853\
91299891453122800000688779148240044871428926990063486244781615463646388363947317\
02604046635397090499655816239880894462960562331164953616422197033268134416890 89\
84458505602379484807914058900934776500429002716706625830522008132236281291761 26\
78833172065989953964181270217798584040421598531832515408894339020919205549577 83\
58967203916008195721663058275538042558372601552834878641943205450891527578388 26\
251754355288008228427708179654537621848511490432 73
```

"i", 1484, 14847, " = ", (950)

```
11481306952742545242328332011776819840223177020886952004776427368257766261392370\
3138566594863165062699184459646389874627734471189608630553314259313561666531853\
91299891453122800000688779148240044871428926990063486244781615463646388363947317\
02604046635397090499655816239880894462960562331164953616422197033268134416890 89\
84458505602379484807914058900934776500429002716706625830522008132236281291761 26\
78833172065989953964181270217798584040421598531832515408894339020919205549577 83\
58967203916008195721663058275538042558372601552834878641943205450891527578388 26\
251754355288008228427708179654537621848511490442 23
```

Anmerkung: In diesem Intervall sind fünf 660-stellige Primzahlen mit der Endziffer EZ3 situiert.

6.3 Die Bestimmung von Primzahlen mit der Endziffer EZ7 im Umfeld der Zahl $n_g = 2^{\wedge}2000$:

- ```
  for i from 0 to 1000 do: if isprime (2^2000+1+10*i)
  then print ("i", i, 1+10*i, " = ", 2^2000+1+10*i) end_if: end_for:
  ```

"i", 84, 841, " = ",

```
11481306952742545242328332011776819840223177020886952004776427368257766261392370\
3138566594863165062699184459646389874627734471189608630553314259313561666531853\
91299891453122800000688779148240044871428926990063486244781615463646388363947317\
02604046635397090499655816239880894462960562331164953616422197033268134416890 89\
84458505602379484807914058900934776500429002716706625830522008132236281291761 26\
78833172065989953964181270217798584040421598531832515408894339020919205549577 83\
58967203916008195721663058275538042558372601552834878641943205450891527578388 26\
251754355288008228427708179654537621848511490302 17
```

"i", 524, 5241, " = ", (4400)

```
11481306952742545242328332011776819840223177020886952004776427368257766261392370\
3138566594863165062699184459646389874627734471189608630553314259313561666531853\
91299891453122800000688779148240044871428926990063486244781615463646388363947317\
02604046635397090499655816239880894462960562331164953616422197033268134416890 89\
84458505602379484807914058900934776500429002716706625830522008132236281291761 26\
78833172065989953964181270217798584040421598531832515408894339020919205549577 83\
58967203916008195721663058275538042558372601552834878641943205450891527578388 26\
251754355288008228427708179654537621848511490346 17
```

Anmerkung:
In diesem Intervall sind 2 Primzahlen mit der Endziffer EZ7 situiert, die einen Zwilling mod 4400 bilden.

6.4 Die Bestimmung von Primzahlen mit der Endziffer EZ9 im Umfeld der Zahl $n_g = 2^{\wedge}2000$:

- ```
  for i from 0 to 1000 do: if isprime (2^2000+3+10*i)
  then print ("i", i, 3+10*i, " = ", 2^2000+3+10*i) end_if: end_for:
  ```

"i", 453, 4533, " = ",

```
11481306952742545242328332011776819840223177020886952004776427368257766261392370\
3138566594863165062699184459646389874627734471189608630553314259313561666531853\
```

```
912998914531228000068877914824004487142892699006348624478161546364638836394731 7\
026040466353970904996558162398808944629605623311649536164221970332681344168908 9\
844585056023794848079140589009347765004290027167066258305220081322362812917612 6\
788331720659899539641812702177985840404215985318325154088943390209192055495778 3\
589672039160081957216630582755380425583726015528348786419432054508915275783882 6\
25175435528800822842770817965453762184851149033909
```

```
"i", 597, 5973, " = ",
```

(1440)

```
114813069527425452423283320117768198402231770208869520047764273682576626139237 0\
313856659486316506269918445964638987462773447118960863055331425931356166653185 3\
912998914531228000068877914824004487142892699006348624478161546364638836394731 7\
026040466353970904996558162398808944629605623311649536164221970332681344168908 9\
844585056023794848079140589009347765004290027167066258305220081322362812917612 6\
788331720659899539641812702177985840404215985318325154088943390209192055495778 3\
589672039160081957216630582755380425583726015528348786419432054508915275783882 6\
25175435528800822842770817965453762184851149035349
```

<u>Anmerkung:</u>

In diesem Intervall sind 2 Primzahlen mit der Endziffer EZ9 situiert, die einen Zwilling mod 1440 bilden.

Für den Nachweis der Effizienz des "modifizierten - horizontalen - Testverfahrens" **werden im Beispiel 6. 5** die Primzahlen mit der Endziffer EZ3 bestimmt, die im Intervall: 1 +10*i für i := 0 bis 1000 im Umfeld der Zahl n_g = 2^5000 situiert sind. In den Beispielen: 6. 6, 6.7, 6. 8 und 6. 9 werden systematisch die Primzahlen mit den Endziffern: EZ1, EZ3, EZ7 und EZ9 im Umfeld der Zahl n_g = 2^1000 bestimmt.

6. 5 Die Bestimmung der Primzahlen mit der Endziffer EZ3 im Umfeld der Zahl n_g = 2^5000 im Intervall: 2^5000 + 7+10*i für i := 0 bis 1000:

- ```
      for i from 0 to 1000 do: if isprime (2^5000+7+10*i)
  then print ("i",i, 7+10*i, " = ", 2^5000 +7 +10*i) end_if: end_for:
  ```

```
"i", 215, 2157, " = ",

141246703213942603683520966701614733366889617518454111681368808585711816984270 7\
512558089126316711526373356032084313660827642038380699793383359711857266399234 3\
105177785186539901187799964513170706937349821263132375255311121537284403595090 0\
535954860733418453405575566736801565587405464699640499050849699472357900905617 5\
713766182282164342131815209915566771264986517822041740618309392391768613413832 9\
401824022583869272559614700514424328107527562949533909381319896673563360632969 1\
023842454125835888656873133981287240980008838073668221804264432910894030789020 2\
194405781984882673397682388722799021574203072475705104238458688725967358918058 1\
872779643575301851808664135601285130254672682300925021832801825190734024544986 3\
183265637987862198511046362985461949587281119139907228004385942880953958816554 5\
676252960869168857748289344499413624165886753269403325611036645569826222068344 7\
421981108187240492950348199137674037982599879141187980271758388549857511529947 1\
743469241117070230398103378615232793710290992656444842895511830355733152020804 1\
579200900418119518804567055154683494461827317423276859892776076207095258783187 6\
648836834896501547499786411976544143335692801234411176573533639355787921493700 4\
347568208665958717764059293592887514292843557047089164876483116615691886203812 9\
975556901718921697337552244690324750787978309013215799401273372106943772834399 2\
228027406079823478674043489345812019834110103381250672004660989116070028400210 0\
980452964039788704335302619337597862052192280371481132164147186514169090917191 9\
11533

"i", 314, 3147, " = ",
```

(990)

```
141246703213942603683520966701614733366889617518454111681368808585711816984270 7\
512558089126316711526373356032084313660827642038380699793383359711857266399234 3\
105177785186539901187799964513170706937349821263132375255311121537284403595090 0\
535954860733418453405575566736801565587405464699640499050849699472357900905617 5\
713766182282164342131815209915566771264986517822041740618309392391768613413832 9\
401824022583869272559614700514424328107527562949533909381319896673563360632969 1\
023842454125835888656873133981287240980008838073668221804264432910894030789020 2\
194405781984882673397682388722799021574203072475705104238458688725967358918058 1\
872779643575301851808664135601285130254672682300925021832801825190734024544986 3\
183265637987862198511046362985461949587281119139907228004385942880953958816554 5\
676252960869168857748289344499413624165886753269403325611036645569826222068344 7\
421981108187240492950348199137674037982599879141187980271758388549857511529947 1\
743469241117070230398103378615232793710290992656444842895511830355733152020804 1\
579200900418119518804567055154683494461827317423276859892776076207095258783187 6\
648836834896501547499786411976544143335692801234411176573533639355787921493700 4\
347568208665958717764059293592887514292843557047089164876483116615691886203812 9\
975556901718921697337552244690324750787978309013215799401273372106943772834399 2\
228027406079823478674043489345812019834110103381250672004660989116070028400210 0\
```
```

```
9804529640397887043353026193375978620521922803714811321641471865141690909171919\
12523

"i", 863, 8637, " = ",                                                        (5490)

1412467032139426036835209667016147333668896175184541116813688085857118169842707\
5125580891263167115263733560320843136608276420383806997933833597118572663992343\
1051777851865399011877999645131707069373498212631323752553111215372844035950900\
5359548607334184534055755667368015655874054646996404990508496994723579009056175\
7137661822821643421318152099155667712649865178220417406183093923917686134138329\
4018240225838692725596147005144243281075275629495339093813198966735633606329691\
0238424541258358886568731339812872409800088380736682218042644329108940307890202\
1944057819848826733976823887227990215742030724757051042384586887259673589180581\
8727796435753018518086641356012851302546726823009250218328018251907340245449863\
1832656379878621985110463629854619495872811191399072280043859428809539588165545\
6762529608691688577482893444994136241658867532694033256110366455698262220683447\
4219811081872404929503481991376740379825998791411879802717583885498575115299471\
7434692411170702303981033786152327937102909926564448428955118303557331520208041\
5792009004181195188045670551546834944618273174232768598927760762070952587831876\
6488368348965015474997864119765441433356928012344111765735336393557879214937004\
3475682086659587177640592935928875142928435570470891648764831166156918862038129\
9755569017189216973375522446903247507879783090132157994012733721069437728343992\
2280274060798234786740434893458120198341101033812506720046609891160700284002100\
9804529640397887043353026193375978620521922803714811321641471865141690909171919\
18013
```

Anmerkung:

1. In diesem Intervall sind drei 1506-stellige Primzahlen mit der Endziffer EZ3 situiert.
2. Diese Primzahlen sind größer als die bis zum Jahr 1961 bestimmte Primzahl $M_{p,20}$.

---

6. 6 Die Bestimmung der Primzahlen mit der Endziffer EZ1 im Umfeld der Zahl n_g = 2^1000 im Intervall 2^1000 +5+10*i für i:= 0 bis 2000:

- ```
  for i from 0 to 2000 do: if isprime (2^1000+5+10*i)
  then print ("i", i, 5+10*i, " = ", 2^1000+5+10*i) end_if: end_for:
  ```

```
"i", 647, 6475, " = ",

1071508607186267320948425049060001810561404811705533607443750388370351051124936\
1224931983788156958581275946729175531468251871452856923140435984577574698574803\
9345677748242309854210746050623711418779541821530464749835819412673987675591655\
439460770629145711964776865421676604298316526243868372056680758511

"i", 1363, 13635, " = ",                                                      (7160)

1071508607186267320948425049060001810561404811705533607443750388370351051124936\
1224931983788156958581275946729175531468251871452856923140435984577574698574803\
9345677748242309854210746050623711418779541821530464749835819412673987675591655\
43946077062914571196477686542167660429831652624386837205668083011

"i", 1600, 16005, " = ",                                                      (2370)

1071508607186267320948425049060001810561404811705533607443750388370351051124936\
1224931983788156958581275946729175531468251871452856923140435984577574698574803\
9345677748242309854210746050623711418779541821530464749835819412673987675591655\
43946077062914571196477686542167660429831652624386837205668085381
```

Anmerkung: In diesem Intervall sind drei 302-stellige Primzahlen mit der Endziffer EZ1 situiert.

---

6. 7 Die Bestimmung der Primzahlen mit der Endziffer EZ3 im Umfeld der Zahl n_g = 2^1000 im Intervall 2^1000 +7+10*i für i:= 0 bis 2000:

- ```
  for i from 0 to 2000 do: if isprime (2^1000+7+10*i)
  then print ("i", i, 7+10*i, " = ", 2^1000+7+10*i) end_if: end_for:
  ```

```
"i", 29, 297, " = ",

1071508607186267320948425049060001810561404811705533607443750388370351051124936\
1224931983788156958581275946729175531468251871452856923140435984577574698574803\
9345677748242309854210746050623711418779541821530464749835819412673987675591655\
4394607706291457119647768654216766042983165262438683720566806967 3

"i", 957, 9577, " = ",                                                        (9280)

1071508607186267320948425049060001810561404811705533607443750388370351051124936\
```

```
1224931983788156958581275946729175531468251871452856923140435984577574698574803\
9345677748242309854210746050623711418779541821530464749835819412673987675591655\
4394607706291457119647768654216766042983165262438683720566807895
```

"i", 1062, 10627, " = ", (1050)

```
1071508607186267320948425049060001810561404811705533607443750388370351051124936\
1224931983788156958581275946729175531468251871452856923140435984577574698574803\
9345677748242309854210746050623711418779541821530464749835819412673987675591655\
4394607706291457119647768654216766042983165262438683720566808000
```

"i", 1083, 10837, " = ", (200)

```
1071508607186267320948425049060001810561404811705533607443750388370351051124936\
1224931983788156958581275946729175531468251871452856923140435984577574698574803\
9345677748242309854210746050623711418779541821530464749835819412673987675591655\
4394607706291457119647768654216766042983165262438683720566808021
```

<u>Anmerkung:</u> In diesem Intervall sind vier Primzahlen mit der Endziffer EZ3 situiert.

---

6. 8 Die Bestimmung der Primzahlen mit der Endziffer EZ7 im Umfeld der Zahl $n_g = 2^{\wedge}1000$ im Intervall 2^1000+1+10*i für i:= 0 bis 2000:

- ```
  for i from 0 to 2000 do: if isprime (2^1000+1+10*i)
  then print ("i", i, 1+10*i, " = ", 2^1000+1+10*i) end_if: end_for:
  ```

"i", 408, 4081, " = ",

```
1071508607186267320948425049060001810561404811705533607443750388370351051124936\
1224931983788156958581275946729175531468251871452856923140435984577574698574803\
9345677748242309854210746050623711418779541821530464749835819412673987675591655\
4394607706291457119647768654216766042983165262438683720566807345
```

"i", 1022, 10221, " = ", (6140)

```
1071508607186267320948425049060001810561404811705533607443750388370351051124936\
1224931983788156958581275946729175531468251871452856923140435984577574698574803\
9345677748242309854210746050623711418779541821530464749835819412673987675591655\
4394607706291457119647768654216766042983165262438683720566807959
```

"i", 1356, 13561, " = ", (3340)

```
1071508607186267320948425049060001810561404811705533607443750388370351051124936\
1224931983788156958581275946729175531468251871452856923140435984577574698574803\
9345677748242309854210746050623711418779541821530464749835819412673987675591655\
4394607706291457119647768654216766042983165262438683720566808293
```

"i", 1592, 15921, " = ", (2360)

```
1071508607186267320948425049060001810561404811705533607443750388370351051124936\
1224931983788156958581275946729175531468251871452856923140435984577574698574803\
9345677748242309854210746050623711418779541821530464749835819412673987675591655\
4394607706291457119647768654216766042983165262438683720566808529
```

"i", 1782, 17821, " = ", (1900)

```
1071508607186267320948425049060001810561404811705533607443750388370351051124936\
1224931983788156958581275946729175531468251871452856923140435984577574698574803\
9345677748242309854210746050623711418779541821530464749835819412673987675591655\
4394607706291457119647768654216766042983165262438683720566808719
```

"i", 1827, 18271, " = ", (4509

```
1071508607186267320948425049060001810561404811705533607443750388370351051124936\
1224931983788156958581275946729175531468251871452856923140435984577574698574803\
9345677748242309854210746050623711418779541821530464749835819412673987675591655\
4394607706291457119647768654216766042983165262438683720566808764
```

"i", 1913, 19131, " = ", (860)

```
1071508607186267320948425049060001810561404811705533607443750388370351051124936\
1224931983788156958581275946729175531468251871452856923140435984577574698574803\
9345677748242309854210746050623711418779541821530464749835819412673987675591655\
4394607706291457119647768654216766042983165262438683720566808850
```

"i", 1991, 19911, " = ", (780)

```
1071508607186267320948425049060001810561404811705533607443750388370351051124936\
1224931983788156958581275946729175531468251871452856923140435984577574698574803\
```
```

```
9345677748242309854210746050623711418779541821530464749835819412673987675591655\
43946077062914571196477686542167660429831652624386837205668089287
```

Anmerkung: In diesem Intervall sind 8 Primzahlen mit der Endziffer EZ7 situiert.

---

6. 9 Die Bestimmung der Primzahlen mit der Endziffer EZ9 im Umfeld der Zahl $n_g = 2$^1000 im Intervall 2^1000+3+10*i, für i:= 0 bis 2000:

- ```
  for i from 0 to 2000 do: if isprime (2^1000+3+10*i)
  then print ("i", i, 3+10*i, " = ", 2^1000+3+10*i) end_if: end_for:
  ```

```
"i", 534, 5343, " = ",

1071508607186267320948425049060001810561404811705533607443750388370351051124936\
1224931983788156958581275946729175531468251871452856923140435984577574698574803\
9345677748242309854210746050623711418779541821530464749835819412673987675591655\
43946077062914571196477686542167660429831652624386837205668074719
```

"i", 679, 6793, " = ", (1450)

```
1071508607186267320948425049060001810561404811705533607443750388370351051124936\
1224931983788156958581275946729175531468251871452856923140435984577574698574803\
9345677748242309854210746050623711418779541821530464749835819412673987675591655\
43946077062914571196477686542167660429831652624386837205668076169
```

"i", 1087, 10873, " = ", (4080)

```
1071508607186267320948425049060001810561404811705533607443750388370351051124936\
1224931983788156958581275946729175531468251871452856923140435984577574698574803\
9345677748242309854210746050623711418779541821530464749835819412673987675591655\
43946077062914571196477686542167660429831652624386837205668080249
```

"i", 1540, 15403, " = ", (4530)

```
1071508607186267320948425049060001810561404811705533607443750388370351051124936\
1224931983788156958581275946729175531468251871452856923140435984577574698574803\
9345677748242309854210746050623711418779541821530464749835819412673987675591655\
43946077062914571196477686542167660429831652624386837205668084779
```

Anmerkung:

1. In diesem Intervall sind 4 Primzahlen mit der Endziffer EZ9 situiert.
2. In diesem Intervall sind daher im Umfeld der Zahl $n_g = 2^{4+249 \times 4} = 2^{1000}$ insgesamt 19 Primzahlen situiert.
3. Die Zahl $n_g = 2^{1000}$ ist ein Element der *Folge 4* und für die Zahlen: $n_g = 2^n$ der *Folge 1*, *Folge 2* und der *Folge 3* können mit der "horizontal-selektiven Testmethode" in analoger Weise, die Primzahlen mit den vier Endziffern: EZ1, EZ3, EZ7 und EZ9 bestimmt werden, die im Umfeld dieser Zahlen in definierten Intervallen situiert sind.

---

6. 10 Die Bestimmung der Primzahlen mit der Endziffer EZ9, im Umfeld der Zahl $n_g = 2$^9691 im Intervall 2^9691 +1 +10*i für i:= 0 bis 1000:

- ```
  for i from 0 to 1000 do: if isprime (2^9691+1+10*i)
  then print ("i", 1+10*i, " = ", 2^9691+1+10*i) end_if: end_for:
  ```

```
"i", 6461, " = ",

1912881115221844811811357194640023638965988689608946003405338043967351580377065\
1881115710744450831578347298883926097025277226340210707336349923337717733059189\
7035729904947533240865328358193894192231997733652393039560549752247484383241725\
9472325511328612198686240625313049765616159257283048814508876163436316214881390\
1024422256286317055471500963942293426090624434168514309284231516209315460141423\
4944627174546235597028462976806153283670096913721876752457096026485422372140674\
1481590725074554300150491115179832232832732504690280401399282604931949070893395\
8043813877502732148153495998787086343155622556462090453621982828738096632876832\
8950657682360583733226280388886158495074136950819445543123591024852046712451966\
6562944698642679123274703186765868498429485161680235363569274555095155070117154\
7815188267923869624214120491414156147861960899139698596031819594714013258622185\
9020180067447494194678574982104564825631317984898081111558485544106637325907507\
8528835801711516649860607724931132702621511750877809869358327712594667430553607\
8362880717653398331881277575537555241978184161368350146254513328608292726457202\
```

```
88707748570495407016208338085866125320473420785036739823575399610139512150686 59\
08295722537747360337873023749773806761263997396550658555875890456796061219626 19\
12762249086848997882829588652038151029737652316820074541289378661825827830973 27\
54017991300059465386948852156839957940858076399293151508994660205204326707961 15\
70074877000267788862826144864994784962277027462217184886208846241711114650183 74\
79952042807446504059058971738393207346050934538509307035322428056414370193165 99\
09735718869472306244383839964523675915295340052654269064574387687295990878773 54\
97581615986494702322112844828545585483432224204643127083941810307952129335251 75\
70910084077785181099612553422079407883837155409456612071568747005640581648123 84\
76508373747615808839312127067576824530228939857455361222594939716353689514517 15\
49888924876129541124136367297226475790962909062097139573217899445819768307196 16\
69577886633531266856417433248271620076581093051494813598988288240675302513096 17\
08068129042691192137391730570292036735925231087728982157905584242635285730641 26\
24565962962230792220665052177790335026846065960724773768947437696607375181557 34\
30617285197762194213806234370936982473005872548588824251511240849636948320859 69\
19339854071810908121185188062771074604654020032162913069700945057878284307907 54\
64549211157254438697530876069542033938653493904483374271932260238232291740442 55\
01769632294026683319489351190754997559343380923825559844824652745375678684483 45\
85753859788385292029091680365034458907199904998683941107801414293569766481063 08\
82964994734368811315933244563335693209735671191907961245703374477403747017935 24\
47776352510535552336817807219649817535536723203781102506672230515819053538565 220\
43101509298832998321813342286992786010028327892186569130442279004225715919580 47\
287715439054089162155957949504585685596436461434580232759707873049030229 09
```

<u>Anmerkungen:</u>

1. In diesem Intervall ist eine 2918 -stellige Primzahl, EZ9 situiert. Der Test wird anschließend wiederholt.

2. Diese Primzahl ist größer als die 2917-stellige Mersenne - Primzahl $M_{p,22}$, EZ1, die im Jahr 1963 von Gillies gefunden wurde.

3. Für die Bestimmung der Primzahl $M_{p,22}$ musste ein Test an 3704 Zahlen vom Typ 2^p -1 vorgenommen werden!

4. Für die Bestimmung der Primzahl p = 2^9691 - 6461, EZ9 mussten hingegen nur 646 Testungen mit dem MuPAD - Testprogramm (mit einem Zeitaufwand von 10 Stunden) durchgeführt werden.

5. Dieser Vergleich zeigt den Vorteil der " horizontal - selektiven Testmethode" bei der Suche nach sehr großen Primzahlen gegenüber dem bisherigen GIMPs - Testverfahren.

6. In noch höheren Primzahl-Bereichen p > 25.964.951 sind nach dem GIMPs - Verfahren mehrere Millionen Testungen vom Typ 2^p -1 erforderlich, um noch größere Primzahlen $M_{p,43}$, $M_{p,44}$ usw. zu finden.

7. Mit dem GIMPs - Verfahren können des weiteren keine sehr großen Primzahlen mit den Endziffern EZ3 und EZ9 bestimmt werden.

6.10 Der wiederholte Test für die gefundene Primzahl p = 2^9691 + 6461:

- ```
  for i from 6461 to 6461 do: if isprime (2^9691+i)
  then print ("i", i, " = ", 2^9691+i) end_if: end_for:
  ```

```
"i", 6461, " = ",

19128811152218448118113571946400236389659886896089460034053380439673515803770 65\
18811157107444508315783472988839260970252772263402107073363499233377177330591 89\
70357299049475332408653283581938941922319977336523930395605497522474843832417 25\
94723255113286121986862406253130497656161592572830488145088761634363162148813 90\
10244222562863170554715009639422934260906244341685143092842315162093154601414 23\
49446271745462355970284629768061532836700969137218767524570960264854223721406 74\
14815907250745543001504911151798322328327325046902804013992826049319490708933 95\
80438138775027321481534959987870863431556225564620904536219828287380966328768 32\
89506576823605837332262803888861584950741369508194455431235910248520467124519 66\
65629446986426791232747031867658684984294851616802353635692745550951550701171 54\
78151882679238696242141204914141561478619608991396985960318195947140132586221 85\
90201800674474941946785749821045648256313179848980811115584855441066373259075 07\
85288358017115166498606077249311327026215117508778098693583277125946674305536 07\
83628807176533983318812775755375552419781841613683501462545133286082927264572 02\
88707748570495407016208338085866125320473420785036739823575399610139512150686 59\
08295722537747360337873023749773806761263997396550658555875890456796061219626 19\
12762249086848997882829588652038151029737652316820074541289378661825827830973 27\
54017991300059465386948852156839957940858076399293151508994660205204326707961 15\
```
```

```
70074877000267788862826144864994784962277027462217184886208846241711114650183 74\
79952042807446504059058971738393207346050934538509307035322428056414370193165 99\
09735718869472306244383839964523675915295340052654269064574387687295990878773 54\
97581615986494702322112844828545585483432224204643127083941810307952129335251 75\
70910084077785181099612553422079407883837155409456612071568747005640581648123 84\
76508373747615808839312127067576824530228939857455361222594939716353689514517 15\
49888924876129541124136367297226475790962909062097139573217899445819768307196 16\
69577886633531266856417433248271620076581093051494813598988288240675302513096 17\
08068129042691192137391730570292036735925231087728982157905584242635285730641 26\
24565962962230792220665052177790335026846065960724773768947437696607375181557 34\
30617285197762194213806234370936982473005872548588824251511240849636948320859 69\
19339854071810908121185188062771074604654020032162913069700945057878284307907 54\
64549211157254438697530876069542033938653493904483374271932260238232291740442 55\
01769632294026683319489351190754997559343380923825559844824652745375678684483 45\
85753859788385292029091680365034458907199904998683941107801414293569766481063 08\
82964994734368811315933244563335693209735671191907961245703374477403747017935 24\
47776352510535523368178072196498175355367232037811025066722305158190535385652 20\
43101509298832998321813342286992786010028327892186569130442279004225715919580 47\
2877154390540891621559579495045856855964364614345802327597078730490302290 9.
```

<u>Anmerkung:</u> Die Primzahl p = 2^9691+6461, EZ9 wurde durch einen wiederholten Test bestätigt.

---

6. 11 Die Bestimmung der Primzahlen mit den dezidierten Endziffern: EZ1, EZ3, EZ7, EZ9 in der Umgebung der Zahl $n_g = 41!$ in den Intervallen: i:= 0 bis 431::

```
for i from 0 to 431 do: if isprime (41!+1+10*i)
then print ("i", i,1+10*i, " = ", 41!+1+10*i) end_if: end_for:

"i", 0, 1, " = ", 33452526613163807108170062053440751665152000000001

"i", 6, 61, " = ", 33452526613163807108170062053440751665152000000061

"i", 21, 211, " = ", 33452526613163807108170062053440751665152000000211

"i", 57, 571, " = ", 33452526613163807108170062053440751665152000000571

"i", 216, 2161, " = ", 33452526613163807108170062053440751665152000002161

"i", 274, 2741, " = ", 33452526613163807108170062053440751665152000002741

"i", 327, 3271, " = ", 33452526613163807108170062053440751665152000003271

"i", 349, 3491, " = ", 33452526613163807108170062053440751665152000003491

"i", 420, 4201, " = ", 33452526613163807108170062053440751665152000004201
```

```
for i from 0 to 431 do: if isprime (41!+3+10*i)
then print ("i", i, 3+10*i, " = ", 41!+3+10*i) end_if: end_for:

"i", 28, 283, " = ", 33452526613163807108170062053440751665152000000283

"i", 37, 373, " = ", 33452526613163807108170062053440751665152000000373

"i", 67, 673, " = ", 33452526613163807108170062053440751665152000000673

"i", 85, 853, " = ", 33452526613163807108170062053440751665152000000853

"i", 166, 1663, " = ", 33452526613163807108170062053440751665152000001663

"i", 178, 1783, " = ", 33452526613163807108170062053440751665152000001783

"i", 193, 1933, " = ", 33452526613163807108170062053440751665152000001933

"i", 199, 1993, " = ", 33452526613163807108170062053440751665152000001993

"i", 238, 2383, " = ", 33452526613163807108170062053440751665152000002383

"i", 268, 2683, " = ", 33452526613163807108170062053440751665152000002683

"i", 269, 2693, " = ", 33452526613163807108170062053440751665152000002693

"i", 302, 3023, " = ", 33452526613163807108170062053440751665152000003023

"i", 373, 3733, " = ", 33452526613163807108170062053440751665152000003733

"i", 383, 3833, " = ", 3345252661316380710817006205344075166515200000
```

```
for i from 0 to 431 do: if isprime (41!+7+10*i)
then print ("i", i, 7+10*i, " = ", 41!+7+10*i) end_if: end_for:

"i", 30, 307, " = ", 33452526613163807108170062053440751665152000000307
"i", 31, 317, " = ", 33452526613163807108170062053440751665152000000317
"i", 123, 1237, " = ", 33452526613163807108170062053440751665152000001237
"i", 202, 2027, " = ", 33452526613163807108170062053440751665152000002027
"i", 213, 2137, " = ", 33452526613163807108170062053440751665152000002137
"i", 277, 2777, " = ", 33452526613163807108170062053440751665152000002777
"i", 291, 2917, " = ", 33452526613163807108170062053440751665152000002917
"i", 351, 3517, " = ", 33452526613163807108170062053440751665152000003517
"i", 361, 3617, " = ", 33452526613163807108170062053440751665152000003617
"i", 390, 3907, " = ", 33452526613163807108170062053440751665152000003907
```

```
for i from 0 to 431 do: if isprime (41!+9+10*i)
then print ("i", i, 9+10*i, " = ", 41!+9+10*i) end_if: end_for:

"i", 22, 229, " = ", 33452526613163807108170062053440751665152000000229
"i", 76, 769, " = ", 33452526613163807108170062053440751665152000000769
"i", 148, 1489, " = ", 33452526613163807108170062053440751665152000001489
"i", 199, 1999, " = ", 33452526613163807108170062053440751665152000001999
"i", 206, 2069, " = ", 33452526613163807108170062053440751665152000002069
"i", 217, 2179, " = ", 33452526613163807108170062053440751665152000002179
"i", 220, 2209, " = ", 33452526613163807108170062053440751665152000002209
"i", 245, 2459, " = ", 33452526613163807108170062053440751665152000002459
"i", 269, 2699, " = ", 33452526613163807108170062053440751665152000002699
"i", 346, 3469, " = ", 33452526613163807108170062053440751665152000003469
"i", 377, 3779, " = ", 33452526613163807108170062053440751665152000003779
"i", 413, 4139, " = ", 33452526613163807108170062053440751665152000004139
```

Anmerkungen:

1. In dem gleichen (horizontalen) Intervall i:= 0 bis 431 sind 9 Primzahlen, EZ1, 14 Primzahlen, EZ3, 10 Primzahlen, EZ7 und 12 Primzahlen, EZ9 situiert.

2. Alle diese (insgesamt) 45 Primzahlen treten in Primzahl-Abständen gegenüber der Zahl $n_g = 41!$ auf.

6.12 Die Bestimmung der Primzahlen mit dezidierten Endziffern: EZ1, EZ3, EZ7, EZ9 in der Umgebung der Zahl $n_g = 41!^3$ im Intervall i:= 0 bis 431:

```
for i from 0 to 431 do: if isprime (41!^3+1+10*i)
then print ("i", i, 1+10*i, " = ", 41!^3+1+10*i) end_if: end_for:

"i", 174, 1741, " = ",

37435770366984408256341512850548309818411283674720078098202042770914088100326 25\
458714656326740390616910788141413900799180800000000000000000000000001741

"i", 382, 3821, " = ",

3743577036698440825634151285054830981841128367472007809820204277091408810032625\
458714656326740390616910788141413900799180800000000000000000000000003821

for i from 0 to 431 do: if isprime (41!^3+3+10*i)
```

```
then print ("i", i, 3+10*i, " = ", 41!^3+3+10*i) end_if: end_for:

"i", 238, 2383, " = ",

3743577036698440825634151285054830981841128367472007809820204277091408810032625\
458714656326740390616910788141413900799180800000000000000000000000002383

"i", 242, 2423, " = ",

3743577036698440825634151285054830981841128367472007809820204277091408810032625\
458714656326740390616910788141413900799180800000000000000000000000002423

"i", 428, 4283, " = ",

3743577036698440825634151285054830981841128367472007809820204277091408810032625\
458714656326740390616910788141413900799180800000000000000000000000004283

    for i from 0 to 431 do: if isprime (41!^3+7+10*i)
  then print ("i", i, 7+10*i, " = ", 41!^3+7+10*i) end_if: end_for:

"i", 54, 547, " = ",

3743577036698440825634151285054830981841128367472007809820204277091408810032625\
4587146563267403906169107881414139007991808000000000000000000000000000547

"i", 166, 1667, " = ",

3743577036698440825634151285054830981841128367472007809820204277091408810032625\
458714656326740390616910788141413900799180800000000000000000000000001667

"i", 318, 3187, " = ",

3743577036698440825634151285054830981841128367472007809820204277091408810032625\
458714656326740390616910788141413900799180800000000000000000000000003187

    for i from 0 to 431 do: if isprime (41!^3+9+10*i)
  then print ("i", i, 9+10*i, " = ", 41!^3+9+10*i) end_if: end_for:

"i", 41, 419, " = ",

3743577036698440825634151285054830981841128367472007809820204277091408810032625\
4587146563267403906169107881414139007991808000000000000000000000000000419

"i", 371, 3719, " = ",

3743577036698440825634151285054830981841128367472007809820204277091408810032625\
458714656326740390616910788141413900799180800000000000000000000000003719

"i", 391, 3919, " = ",

3743577036698440825634151285054830981841128367472007809820204277091408810032625\
458714656326740390616910788141413900799180800000000000000000000000003919
```

Anmerkungen:

- In dem gleichen (horizontalem) Intervall i:= 0 bis 431 sind 2 Primzahlen, EZ1, 3 Primzahlen, EZ3, 3 Primzahlen, EZ7 und 3 Primzahlen, EZ9 situiert.
- Insgesamt sind in diesem (gleichen) Intervall (wie in Beispiel **6.11**) 11 Primzahlen situiert.
- Wie aus diesem Vergleich ersichtlich, ist für die Abnahme der Primzahl-Dichte das Asymptote-Gesetz ursächlich.

---

13 Die Bestimmung der "Dichten", in denen Zwillinge mod 2 im gleichen (jedoch verschobenen) Intervall i:= 0 bis 20000 in den Umgebungen der Zahlen: $n_g = 41!$ und $n_g = 41!^3$ auftreten:

```
  for i from 0 to 20000 do: if (isprime (41!+i) and isprime (41!+i+2))
  then print (("i", i, 41!+i," = ", 41!+i+2)) end_if: end_for:

i", 6131, 33452526613163807108170062053440751665152000006131, " = ",   p = 6131

  33452526613163807108170062053440751665152000006133                 p = 6133

i", 9929, 33452526613163807108170062053440751665152000009929, " = ",   p = 9929
```

```
33452526613163807108170062053440751665152000009931                p = 9931
"i", 14429, 33452526613163807108170062053440751665152000014429, " = ",   14429 = 47.307
   33452526613163807108170062053440751665152000014431              p = 14431
"i", 17837, 33452526613163807108170062053440751665152000017837, " = ",   p = 17837
   33452526613163807108170062053440751665152000017839              p = 17839

     for i from 0 to 20000 do: if (isprime (41!^3+i) and isprime (41!^3+i+2))
     then print (("i", i, 41!^3+i, " = ", 41!^3+i+2)) end_if: end_for:
"i", 14009,

3743577036698440825634151285054830981841128367472007809820204277091408810032625\
45871465632674039061691078814141390079918080000000000000000000000014009      p = 14009

   , " = ",

3743577036698440825634151285054830981841128367472007809820204277091408810032625\
45871465632674039061691078814141390079918080000000000000000000000014011      p = 14011
```

Anmerkungen:

1. Im gleichen Intervall i:= 0 bis 200000 sind in der Umgebung der Zahl $n_g = 41!$ insgesamt 4 Zwillinge mod 2 situiert, dagegen aber nur 1 Zwilling mod 2 in der Umgebung der Zahl $n_g = 41!$^3

2. Dieser Sachverhalt bestätigt das Asymptote-Gesetz.

3. Aus diesem Beispiel ergibt sich die Bestätigung von Gesetz 2, da alle Abstände, in denen die Primzahlen der Zwillinge mod 2 gegenüber den Zahlen $n_g = 41!$ bzw. $n_g = 41!$^3 auftreten, Primzahlen $p > 41$ sind, mit Ausnahme des Abstandes $\Delta = 14429 = 47 \cdot 307$, der eine ungerade Zahl ist, die aus zwei Primzahlen $p > 41$ besteht.

4. Beachtlich ist, dass die Zwillinge mod 2:
(41! +6131, 41! +6133), (41! +9929, 41! +9931) aus einer "Verschiebung" der Zwillinge (6131, 6133) und (9929, 9931) um die Zahl +41! entstanden sind, wogegen der Zwilling mod 2 (41! +14429, 41! +14431) im verschobenen Zahlenbereich "neu kreiert" wurde.
Beachtlich ist auch, dass der Zwilling mod 2: (41!^3 +14009, 41!^3 +14011) aus dem Zwilling mod 2: (14009, 14011) durch die Verschiebung um die Zahl $n_g = 41!$^3 entstanden ist.

Die Schlussfolgerungen:

1. Aus den in der Anlage 12 angestellten Untersuchungen (Analysen) geht hervor, dass die "horizontale Bestimmung von Primzahlen mit definierten Endziffern" in großen Zahlen-Bereichen eine optimale Methode darstellt.

2. Das hier eingeführte Verfahren ist dafür geeignet mit einer geringeren Zahl von parallel arbeitenden Rechnern fortlaufend größere Primzahlen mit definierten Endziffern zu finden.

3. Die hier eingeführte (und begründete) Verfahren kann als eine "horizontale Sieb-Methode " bezeichnet werden, die effektiver ist als die GIMPs - Methode, die als eine "vertikale Sieb-Methode" bezeichnet werden kann.

4. Aus den Schlussfolgerungen: 1 bis 3 ergibt sich die Empfehlung die GIMPs - Methode nach dem Prinzip der "horizontalen Sieb-Methode" neu zu entwickeln.

Anlage 13

Die Ursachen für das Asymptote-Gesetz und die Schwankungen der Primzahl-Dichten, die durch Gesetz 1 und Gesetz 2 der Faktor-Zerlegung von Zahlen $n \in N$ bedingt sind:

An den folgenden Bespielen, die mit dem MuPAD-Test bestimmt wurden, können die tieferen Ursachen für das Asymptote-Gesetz und die Schwankungen der Primzahl-Dichten in steigenden Zahlen-Intervallen wahrgenommen werden:

Beispiel 1: Die Bestimmung der Primzahlen (p_1 und p_2) im direkten Umfeld der Zahl $n_g = 2 \cdot 3 \cdot .. \cdot 53$:

```
for i from -73 to +59 do: if isprime (32589158477190044730+i)
then print ("i", i, " = ", 32589158477190044730+i) end_if: end_for:
```

```
"i", -73, " = ", 32589158477190044657          p1
                                                      (132)
"i", 59, " = ", 32589158477190044789           p2
```

Beispiel 2: Die Bestimmung der Primzahlen (p_1 und p_2) im direkten Umfeld der Zahl $n_g = 53!$:

```
for i from -59 to +71 do: if isprime (53! +i)
then print ("i", i, " = ", 53!+i) end_if: end_for:
```

```
"i", -59, " = ",
4274883284060025564298013753389399649690343788366813724671999999999941     p1
"i", 71, " = ",                                                            (130)
4274883284060025564298013753389399649690343788366813724672000000000071     p2
```

Beispiel 3: Die Bestimmung der Primzahlen (p_1 und p_2) im direkten Umfeld der Zahl $n_g = 53!\text{^}2$:

```
for i from -557 to + 251 do: if isprime (53!^2+i)
then print ("i", i, " = ", 53!^2+i) end_if: end_for:
```

```
"i", -557, " = ",
18274627092335829218864095445902180793856136322602419837579292998750581286734l3\
246943996297675146738367974582150758399999999999999999999999443     p1
"i", 181, " = ",                                                    (738)
1827462709233582921886409544590218079385613632260241983757929299875058128673413\
246943996297675146738367974582150758400000000000000000000000181     p2
```

Beispiel 3: Die Bestimmung der Primzahlen (p_1 und p_2) im direkten Umfeld der Zahl $n_g = 53!\text{^}3$:

```
for i from -541 to +1051 do: if isprime (53!^3+i)
then print ("i", i, " = ", 53!^3+i) end_if: end_for:
```

```
"i", -541, " = ",
781218978794569056455541275273163671331495422239465647156444725682199526902813 8\
126821455504085303328474170354776484913408502845533105053870293326358382387928 1\
6173733359124479999999999999999999999999999999999459     p1
"i", 1013, " = ",                                        (1554)
7812189787945690564555412752731636713314954222394656471564447256821995269028138\
1268214555040853033284741703547764849134085028455331050538702933263583823879281\
6173733359124480000000000000000000000000000000001013     p2
```

<u>Beispiel 4:</u> Die Bestimmung der Primzahlen (p_1 und p_2) im direkten Umfeld der Zahl $n_g = 53!^\wedge 7$:

```
    for i from -103 to +199 do: if isprime (53!^7+i)
  then print ("i", i, " = ", 53!^7+i) end_if: end_for:

"i", -103, " = ",

2608974489744096593999963259584288866327809535386060473416779419330070525208163\
7398733067921242612715103263315595387916184131642834396662041795725639918936316\
1791093853850215025471660274485005445141067926322404589720910423761396844606877\
6833882937036766653549194163002057800223606773706021049182780816950510262080420\
0839923496025503745439312268998233372140131455641689271449829210296345158759516\
4587130879999999999999999999999999999999999999999999999999999999999999999999999\
99999999999897                                                          p1
                                                                        (302)
"i", 199, " = ",

2608974489744096593999963259584288866327809535386060473416779419330070525208163\
7398733067921242612715103263315595387916184131642834396662041795725639918936316\
1791093853850215025471660274485005445141067926322404589720910423761396844606877\
6833882937036766653549194163002057800223606773706021049182780816950510262080420\
0839923496025503745439312268998233372140131455641689271449829210296345158759516\
4587130880000000000000000000000000000000000000000000000000000000000000000000000\
00000000000199                                                          p2
```

<u>Beispiel 5:</u> Die Bestimmung der Primzahlen (p_1 und p_2) im direkten Umfeld der Zahl $n_g = 2 \cdot 3 \cdot \ldots \cdot 107$:

```
•  for i from -151 to +127 do: if isprime (25663761175949999414479597815340071648394470+i)
   then print ("i", i, " = ", 25663761175949999414479597815340071648394470+i) end_if:
   end_for:

     "i", -151, " = ", 25663761175949999414479597815340071648394319     p1
                                                                        (278)
      "i", 127, " = ", 25663761175949999414479597815340071648394597     p2
```

<u>Beispiel 5:</u> Die Bestimmung der Primzahlen (p_1 und p_2) im direkten Umfeld der Zahl $n_g = 107!$:

```
    for i from -619 to +1069 do: if isprime (107!+i)
  then print ("i", i, " = ", 107!+i) end_if: end_for:

"i", -619, " = ",

1226520203196137939351751701038733888713156815438294505265325141201353532492214\
4249034658613287059061933743916719318560380966506520420000368175349759999999999\
999999999999381                                                         p1
                                                                        (1688)
"i", 1069, " = ",

1226520203196137939351751701038733888713156815438294505265325141201353532492214\
4249034658613287059061933743916719318560380966506520420000368175349760000000000\
000000000001069                                                         p2
```

<u>Beispiel 6:</u> Die Bestimmung der Primzahlen (p_1 und p_2) im direkten Umfeld der Zahl $n_g = 107!^\wedge 2$:

```
    for i from -1307 to +131 do: if isprime(107!^2+i)
  then print ("i", i, " = ", 107!^2+i) end_if: end_for:

"i", -1297, " = ",

1504351808848295499417894460780397828581573119471371724555258789925276162286373\
7298843366066990678729868559778272011138327361796038937279185328045229922207792\
8365195884629373430482383481121321791003138953226452790715644698971666304538274\
2500410660495880598101210211700019082997514881708983320575999999999999999999999\
99999999999999999999998703                                              p1
                                                                        (1428)
"i", 131, " = ",

1504351808848295499417894460780397828581573119471371724555258789925276162286373\
7298843366066990678729868559778272011138327361796038937279185328045229922207792\
8365195884629373430482383481121321791003138953226452790715644698971666304538274\
2500410660495880598101210211700019082997514881708983320576000000000000000000000\
00000000000000000000000131                                              p2
```

Beispiel 7: Die Bestimmung der Primzahlen (p_1 und p_2) im direkten Umfeld der Zahl n_g = 107!^3:

```
for i from -353 to + 467 do; if isprime (107!^3+i)
then print ("i", i, " = ", 107!^3+i) end_if: end_for:

"i", -353, " = ",

1845117886267089055997607246510834790809868744831863026568715419178051673564664\
2719399711475121515342697631431899258952720536147817294391928420835197430353618\
3187255212598549171291004418361249486940340090347613455181690557766246263978213\
2704997862463972078682598392145067185329303476999210013440705011113722771426729\
5988115079435971698633293522580599538247109517703680753418562617972485136148359\
7160807499262398323902379341718805870750046617599999999999999999999999999999999\
9999999999999999999999999999999999999999647
```

p_1

```
"i", 467, " = ",
```

(820)

```
1845117886267089055997607246510834790809868744831863026568715419178051673564664\
2719399711475121515342697631431899258952720536147817294391928420835197430353618\
3187255212598549171291004418361249486940340090347613455181690557766246263978213\
2704997862463972078682598392145067185329303476999210013440705011113722771426729\
5988115079435971698633293522580599538247109517703680753418562617972485136148359\
7160807499262398323902379341718805870750046617600000000000000000000000000000000\
0000000000000000000000000000000000000000467
```

p_2

Beispiel 8: Die Bestimmung der Primzahlen (p_1 und p_2) im direkten Umfeld der Zahl n_g = 103!^3:

```
for i from -1447 to + 601 do: if isprime (103!^3+i)
then print ("i", i, " = ", 103!^3+i) end_if: end_for:

"i", -1447, " = ",

9711521475932758626484978976009526568953179688167171099482746330054192960554390\
9136194029154945712699141890282808784694505460010647511755575368773504414313899\
9450827417288867175760925202156600034604099023046241437941509437523075922864130\
9952423814171022638519787180219648617586710835628270434865641135389429721852510\
0628980513853517719838130160823515353389198324931810246952199988756681921181911\
5083842830725470385864703999999999999999999999999999999999999999999999999999999\
999999999999998553
```

p_1

```
"i", 601, " = ",
```

(2048)

```
9711521475932758626484978976009526568953179688167171099482746330054192960554390\
9136194029154945712699141890282808784694505460010647511755575368773504414313899\
9450827417288867175760925202156600034604099023046241437941509437523075922864130\
9952423814171022638519787180219648617586710835628270434865641135389429721852510\
0628980513853517719838130160823515353389198324931810246952199988756681921181911\
5083842830725470385864704000000000000000000000000000000000000000000000000000000\
000000000000000601
```

p_2

Schlussfolgerungen:

1. Die Bestimmung der Primzahlen (p_1 und p_2) im direkten Umfeld der in den Beispielen 1 bis 8 analysierten geraden Zahlen $n_g = \Pi p^{\alpha_i}{}_i$ wurde gemäß der "horizontalen Testmethode" durchgeführt.

2. Aus Gesetz 1 und Gesetz 2 der Faktor-Zerlegung beliebiger Zahlen (n_g und n_u) folgt, dass die Addition bzw. Subtraktion von Primzahlen (oder Primzahl-Produkten), die in den Zahlen (n_g bzw. n_u) faktormäßig auftreten, niemals zu Primzahlen führen kann. Sofern hingegen Primzahlen (oder Produkte von Primzahlen) addiert bzw. subtrahiert werden, können (fallweise) größere Primzahlen entstehen.

Anmerkung:
Dieser Sachverhalt wird am Beispiel der Zahl $n_g = 2 \cdot 3 \cdot 5 \cdot .. \cdot 53 = 32589158477190044730$ an mehreren Beispielen demonstriert:

```
ifactor(32589158477190044730-2)
```

$$2^3 \cdot 630623621 \cdot 6459708571$$

- `ifactor(32589158477190044730+2)`

$2^2 \cdot 683 \cdot 11928681726643501$

- `ifactor(325891584771900447730-37)`

  $37 \cdot 8807880669510822891$

- `ifactor(325891584771900447730+59)`

  325891584771900447789

- `ifactor(325891584771900447730+61)`

  $300229 \cdot 1085476702023791$

- `ifactor(325891584771900447730+73)`

  325891584771900448803

3. Aus Gesetz 1 und Gesetz 2 folgt das Asymptote-Gesetz mit den charakteristischen Schwankungen der Primzahl-Dichten im Umfeld der hier analysierten Zahlen:

   - $n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 53$, $n_g = 53!$, $n_g = 53!\text{^}2$, $n_g = 53!\text{^}3$ ... $p_n = 53$

   - $n_g = 2 \cdot 3 \cdot 5 \cdot \ldots \cdot 107$, $n_g = 107!$, $n_g = 107!\text{^}2$, n_g 0 $107!\text{^}3$ ... $p_n = 107$

   - $n_g = 103!\text{^}3 < 53!\text{^}7$ ... $p_n = 103$

4. Die Zahlen: $n_g = 53!\text{^}7$, $n_g = 107!\text{^}3$, $n_g = 103!\text{^}3$ haben trotz einer vergleichbaren Stellenzahl in ihren

   Umgebungen unterschiedliche Primzahl-Dichten und dafür ursächlich sind die Folgen der Primzahlen, die faktoriell in diesen Zahlen auftrete.

5. Schließlich ist aus den Beispielen zu entnehmen, dass die in größeren Zahlenbereichen situierten nicht primen Zahlen, die in "Primzahl-Abständen" von den (analysierten) Zahlen auftreten, offensichtlich die Eigenschaft aufweisen aus Primzahl-Faktoren zu bestehen, in denen nur Primzahlen $p > p_n$ auftreten und damit Gesetz 2 erfüllen.

6. Diese Beispiele bestätigen das Asymptote-Gesetz und erklären die lokalen Schwankungen der Primzahl-Dichten. .

7. Wie wohl die Zahl $n_g = 53!\text{^}7$ größer als die Zahl $n_g = 103!\text{^}3$ ist, treten in der Umgebung dieser Zahl ($n_g = 53!\text{^}7$ zwangsläufig mehr Primzahlen auf als in der Umgebung der Zahl $n_g = 103!\text{^}3$.

Die nachfolgen Beispiele belegen, dass das Asymptote-Gesetz auch für die Verteilung von Zwillingen mod 2, mod 4 (und allgemein für Zwillinge mod 2n, n ≥1), die in wachsenden Zahlenbereichen situiert sind, Geltung hat:

Beispiel 1: Die Zwillinge mod 2, die im Intervall: 10^10 -1000 bis 10^10+1000 situiert sind:

```
for i from -1000 to +1000 do: if (isprime (10^10+i) and isprime (10^10+i+2))
  then print (("i", i, 10^10+i, " = " , 10^10+i+2)) end_if: end_for:

              "i", -983, 9999999017, " = ", 9999999019

              "i", -299, 9999999701, " = ", 9999999703
```

```
"i", 277, 10000000277, " = ", 10000000279

"i", 991, 10000000991, " = ", 10000000993
```

Anmerkung: In diesem Intervall sind 4 Zwillinge mod 2 situiert.

Beispiel 2: Die Zwillinge mod 2, die im Intervall: 10^20-4000 bis 10^20+4000 situiert sind:

```
for i from -4000 to +4000 do: if (isprime (10^20+i) and isprime (10^20+i+2))
  then print (("i", i, 10^20+i, " = ", 10^20+i+2)) end_if: end_for:

"i", -2309, 99999999999999997691, " = ", 99999999999999997693

"i", -2261, 99999999999999997739, " = ", 99999999999999997741

"i", 391, 100000000000000000391, " = ", 100000000000000000393

"i", 559, 100000000000000000559, " = ", 100000000000000000561
```

Anmerkung: In diesem, um den Faktor 10^10 vergrößerten Zahlenbereich .sind im vergrößerten Intervall 10^20-4000 bis 10^20+4000 ebenfalls 4 Zwillinge mod 2 situiert und dies bestätigt das Asymptote-Gesetz auch bezüglich der Dichte-Abnahme der Zwillinge mod 2.

Beispiel 3: Die Zwillinge mod 4, die im Intervall: 10^10 -1100 bis 10^10+1100 situiert sind:

```
for i from -1100 to +1100 do: if (isprime (10^10+i) and isprime (10^10+i+4))
  then print (("i", i, 10^10+i, " = " , 10^10+i+4)) end_if: end_for:

"i", -843, 9999999157, " = ", 9999999161

"i", -633, 9999999367, " = ", 9999999371

"i", -297, 9999999703, " = ", 9999999707

"i", 597, 10000000597, " = ", 10000000601

"i", 1047, 10000001047, " = ", 10000001051
```

Anmerkung: In diesem Intervall sind in der Umgebung der Zahl $n_g = 10^{10}$ fünf Zwillinge mod 4 situiert.

Beispiel 4: Die Zwillinge mod 4, die im Intervall: 10^20-2500 bis 10^20+2500 situiert sind:

```
for i from -2500 to +2500 do: if (isprime (10^20+i) and isprime (10^20+i+4))
  then print (("i", i, 10^20+i, " = ", 10^20+i+4)) end_if: end_for:

"i", -2493, 99999999999999997507, " = ", 99999999999999997511

"i", -2181, 99999999999999997819, " = ", 99999999999999997823

"i", -843, 99999999999999999157, " = ", 99999999999999999161

"i", -201, 99999999999999999799, " = ", 99999999999999999803

"i", 753, 100000000000000000753, " = ", 100000000000000000757
```

Anmerkung: In diesem, um den Faktor 10^10 vergrößerten Zahlenbereich, sind im vergrößerten Intervall 10^20-2500 bis 10^20+2500 ebenfalls fünf Zwillinge mod 4 situiert, und dies bestätigt die Geltung des Asymptote-Gesetzes auch bezüglich der Dichte-Abnahme der Zwillinge mod 4.

Beispiel 5: Die Zwillinge mod 10, die im Intervall: 23^50-25000 bis 23^50+25000 situiert sind:

```
for i from -25000 to + 25000 do: if (isprime (23^50+i) and isprime (23^50+i+10))
  then print (("i", i, 23^50+i, " = ", 23^50+i+10)) end_if: end_for:

"i", -3540,

  122008981252869411022491112993141891091036959856659100591281395339709,      p1

 " = ", 122008981252869411022491112993141891091036959856659100591281395339719   p2

"i", -1986,                                                                    (1554)
```

122008981252869411022491112993141891091036959856659100591281395341263, p_3

" = ", 122008981252869411022491112993141891091036959856659100591281395341273 p_4

"i", 2580, (4566)

122008981252869411022491112993141891091036959856659100591281395345829 p_5

, " = ",

122008981252869411022491112993141891091036959856659100591281395345839 p_6

"i", 3480, (900)

122008981252869411022491112993141891091036959856659100591281395346729 p_7

, " = ",

122008981252869411022491112993141891091036959856659100591281395346739 p_8

"i", 6042, (2562)

122008981252869411022491112993141891091036959856659100591281395349291 p_9

, " = ",

122008981252869411022491112993141891091036959856659100591281395349301 p_{10}

<u>Anmerkung:</u>

1. In diesem Intervall sind 5 Zwillinge mod 10 situiert, in den angegebenen Abständen: (1554), (4566), (900) und (2562) aufeinander folgen.

2. Die Primzahlen, die diese Zwillinge bilden, sind jeweils Elemente der Obergruppen OG I und OG II.

<u>Beispiel 6:</u>

```
for i from -200000 to +200000 do: if (isprime (23^100+i) and isprime (23^100+i+10))
   then print (("i", i, 23^100+i, " = ", 23^100+i+10)) end_if: end_for:
```

"i", -145128,

1488619150636303939379155658655975423198711965380136868657698820922243327853933\
1352152390143277346804233476592179447310859520222529730873 p_1

, " = ",

1488619150636303939379155658655975423198711965380136868657698820922243327853933\
1352152390143277346804233476592179447310859520222529730883 p_2

"i", -75534, (69594)

1488619150636303939379155658655975423198711965380136868657698820922243327853933\
1352152390143277346804233476592179447310859520222529800467 p_3

, " = ",

1488619150636303939379155658655975423198711965380136868657698820922243327853933\
1352152390143277346804233476592179447310859520222529800477 p_4

"i", 3018, ((78552)

1488619150636303939379155658655975423198711965380136868657698820922243327853933\
1352152390143277346804233476592179447310859520222529879019 p_5

, " = ",

1488619150636303939379155658655975423198711965380136868657698820922243327853933\
1352152390143277346804233476592179447310859520222529879029 p_6

"i", 52368, ((49350)

1488619150636303939379155658655975423198711965380136868657698820922243327853933\
1352152390143277346804233476592179447310859520222529928369 p_7

, " = ",

1488619150636303939379155658655975423198711965380136868657698820922243327853933\
1352152390143277346804233476592179447310859520222529928379 p_8

"i", 172530, (120162)

14886191506363039393791556586559754231987119653801368686576988209222433278539 33\
135215239014327734680423347659217944731085952022253004853 1 p_9

, " = ",

1488619150636303939379155658655975423198711965380136868657698820922243327853933\
1352152390143277346804233476592179447310859520222530048541 p_{10}

Anmerkung:

1. In dem vierfach erweiterten Intervall um die im Zahlenbereich weit verschobene Zahl n_u = 23^100 sind ebenfalls 5 Zwillinge mod 10 situiert, die in den wesentlich .größeren aufeinander folgenden Abständen: (69594), (78552), (49350), (120162) auftreten.

2. Dieser Sachverhalt bestätigt, dass das Asymptote-Gesetz auch für Zwillinge mod 10 gilt. Des weiteren offenbart sich (auch hier) das Phänomen der Dichte-Schwankungen dieser Zwillinge mod 10.

Hinweis:
Durch weitere Teste findet man die Geltung des Asymptote-Gesetzes für alle Zwillinge mod 2n, $n \geq 1$ bestätigt, wie auch das Phänomen der Dichte-Schwankungen..

Die Schlussfolgerungen:

1. Aus den vorgestellten Testen ergibt sich eine generelle Bestätigung des Asymptote-Gesetzes.

2. Die lokalen Schwankungen der Primzahl-Dichten in wachsenden Zahlenbereichen ist durch die Faktor-Struktur der Zahlen n_g bzw. n_u bedingt und damit eine Folge der Wirkungen von Gesetz 1 und Gesetz 2 der "Primzahl-Faktor-Genese".

3. Sehr große Zahlen ($n \to \infty$) können in "wachsenden Maße" aus Primzahl-Faktoren gebildet werden, in denen nur größere Primzahlen auftreten als die größte Primzahl (p_n) die in den Zahlen Πp_i, ($p_1, p_2, \ldots, p_n$) auftreten.

q.e.d.

Blockschema I

Die erweiterten Gleichungen für den Beweis des Hauptsatzes der Primzahl-Theorie:

1. Gleichung: $p = n_g + 1, + 3, +5, \ldots, + (n_u = p_{n+1} \cdot p_{n+2} \cdot \ldots \cdot p_{n+n})$
2. Gleichung: $p = n_g - 1, - 3, -5, \ldots., - (n_u = p_{n+1} \cdot p_{n+2} \cdot \ldots \cdot p_{n+n})$
3. Gleichung: $p = n_u + 2, +4, +6, \ldots, + (n_g = 2 \cdot p_{n+1} \cdot p_{n2} \cdot \ldots. \cdot p_{n+n})$
4. Gleichung: $p = n_u - 2, - 4, - 6,, \ldots, - (n_g = 2 \cdot p_{n+1} \cdot p_{n+2} \cdot \ldots \cdot p_{n+n})$

Primzahlen die in den addierten bzw. subtrahierten geraden bzw. ungeraden Zahlen als Faktoren auftreten, dürfen in den vorgegebenen geraden bzw. ungeraden Zahlen nicht auftreten. Unter dieser Prämisse ergeben sich fallweise aus den vier Gleichungen prime Zahlen oder aber Zahlen, in denen Primzahl-Faktoren mit größeren Primzahlen als die Primzahlen, die in den Zahlen: n_g bzw. n_u auftreten.

Blockschema II

Die erweiterten Dirichlet - Progressionen:

OG I:

Klasse 1, EZ1, Rest 1:= 41 + 20n ... 41, 61, 101, ...
Klasse 2, EZ3, Rest 1:= 13 + 20n ... 13, 53, 73, ...
Klasse 3, EZ7, Rest 1:= 17 + 20n ... 17, 37, 97, ...
Klasse 4, EZ9, Rest 1:= 29 + 20n ... 29, 89, 109, ...

OG II:

Klasse 5, EZ1, Rest 3:= 11 + 20n ... 11, 31, 71, ...
Klasse 6, EZ3, Rest 3:= 3 + 20n ... 3, 23, 43, ...
Klasse 7, EZ7, Rest 3:= 7 + 20n ... 7, 47, 67, ...
Klasse 8, EZ9, Rest 3:= 19 + 20n ... 19, 59, 79, ...

Blockschema III

Die Bildungs-Gesetze der *Folgen* in denen Zwillinge mod 2n, n ≥1 auftreten

Zwillinge mod 2n, n ≥1:	*Folge 1*	*Folge 2*	*Folge 3*	*Folge 4*
Zwillinge mod 2:	12 + 30n	18 + 30n	30 + 30n	---------
Zwillinge mod 4:	5 + 10n	9 + 10n	11 + 10n	---------
Zwillinge mod 6:	14 + 10n	16 + 10n	10 + 10n	---------
Zwillinge mod 8:	7 + 10n	13 + 10n	15 + 10n	---------
Zwillinge mod 10:	6 + 10n	8 + 10n	12 + 10n	14 + 10n
Zwillinge mod 12:	7 + 10n	13 + 10n	15 + 10n	---------
Zwillinge mod 14:	10 + 10n	14 + 10n	16 + 10n	---------
Zwillinge mod 16:	9 + 10n	11 + 10n	15 + 10n	---------
Zwillinge mod 18	10 + 10n	12 + 10n	18 + 10n	---------
Zwillinge mod 20:	11 + 10n	13 + 10n	17 + 10n	19 + 10n
Zwillinge mod 100:	51 + 10n	53 + 10n	57 + 10n	59 + 10n

Neue Aspekte der Primzahl-Theorie

Teil II

Primzahl-Wellen in den Weiten der natürlichen Zahlen

Der Nachweis, dass es unendlich viele Anti-Goldbach'sche Zerlegungen gerader Zahlen in die Differenzen zweier Primzahlen der Form: $n_g = p_1 - p_2$ gibt

Verfasser: Georg Viktor Cwienk

Bochum, August 2006

Inhaltsverzeichnis:

Seiten-Umfang insgesamt: 60 Seiten DIN A4

Anlagenverzeichnis:

Anlage 1: Die Liste der ungeraden Primzahlen im Intervall n:= 0 bis 100.000

Anlage 2: Beispiel 1 für "verkoppelte" Goldbach-Zerlegungen

Anlage 3: Beispiel 1 für "verkoppelte" Anti-Goldbach-Zerlegungen

Anlage 4: Beispiel 1 für "verkoppelte" Quasi-Goldbach-Zerlegungen

Anlage 5: Beispiel 1 für "verkoppelte" Quasi-Anti-Goldbach-Zerlegungen

Anlage 6: Beispiel 1 für gemischte Goldbach - und Anti-Goldbach-Zerlegungen

Anlage 7: Beispiel 2 für gemischte Goldbach - und Anti-Goldbach-Zerlegungen

Anlage 8: Die Quasi-Goldbach-Zerlegungen der geraden Zahlen:
$n_{g,1} = (10!+12-i)$, $n_{g,2} = (10!+14-i)$, $n_{g,3} = (10!+18-i)$, $n_{g,4} = (10!+20-i)$
$n_{g,5} = (10!+30-i)$, $n_{g,6} = (10!+32-i)$ im Intervall: 0 bis 10!

Anlage 9: Die Quasi-Anti-Goldbach-Zerlegungen der geraden Zahlen:
$n_{g,1} = (10+i)$, $n_{g,2} = (12+i)$, $n_{g,3} = (16+i)$, $n_{g,4} = (18+i)$
$n_{g,5} = (28+i)$, $n_{g,6} = (30+i)$ im Intervall: 0 bis 10!

Anlage 10: Die Goldbach-Zerlegungen der geraden Zahlen:
$n_{g,1} = (10!+12-i)$, $n_{g,2} = (10!+14-i)$, $n_{g,3} = (10!+18-i)$, $n_{g,4} = (10!+20-i)$
$n_{g,5} = (10!+30-i)$, $n_{g,6} = (10!+32-i)$ im Intervall: 0 bis 10!

Anlage 11: Die Anti-Goldbach-Zerlegungen der geraden Zahlen:
$n_{g,1} = (10+i)$, $n_{g,2} = (12+i)$, $n_{g,3} = (16+i)$, $n_{g,4} = (16+i)$
$n_{g,5} = (28+i)$, $n_{g,6} = (30+i)$ im Intervall: 0 bis 10!

Anlage 12: Beispiel 1 für die Existenz von "Tsunami - Primzahl-Wellen" von Goldbach-Zerlegungen gerader Zahlen, die in großen Abständen auftreten.

Anlage 13: Beispiel 1 für die Existenz von "Tsunami - Primzahl-Wellen" von Anti-Goldbach-Zerlegungen gerader Zahlen, die in großen Abständen auftreten.

Anlage 14: Beispiel 1 für die Existenz von "Tsunami - Primzahl-Wellen" von Goldbach-Zerlegungen und Anti-Goldbach-Zerlegungen gerader Zahlen. die in großen Abständen auftreten.

1. Einführung:

Diese Monographie wurde im Auftrag der dl-DATEN GmbH erstellt und wird nicht veröffentlicht, da sie "Basis-Informationen" für die Datenverschlüsselung umfasst, die Inhalt von zwei Patent-Anmeldungen beinhaltet.

Patent-Anmeldung 1:
"Die SynD-A Patentschrift für ein System nicht Dechiffrierbarer Authentisierung"

Patent-Anmeldung 2:
"Ein System gesicherter (nicht entschlüsselbarer) Text-Übertragung"

2. Rückblick auf den Beweis der Goldbach'schen Vermutung:

Diese Monographie ist eine "Weiter-Entwicklung" der folgenden Monographien:

1. Der Beweis der Goldbach'schen Vermutung - ISBN: 3 - 9805713 - 5 - 1 (April, 2006)

2. Neue Aspekte der Primzahl-Theorie (Mai, 2006) - unveröffentlicht.

Die in den hier zitierten Monographien gefundenen (bisher unbekannten) Gesetzmäßigkeiten. die innerhalb der unendlichen Folge der Primzahlen: $p := 2, 3, 5, 7, \ldots, p_i \to \infty$ auftreten, werden in der hier vorgestellten Monographie (entscheidend) um die folgenden erweitert - siehe Abschnitt 3.

Zunächst wird der Beweis erbracht, dass die "Anti - Goldbach - Zerlegungen" die gleiche (nachstehende) Summenformel erfüllen, die für die Goldbach-Zerlegungen deduziert wurde:

$$\lim_{n \to \infty} \Sigma Z(n) = \pi(n)^2 \quad \text{.... Summenformel}$$

Anmerkung:
Diese Formel wurde in der Monographie: Der Beweis der Goldbach'schen Vermutung deduziert.

Nach dem Gesetz der "Anti-Goldbach-Zerlegung" hat jede gerade Zahl unendlich viele Anti-Goldbach-Zerlegungen. So hat auch die kleinste gerade Zahl: $n_g = 2$ unendlich viele Anti-Goldbach-Zerlegungen, da es unendlich viele Primzahl-Zwillinge mod 2: p_i, $p_i + 2$ gibt, deren Differenz: $p_i + 2 - p_i = 2$ ist. Und nachdem es (wie in der 2. Monographie bewiesen wurde) ebenfalls unendlich viele Zwillinge mod 2n, $n > 1$ gibt, hat jede gerade Zahl (n_g) unendlich viele Anti-Goldbach-Zerlegungen. In diesem Sinne unterscheiden sich (auf den ersten Blick) die Goldbach-Zerlegungen von den Anti - Goldbach - Zerlegungen.

Der Beweis, dass die Summenformel, auch für die Anti-Goldbach-Zerlegungen gilt, kann leicht erbracht werden. Es genügt hier die Anti - Goldbach - und Goldbach - Zerlegungen im Intervall: 0 bis 10^2 zu analysieren:

- ```
  for i from 0 to 2*10^2 do: if isprime (i)and isprime (i+2) then
  print ("i",i,i+2) end_if: end_for:
  ```

"i", 3, 5	bzw.	5 - 3 = 2
"i", 5, 7	bzw.	7 - 5 = 2
"i", 11, 13	bzw.	13 - 11 = 2
"i", 17, 19	bzw.	19 - 17 = 2
"i", 29, 31	bzw.	31 - 29 = 2
"i", 41, 43	bzw.	43 - 41 = 2
"i", 59, 61	bzw.	61 - 59 = 2
"i", 71, 73	bzw.	73 - 71 = 2
"i", 101, 103	bzw.	103 - 101 = 2
"i", 107, 109	bzw.	109 - 107 = 2
"i", 137, 139	bzw.	139 - 137 = 2
"i", 149, 151	bzw.	151 - 149 = 2
"i", 179, 181	bzw.	181 - 179 = 2
"i", 191, 193	bzw.	193 - 191 = 2
"i", 197, 199	bzw.	199 - 197 = 2

<u>Anmerkung:</u>
Im Intervall: 0 bis 2*10^2 liegt eine 15-malige Zerlegung der Zahl $n_g = 2$ in die Differenz zweier Primzahlen: $(p_i + 2) - p_i$ vor.

- ```
  for i from 0 to 2*10^2 do: if isprime (i)and isprime (i-2) then
  print ("i",i,i-2) end_if: end_for:
  ```

"i", 5, 3	bzw.	5 + 3 = 8
"i", 7, 5	bzw.	7 + 5 = 12
"i", 13, 11	bzw.	13 + 11 = 24
"i", 19, 17	bzw.	19 + 17 = 36
"i", 31, 29	bzw.	31 + 29 = 60
"i", 43, 41	bzw.	43 + 41 = 84
"i", 61, 59	bzw.	61 + 59 = 120
"i", 73, 71	bzw.	73 + 71 = 144
"i", 103, 101	bzw.	103 + 101 = 204
"i", 109, 107	bzw.	109 + 107 = 216
"i", 139, 137	bzw.	139 + 137 = 276
"i", 151, 149	bzw.	151 + 149 = 300
"i", 181, 179	bzw.	181 + 179 = 360
"i", 193, 191	bzw.	193 + 191 = 384
"i", 199, 197	bzw.	199 + 197 = 396

<u>Anmerkungen</u>·
Im Intervall: 0 bis 2*10^2 sind 15 Goldbach-Zerlegungen diverser gerader Zahlen situiert, wobei jeweils <u>eine</u> Goldbach-Zerlegung dieser Zahlen in den 15 Zeilen auftritt!

Nachfolgend wird die Zahl aller Goldbach - Zerlegungen der Zahl: $n_g = 36$ vorgestellt:

- `for i from 0 to 10^2 do: if isprime (i) and isprime (36-i) then print ("i",i,36-i) end_if: end_for:`

```
"i", 5, 31      bzw.   31 + 5 = 36
"i", 7, 29      bzw.   29 + 7 = 36
"i", 13, 23     bzw.   23 + 13 = 36
"i", 17, 19     bzw.   19 + 17 = 36
```

Anmerkungen:

1. Die gerade Zahl $n_g = 36$ hat 4 Goldbach - Zerlegungen.
2. Die unterstrichene Zerlegung: 19 + 17 = 36 entspricht der Anti - Goldbach - Zerlegung: 19 - 17 = 2.
3. Hiermit ist der Beweis erbracht, dass allen Anti - Goldbach - Zerlegungen eine definierte Goldbach - Zerlegung entspricht - q.e.d.

---

3. Die "verkoppelten" Goldbach- und Anti-Goldbach-Zerlegungen gerader Zahlen:

Aus den vom Verfasser angestellten (tiefgreifenden) Untersuchungen resultieren drei wichtige Gesetze, die hier unter Beweis gestellt werden sollen und durch Beispiele belegt werden:

Gesetz 1:

"Alle geraden Zahlen: $n_g := 2, 4, 6, 8, \dots, n_g \to \infty$ können als Differenzen zweier Primzahlen: $p_j - p_i$ ($p_j > p_i$) dargestellt werden".

Dieser Sachverhalt wird vom Autor als die "Anti-Goldbach-Zerlegung" bezeichnet, die in der Mathematik bisher unbekannt ist, zumindest (bisher) nicht näher analysiert wurde

$$\lim_{n \to \infty} \Sigma Z(n) = \pi(n)^2 \quad \text{.... Summenformel}$$

Nach dem Gesetz der "Anti-Goldbach-Zerlegung" hat jede gerade Zahl unendlich viele Anti-Goldbach-Zerlegungen. So hat auch die kleinste gerade Zahl: $n_g = 2$ unendlich viele Anti-Goldbach-Zerlegungen, da es unendlich viele Primzahl-Zwillinge mod 2: p_i, $p_i + 2$ gibt, deren Differenz: $p_i + 2 - p_i = 2$ ist. Und nachdem es (wie in der 2. Monographie bewiesen wurde) ebenfalls unendlich viele Zwillinge mod 2n, n > 1 gibt, hat jede gerade Zahl (n_g) unendlich viele Anti-Goldbach-Zerlegungen. In diesem Sinne unterscheiden sich (auf den ersten Blick) die Goldbach-Zerlegungen von den Anti - Goldbach - Zerlegungen.

Gesetz 2:

"in der unendlichen Folge der Primzahlen wiederholen sich in großen Zahlen-Intervallen fortlaufend gekoppelte Goldbach'sche, wie auch Anti-Goldbach'sche Zerlegungen, die innerhalb der fallenden bzw. steigenden Primzahl-Folge "Abstands-Wellen" bilden, in denen die aufeinander folgenden Zerlegungen unentwegt gleiche Abstände der Primzahlen aufweisen.

Gesetz 3:

"In der unendlichen Folge der Primzahlen, können unendlich viele "kombinierte" Goldbach- und Anti-Goldbach-Zerlegungen auftreten, wobei der Abstand zwischen den Goldbach-Zerlegungen und den Anti-Goldbach-Zerlegungen "sprunghaft" steigt.

Die (uneingeschränkte) Gültigkeit dieser drei Gesetze wird nachfolgend an drei Beispielen demonstriert, die auf der Basis von (jeweils definierten) Boole'schen Algorithmen errechnet wurden:

Beispiel 1 für Gesetz 2:

- ```
  for i from 0 to 10^7 do: if isprime (i)and isprime
  (10!+42300-i)and isprime (10!+54012-i)and isprime(10!+66588-i)and
  isprime(10!+86070-i)and isprime (10!+95462-i)and isprime
  (10!+99972-i) thenprint("i",i,10!+42300-i,10!+54012-i,10!+66588-i,
   10!+86070-i,10!+95462-i,10!+99972-i) end_if: end_for:
  ```

```
"i", 110629, 3560471, 3572183, 3584759, 3604241, 3613633, 3618143
"i", 232681, 3438419, 3450131, 3462707, 3482189, 3491581, 3496091
"i", 330409, 3340691, 3352403, 3364979, 3384461, 3393853, 3398363
"i", 332221, 3338879, 3350591, 3363167, 3382649, 3392041, 3396551
"i", 628399, 3042701, 3054413, 3066989, 3086471, 3095863, 3100373
"i", 894511, 2776589, 2788301, 2800877, 2820359, 2829751, 2834261
"i", 1049281, 2621819, 2633531, 2646107, 2665589, 2674981, 2679491
"i", 1200361, 2470739, 2482451, 2495027, 2514509, 2523901, 2528411
"i", 1238749, 2432351, 2444063, 2456639, 2476121, 2485513, 2490023
"i", 1738669, 1932431, 1944143, 1956719, 1976201, 1985593, 1990103
"i", 2056441, 1614659, 1626371, 1638947, 1658429, 1667821, 1672331
"i", 2568091, 1103009, 1114721, 1127297, 1146779, 1156171, 1160681
"i", 2929621, 741479, 753191, 765767, 785249, 794641, 799151
"i", 3020551, 650549, 662261, 674837, 694319, 703711, 708221
"i", 3306529, 364571, 376283, 388859, 408341, 417733, 422243
"i", 3325219, 345881, 357593, 370169, 389651, 399043, 403553
"i", 3335221, 335879, 347591, 360167, 379649, 389041, 393551
"i", 3371209, 299891, 311603, 324179, 343661, 353053, 357563
"i", 3420691, 250409, 262121, 274697, 294179, 303571, 308081
"i", 3445639, 225461, 237173, 249749, 269231, 278623, 283133
"i", 3457309, 213791, 225503, 238079, 257561, 266953, 271463
"i", 3461641, 209459, 221171, 233747, 253229, 262621, 267131
"i", 3528289, 142811, 154523, 167099, 186581, 195973, 200483
"i", 3632449, 38651, 50363, 62939, 82421, 91813, 96323
"i", 3645979, 25121, 36833, 49409, 68891, 78283, 82793
```

- Hier treten im Intervall: 0 bis 10^7 25 6-fach-kombinierte Goldbach-Zerlegungen auf.
- In allen 25 Zeilen treten die (sechs) Primzahlen in sich wiederholenden gleichen Abständen auf.
```

Beispiel 2 für Gesetz 2:

```
for i from 0 to 10^7 do: if isprime (i) and isprime
(10!+42298+i)and isprime (10!+54010+i)and isprime (10!+66586+i)and
isprime (10!+86068+i)and isprime (10!+95460+i)and isprime
(10!+99970+i) then print ("i",i,10!+42298+i,10!+54010+i,
10!+66586+i,10!+86068+i,10!+95460+i,10!+99970+i) end_if:
end_for:
```

```
"i", 966883, 4637981, 4649693, 4662269, 4681751, 4691143, 4695653
"i", 979651, 4650749, 4662461, 4675037, 4694519, 4703911, 4708421
"i", 1354291, 5025389, 5037101, 5049677, 5069159, 5078551, 5083061
"i", 1372771, 5043869, 5055581, 5068157, 5087639, 5097031, 5101541
"i", 2034061, 5705159, 5716871, 5729447, 5748929, 5758321, 5762831
"i", 2094373, 5765471, 5777183, 5789759, 5809241, 5818633, 5823143
"i", 3135553, 6806651, 6818363, 6830939, 6850421, 6859813, 6864323
"i", 3736573, 7407671, 7419383, 7431959, 7451441, 7460833, 7465343
"i", 4209871, 7880969, 7892681, 7905257, 7924739, 7934131, 7938641
"i", 4266991, 7938089, 7949801, 7962377, 7981859, 7991251, 7995761
"i", 4581781, 8252879, 8264591, 8277167, 8296649, 8306041, 8310551
"i", 7493851, 11164949, 11176661, 11189237, 11208719, 11218111, 11222621
"i", 7524553, 11195651, 11207363, 11219939, 11239421, 11248813, 11253323
"i", 7566133, 11237231, 11248943, 11261519, 11281001, 11290393, 11294903
"i", 9597043, 13268141, 13279853, 13292429, 13311911, 13321303, 13325813
"i", 9973531, 13644629, 13656341, 13668917, 13688399, 13697791, 13702301
```

Hier treten im Intervall: 0 bis 10^7 16 (sechsfach kombinierte) Anti-Goldbach-Zerlegungen auf und in allen 16 Zeilen treten gleiche Abstände zwischen den Primzahlen auf.

Beispiel 1 für Gesetz 3:

```
for i from 0 to 10^7 do: if isprime (i)and isprime
(10!+42300-i)and isprime (10!+54012-i)and isprime
(10!+665588-i)and
isprime (10!+86068+i)and isprime (10!+95460+i)and isprime
(10!+99970+i) then print ("i",i,10!+42300-i,10!+54012-i,
10!+665588-i,10!+86068+i,10!+95460+i,10!+99970+i) end_if:
end_for:
```

42299

86069

95461

99971

```
"i", 73771, 3597329, 3609041, 4220617, 3788639, 3798031, 3802541
"i", 147409, 3523691, 3535403, 4146979, 3862277, 3871669, 3876179
"i", 424429, 3246671, 3258383, 3869959, 4139297, 4148689, 4153199
"i", 637309, 3033791, 3045503, 3657079, 4352177, 4361569, 4366079
"i", 682069, 2989031, 3000743, 3612319, 4396937, 4406329, 4410839
"i", 888109, 2782991, 2794703, 3406279, 4602977, 4612369, 4616879
"i", 1167241, 2503859, 2515571, 3127147, 4882109, 4891501, 4896011
```

"i", 1356721, 2314379, 2326091, 2937667, 5071589, 5080981, 5085491

"i", 1536991, 2134109, 2145821, 2757397, 5251859, 5261251, 5265761

"i", 1644991, 2026109, 2037821, 2649397, 5359859, 5369251, 5373761

"i", 1701289, 1969811, 1981523, 2593099, 5416157, 5425549, 5430059

"i", 1928929, 1742171, 1753883, 2365459, 5643797, 5653189, 5657699

"i", 2063059, 1608041, 1619753, 2231329, 5777927, 5787319, 5791829

"i", 2355439, 1315661, 1327373, 1938949, 6070307, 6079699, 6084209

"i", 2356429, 1314671, 1326383, 1937959, 6071297, 6080689, 6085199

"i", 2435119, 1235981, 1247693, 1859269, 6149987, 6159379, 6163889

"i", 2440579, 1230521, 1242233, 1853809, 6155447, 6164839, 6169349

"i", 2694499, 976601, 988313, 1599889, 6409367, 6418759, 6423269

"i", 2727301, 943799, 955511, 1567087, 6442169, 6451561, 6456071

"i", 2777161, 893939, 905651, 1517227, 6492029, 6501421, 6505931

"i", 2851789, 819311, 831023, 1442599, 6566657, 6576049, 6580559

"i", 3033409, 637691, 649403, 1260979, 6748277, 6757669, 6762179

"i", 3135049, 536051, 547763, 1159339, 6849917, 6859309, 6863819

"i", 3249391, 421709, 433421, 1044997, 6964259, 6973651, 6978161

"i", 3269719, 401381, 413093, 1024669, 6984587, 6993979, 6998489

"i", 3291709, 379391, 391103, 1002679, 7006577, 7015969, 7020479

"i", 3381919, 289181, 300893, 912469, 7096787, 7106179, 7110689

"i", 3404689, 266411, 278123, 889699, 7119557, 7128949, 7133459

"i", 3461089, 210011, 221723, 833299, 7175957, 7185349, 7189859

"i", 3525979, 145121, 156833, 768409, 7240847, 7250239, 7254749

"i", 3529069, 142031, 153743, 765319, 7243937, 7253329, 7257839

"i", 3595951, 75149, 86861, 698437, 7310819, 7320211, 7324721

- Im Intervall 0 bis 10^7 treten 32 kombinierte Goldbach- und Anti-Goldbach-Zerlegungen auf. In allen 32 Zeilen treten zwischen den Primzahlen der Goldbach- und der Anti-Goldbach-Zerlegungen stets gleiche Abstände auf.
-
- Die Abstände zwischen den Primzahlen der Goldbach- und der Anti-Goldbach-
- Zerlegungen unterliegen von der untersten Zeile bis zur obersten Zeile einem "sprunghaften" Zuwachs.

4. Der Nachweis, dass es gleich viele Quasi-Goldbach - und Quasi-Anti-Goldbach-Zerlegungen in definierten Zahlen-intervallen gibt:

Hier wird auf die Anlagen: **Anlage 8** und **Anlage 9** verwiesen, in denen die Zahlen der Quasi-Goldbach- und Quasi-Anti-Goldbach-Zerlegungen im Intervall: 0 bis 10! bestimmt wurden.

Aus diesen Anlagen ist zu entnehmen, dass in diesem (gleichen) Intervall sowohl 22 Quasi-Goldbach, als auch 22 Quasi-Anti-Goldbach-Zerlegungen auftreten.

Die Zahlen der (regulären) Goldbach-Zerlegungen und der Anti-Goldbach- Zerlegungen unterscheiden sich hingegen im gleichen Intervall, was aus den Anlagen: **Anlage 10** und **Anlage 11** entnommen werden kann.

5. Die Schlussfolgerungen für die "kryptographische Nutzung" der verkoppelten Goldbach-Zerlegungen und Anti-Goldbach-Zerlegungen gerader Zahlen:

Aus den vorgestellten Untersuchungen (siehe die Anlagen: **2** bis **14**) ergibt sich, dass die Goldbach- und Anti-Goldbach-Zerlegungen die Basis für eine hochgradig gesicherte Daten-Verschlüsselung bilden können - sowohl für die "Phase der Authentisierung" ala aoch für die "Nutzdaten-Übertragung".

Hier wird (insbesondere) auf die "Tsunami-artigen - Primzahl-Wellen" verwiesen, die in den Anlagen: **Anlage 12, 13** und **14** generiert wurden.

Sobald für die "Authentisierung" rollierend Primzahl-Folgen aus alternierenden Primzahl-Wellen eingesetzt werden, kann diese Verschlüsselung (von Dritten) nicht erkannt werden.

Bochum, August 2006

Der Verfasser: Dr. Georg Viktor Cwienk

Anlage 1 Die Ordnungs-Liste der ungeraden Primzahlen im Intervall n:= 0 bis 100.000

Primzahlen

3	5	7	11	13	17	19	23
29	31	37	41	43	47	53	59
61	67	71	73	79	83	89	97
101	103	107	109	113	127	131	137
139	149	151	157	163	167	173	179
181	191	193	197	199	211	223	227
229	233	239	241	251	257	263	269
271	277	281	283	293	307	311	313
317	331	337	347	349	353	359	367
373	379	383	389	397	401	409	419
421	431	433	439	443	449	457	461
463	467	479	487	491	499	503	509
521	523	541	547	557	563	569	571
577	587	593	599	601	607	613	617
619	631	641	643	647	653	659	661
673	677	683	691	701	709	719	727
733	739	743	751	757	761	769	773
787	797	809	811	821	823	827	829
839	853	857	859	863	877	881	883
887	907	911	919	929	937	941	947
953	967	971	977	983	991	997	1009
1013	1019	1021	1031	1033	1039	1049	1051
1061	1063	1069	1087	1091	1093	1097	1103
1109	1117	1123	1129	1151	1153	1163	1171
1181	1187	1193	1201	1213	1217	1223	1229
1231	1237	1249	1259	1277	1279	1283	1289
1291	1297	1301	1303	1307	1319	1321	1327
1361	1367	1373	1381	1399	1409	1423	1427
1429	1433	1439	1447	1451	1453	1459	1471
1481	1483	1487	1489	1493	1499	1511	1523
1531	1543	1549	1553	1559	1567	1571	1579
1583	1597	1601	1607	1609	1613	1619	1621
1627	1637	1657	1663	1667	1669	1693	1697
1699	1709	1721	1723	1733	1741	1747	1753
1759	1777	1783	1787	1789	1801	1811	1823
1831	1847	1861	1867	1871	1873	1877	1879
1889	1901	1907	1913	1931	1933	1949	1951
1973	1979	1987	1993	1997	1999	2003	2011
2017	2027	2029	2039	2053	2063	2069	2081
2083	2087	2089	2099	2111	2113	2129	2131
2137	2141	2143	2153	2161	2179	2203	2207
2213	2221	2237	2239	2243	2251	2267	2269
2273	2281	2287	2293	2297	2309	2311	2333
2339	2341	2347	2351	2357	2371	2377	2381
2383	2389	2393	2399	2411	2417	2423	2437
2441	2447	2459	2467	2473	2477	2503	2521
2531	2539	2543	2549	2551	2557	2579	2591
2593	2609	2617	2621	2633	2647	2657	2659
2663	2671	2677	2683	2687	2689	2693	2699
2707	2711	2713	2719	2729	2731	2741	2749
2753	2767	2777	2789	2791	2797	2801	2803
2819	2833	2837	2843	2851	2857	2861	2879
2887	2897	2903	2909	2917	2927	2939	2953
2957	2963	2969	2971	2999	3001	3011	3019
3023	3037	3041	3049	3061	3067	3079	3083
3089	3109	3119	3121	3137	3163	3167	3169
3181	3187	3191	3203	3209	3217	3221	3229
3251	3253	3257	3259	3271	3299	3301	3307
3313	3319	3323	3329	3331	3343	3347	3359
3361	3371	3373	3389	3391	3407	3413	3433
3449	3457	3461	3463	3467	3469	3491	3499
3511	3517	3527	3529	3533	3539	3541	3547

.....3557.....3559.....3571.....3581.....3583.....3593.....3607.....3613
.....3617.....3623.....3631.....3637.....3643.....3659.....3671.....3673
.....3677.....3691.....3697.....3701.....3709.....3719.....3727.....3733
.....3739.....3761.....3767.....3769.....3779.....3793.....3797.....3803
.....3821.....3823.....3833.....3847.....3851.....3853.....3863.....3877
.....3881.....3889.....3907.....3911.....3917.....3919.....3923.....3929
.....3931.....3943.....3947.....3967.....3989.....4001.....4003.....4007
.....4013.....4019.....4021.....4027.....4049.....4051.....4057.....4073
.....4079.....4091.....4093.....4099.....4111.....4127.....4129.....4133
.....4139.....4153.....4157.....4159.....4177.....4201.....4211.....4217
.....4219.....4229.....4231.....4241.....4243.....4253.....4259.....4261
.....4271.....4273.....4283.....4289.....4297.....4327.....4337.....4339
.....4349.....4357.....4363.....4373.....4391.....4397.....4409.....4421
.....4423.....4441.....4447.....4451.....4457.....4463.....4481.....4483
.....4493.....4507.....4513.....4517.....4519.....4523.....4547.....4549
.....4561.....4567.....4583.....4591.....4597.....4603.....4621.....4637
.....4639.....4643.....4649.....4651.....4657.....4663.....4673.....4679
.....4691.....4703.....4721.....4723.....4729.....4733.....4751.....4759
.....4783.....4787.....4789.....4793.....4799.....4801.....4813.....4817
.....4831.....4861.....4871.....4877.....4889.....4903.....4909.....4919
.....4931.....4933.....4937.....4943.....4951.....4957.....4967.....4969
.....4973.....4987.....4993.....4999.....5003.....5009.....5011.....5021
.....5023.....5039.....5051.....5059.....5077.....5081.....5087.....5099
.....5101.....5107.....5113.....5119.....5147.....5153.....5167.....5171
.....5179.....5189.....5197.....5209.....5227.....5231.....5233.....5237
.....5261.....5273.....5279.....5281.....5297.....5303.....5309.....5323
.....5333.....5347.....5351.....5381.....5387.....5393.....5399.....5407
.....5413.....5417.....5419.....5431.....5437.....5441.....5443.....5449
.....5471.....5477.....5479.....5483.....5501.....5503.....5507.....5519
.....5521.....5527.....5531.....5557.....5563.....5569.....5573.....5581
.....5591.....5623.....5639.....5641.....5647.....5651.....5653.....5657
.....5659.....5669.....5683.....5689.....5693.....5701.....5711.....5717
.....5737.....5741.....5743.....5749.....5779.....5783.....5791.....5801
.....5807.....5813.....5821.....5827.....5839.....5843.....5849.....5851
.....5857.....5861.....5867.....5869.....5879.....5881.....5897.....5903
.....5923.....5927.....5939.....5953.....5981.....5987.....6007.....6011
.....6029.....6037.....6043.....6047.....6053.....6067.....6073.....6079
.....6089.....6091.....6101.....6113.....6121.....6131.....6133.....6143
.....6151.....6163.....6173.....6197.....6199.....6203.....6211.....6217
.....6221.....6229.....6247.....6257.....6263.....6269.....6271.....6277
.....6287.....6299.....6301.....6311.....6317.....6323.....6329.....6337
.....6343.....6353.....6359.....6361.....6367.....6373.....6379.....6389
.....6397.....6421.....6427.....6449.....6451.....6469.....6473.....6481
.....6491.....6521.....6529.....6547.....6551.....6553.....6563.....6569
.....6571.....6577.....6581.....6599.....6607.....6619.....6637.....6653
.....6659.....6661.....6673.....6679.....6689.....6691.....6701.....6703
.....6709.....6719.....6733.....6737.....6761.....6763.....6779.....6781
.....6791.....6793.....6803.....6823.....6827.....6829.....6833.....6841
.....6857.....6863.....6869.....6871.....6883.....6899.....6907.....6911
.....6917.....6947.....6949.....6959.....6961.....6967.....6971.....6977
.....6983.....6991.....6997.....7001.....7013.....7019.....7027.....7039
.....7043.....7057.....7069.....7079.....7103.....7109.....7121.....7127
.....7129.....7151.....7159.....7177.....7187.....7193.....7207.....7211
.....7213.....7219.....7229.....7237.....7243.....7247.....7253.....7283
.....7297.....7307.....7309.....7321.....7331.....7333.....7349.....7351
.....7369.....7393.....7411.....7417.....7433.....7451.....7457.....7459
.....7477.....7481.....7487.....7489.....7499.....7507.....7517.....7523
.....7529.....7537.....7541.....7547.....7549.....7559.....7561.....7573
.....7577.....7583.....7589.....7591.....7603.....7607.....7621.....7639
.....7643.....7649.....7669.....7673.....7681.....7687.....7691.....7699
.....7703.....7717.....7723.....7727.....7741.....7753.....7757.....7759
.....7789.....7793.....7817.....7823.....7829.....7841.....7853.....7867
.....7873.....7877.....7879.....7883.....7901.....7907.....7919.....7927
.....7933.....7937.....7949.....7951.....7963.....7993.....8009.....8011

.....8017.....8039.....8053.....8059.....8069.....8081.....8087.....8089
.....8093.....8101.....8111.....8117.....8123.....8147.....8161.....8167
.....8171.....8179.....8191.....8209.....8219.....8221.....8231.....8233
.....8237.....8243.....8263.....8269.....8273.....8287.....8291.....8293
.....8297.....8311.....8317.....8329.....8353.....8363.....8369.....8377
.....8387.....8389.....8419.....8423.....8429.....8431.....8443.....8447
.....8461.....8467.....8501.....8513.....8521.....8527.....8537.....8539
.....8543.....8563.....8573.....8581.....8597.....8599.....8609.....8623
.....8627.....8629.....8641.....8647.....8663.....8669.....8677.....8681
.....8689.....8693.....8699.....8707.....8713.....8719.....8731.....8737
.....8741.....8747.....8753.....8761.....8779.....8783.....8803.....8807
.....8819.....8821.....8831.....8837.....8839.....8849.....8861.....8863
.....8867.....8887.....8893.....8923.....8929.....8933.....8941.....8951
.....8963.....8969.....8971.....8999.....9001.....9007.....9011.....9013
.....9029.....9041.....9043.....9049.....9059.....9067.....9091.....9103
.....9109.....9127.....9133.....9137.....9151.....9157.....9161.....9173
.....9181.....9187.....9199.....9203.....9209.....9221.....9227.....9239
.....9241.....9257.....9277.....9281.....9283.....9293.....9311.....9319
.....9323.....9337.....9341.....9343.....9349.....9371.....9377.....9391
.....9397.....9403.....9413.....9419.....9421.....9431.....9433.....9437
.....9439.....9461.....9463.....9467.....9473.....9479.....9491.....9497
.....9511.....9521.....9533.....9539.....9547.....9551.....9587.....9601
.....9613.....9619.....9623.....9629.....9631.....9643.....9649.....9661
.....9677.....9679.....9689.....9697.....9719.....9721.....9733.....9739
.....9743.....9749.....9767.....9769.....9781.....9787.....9791.....9803
.....9811.....9817.....9829.....9833.....9839.....9851.....9857.....9859
.....9871.....9883.....9887.....9901.....9907.....9923.....9929.....9931
.....9941.....9949.....9967.....9973....10007....10009....10037....10039
....10061....10067....10069....10079....10091....10093....10099....10103
....10111....10133....10139....10141....10151....10159....10163....10169
....10177....10181....10193....10211....10223....10243....10247....10253
....10259....10267....10271....10273....10289....10301....10303....10313
....10321....10331....10333....10337....10343....10357....10369....10391
....10399....10427....10429....10433....10453....10457....10459....10463
....10477....10487....10499....10501....10513....10529....10531....10559
....10567....10589....10597....10601....10607....10613....10627....10631
....10639....10651....10657....10663....10667....10687....10691....10709
....10711....10723....10729....10733....10739....10753....10771....10781
....10789....10799....10831....10837....10847....10853....10859....10861
....10867....10883....10889....10891....10903....10909....10937....10939
....10949....10957....10973....10979....10987....10993....11003....11027
....11047....11057....11059....11069....11071....11083....11087....11093
....11113....11117....11119....11131....11149....11159....11161....11171
....11173....11177....11197....11213....11239....11243....11251....11257
....11261....11273....11279....11287....11299....11311....11317....11321
....11329....11351....11353....11369....11383....11393....11399....11411
....11423....11437....11443....11447....11467....11471....11483....11489
....11491....11497....11503....11519....11527....11549....11551....11579
....11587....11593....11597....11617....11621....11633....11657....11677
....11681....11689....11699....11701....11717....11719....11731....11743
....11777....11779....11783....11789....11801....11807....11813....11821
....11827....11831....11833....11839....11863....11867....11887....11897
....11903....11909....11923....11927....11933....11939....11941....11953
....11959....11969....11971....11981....11987....12007....12011....12037
....12041....12043....12049....12071....12073....12097....12101....12107
....12109....12113....12119....12143....12149....12157....12161....12163
....12197....12203....12211....12227....12239....12241....12251....12253
....12263....12269....12277....12281....12289....12301....12323....12329
12343....12347....12373....12377....12379....12391....12401....12409
....12413....12421....12433....12437....12451....12457....12473....12479
....12487....12491....12497....12503....12511....12517....12527....12539
....12541....12547....12553....12569....12577....12583....12589....12601
....12611....12613....12619....12637....12641....12647....12653....12659
....12671....12689....12697....12703....12713....12721....12739....12743

....12757....12763....12781....12791....12799....12809....12821....12823
....12829....12841....12853....12889....12893....12899....12907....12911
....12917....12919....12923....12941....12953....12959....12967....12973
....12979....12983....13001....13003....13007....13009....13033....13037
....13043....13049....13063....13093....13099....13103....13109....13121
....13127....13147....13151....13159....13163....13171....13177....13183
....13187....13217....13219....13229....13241....13249....13259....13267
....13291....13297....13309....13313....13327....13331....13337....13339
....13367....13381....13397....13399....13411....13417....13421....13441
....13451....13457....13463....13469....13477....13487....13499....13513
....13523....13537....13553....13567....13577....13591....13597....13613
....13619....13627....13633....13649....13669....13679....13681....13687
....13691....13693....13697....13709....13711....13721....13723....13729
....13751....13757....13759....13763....13781....13789....13799....13807
....13829....13831....13841....13859....13873....13877....13879....13883
....13901....13903....13907....13913....13921....13931....13933....13963
....13967....13997....13999....14009....14011....14029....14033....14051
....14057....14071....14081....14083....14087....14107....14143....14149
....14153....14159....14173....14177....14197....14207....14221....14243
....14249....14251....14281....14293....14303....14321....14323....14327
....14341....14347....14369....14387....14389....14401....14407....14411
....14419....14423....14431....14437....14447....14449....14461....14479
....14489....14503....14519....14533....14537....14543....14549....14551
....14557....14561....14563....14591....14593....14621....14627....14629
....14633....14639....14653....14657....14669....14683....14699....14713
....14717....14723....14731....14737....14741....14747....14753....14759
....14767....14771....14779....14783....14797....14813....14821....14827
....14831....14843....14851....14867....14869....14879....14887....14891
....14897....14923....14929....14939....14947....14951....14957....14969
....14983....15013....15017....15031....15053....15061....15073....15077
....15083....15091....15101....15107....15121....15131....15137....15139
....15149....15161....15173....15187....15193....15199....15217....15227
....15233....15241....15259....15263....15269....15271....15277....15287
....15289....15299....15307....15313....15319....15329....15331....15349
....15359....15361....15373....15377....15383....15391....15401....15413
....15427....15439....15443....15451....15461....15467....15473....15493
....15497....15511....15527....15541....15551....15559....15569....15581
....15583....15601....15607....15619....15629....15641....15643....15647
....15649....15661....15667....15671....15679....15683....15727....15731
....15733....15737....15739....15749....15761....15767....15773....15787
....15791....15797....15803....15809....15817....15823....15859....15877
....15881....15887....15889....15901....15907....15913....15919....15923
....15937....15959....15971....15973....15991....16001....16007....16033
....16057....16061....16063....16067....16069....16073....16087....16091
....16097....16103....16111....16127....16139....16141....16183....16187
....16189....16193....16217....16223....16229....16231....16249....16253
....16267....16273....16301....16319....16333....16339....16349....16361
....16363....16369....16381....16411....16417....16421....16427....16433
....16447....16451....16453....16477....16481....16487....16493....16519
....16529....16547....16553....16561....16567....16573....16603....16607
....16619....16631....16633....16649....16651....16657....16661....16673
....16691....16693....16699....16703....16729....16741....16747....16759
....16763....16787....16811....16823....16829....16831....16843....16871
....16879....16883....16889....16901....16903....16921....16927....16931
....16937....16943....16963....16979....16981....16987....16993....17011
....17021....17027....17029....17033....17041....17047....17053....17077
....17093....17099....17107....17117....17123....17137....17159....17167
....17183....17189....17191....17203....17207....17209....17231....17239
....17257....17291....17293....17299....17317....17321....17327....17333
....17341....17351....17359....17377....17383....17387....17389....17393
....17401....17417....17419....17431....17443....17449....17467....17471
....17477....17483....17489....17491....17497....17509....17519....17539
....17551....17569....17573....17579....17581....17597....17599....17609
....17623....17627....17657....17659....17669....17681....17683....17707

....17713....17729....17737....17747....17749....17761....17783....17789
....17791....17807....17827....17837....17839....17851....17863....17881
....17891....17903....17909....17911....17921....17923....17929....17939
....17957....17959....17971....17977....17981....17987....17989....18013
....18041....18043....18047....18049....18059....18061....18077....18089
....18097....18119....18121....18127....18131....18133....18143....18149
....18169....18181....18191....18199....18211....18217....18223....18229
....18233....18251....18253....18257....18269....18287....18289....18301
....18307....18311....18313....18329....18341....18353....18367....18371
....18379....18397....18401....18413....18427....18433....18439....18443
....18451....18457....18461....18481....18493....18503....18517....18521
....18523....18539....18541....18553....18583....18587....18593....18617
....18637....18661....18671....18679....18691....18701....18713....18719
....18731....18743....18749....18757....18773....18787....18793....18797
....18803....18839....18859....18869....18899....18911....18913....18917
....18919....18947....18959....18973....18979....19001....19009....19013
....19031....19037....19051....19069....19073....19079....19081....19087
....19121....19139....19141....19157....19163....19181....19183....19207
....19211....19213....19219....19231....19237....19249....19259....19267
....19273....19289....19301....19309....19319....19333....19373....19379
....19381....19387....19391....19403....19417....19421....19423....19427
....19429....19433....19441....19447....19457....19463....19469....19471
....19477....19483....19489....19501....19507....19531....19541....19543
....19553....19559....19571....19577....19583....19597....19603....19609
....19661....19681....19687....19697....19699....19709....19717....19727
....19739....19751....19753....19759....19763....19777....19793....19801
....19813....19819....19841....19843....19853....19861....19867....19889
....19891....19913....19919....19927....19937....19949....19961....19963
....19973....19979....19991....19993....19997....20011....20021....20023
....20029....20047....20051....20063....20071....20089....20101....20107
....20113....20117....20123....20129....20143....20147....20149....20161
....20173....20177....20183....20201....20219....20231....20233....20249
....20261....20269....20287....20297....20323....20327....20333....20341
....20347....20353....20357....20359....20369....20389....20393....20399
....20407....20411....20431....20441....20443....20477....20479....20483
....20507....20509....20521....20533....20543....20549....20551....20563
....20593....20599....20611....20627....20639....20641....20663....20681
....20693....20707....20717....20719....20731....20743....20747....20749
....20753....20759....20771....20773....20789....20807....20809....20849
....20857....20873....20879....20887....20897....20899....20903....20921
....20929....20939....20947....20959....20963....20981....20983....21001
....21011....21013....21017....21019....21023....21031....21059....21061
....21067....21089....21101....21107....21121....21139....21143....21149
....21157....21163....21169....21179....21187....21191....21193....21211
....21221....21227....21247....21269....21277....21283....21313....21317
....21319....21323....21341....21347....21377....21379....21383....21391
....21397....21401....21407....21419....21433....21467....21481....21487
....21491....21493....21499....21503....21517....21521....21523....21529
....21557....21559....21563....21569....21577....21587....21589....21599
....21601....21611....21613....21617....21647....21649....21661....21673
....21683....21701....21713....21727....21737....21739....21751....21757
....21767....21773....21787....21799....21803....21817....21821....21839
....21841....21851....21859....21863....21871....21881....21893....21911
....21929....21937....21943....21961....21977....21991....21997....22003
....22013....22027....22031....22037....22039....22051....22063....22067
....22073....22079....22091....22093....22109....22111....22123....22129
....22133....22147....22153....22157....22159....22171....22189....22193
....22229....22247....22259....22271....22273....22277....22279....22283
....22291....22303....22307....22343....22349....22367....22369....22381
....22391....22397....22409....22433....22441....22447....22453....22469
....22481....22483....22501....22511....22531....22541....22543....22549
....22567....22571....22573....22613....22619....22621....22637....22639
....22643....22651....22669....22679....22691....22697....22699....22709
....22717....22721....22727....22739....22741....22751....22769....22777

....22783....22787....22807....22811....22817....22853....22859....22861
....22871....22877....22901....22907....22921....22937....22943....22961
....22963....22973....22993....23003....23011....23017....23021....23027
....23029....23039....23041....23053....23057....23059....23063....23071
....23081....23087....23099....23117....23131....23143....23159....23167
....23173....23189....23197....23201....23203....23209....23227....23251
....23269....23279....23291....23293....23297....23311....23321....23327
....23333....23339....23357....23369....23371....23399....23417....23431
....23447....23459....23473....23497....23509....23531....23537....23539
....23549....23557....23561....23563....23567....23581....23593....23599
....23603....23609....23623....23627....23629....23633....23663....23669
....23671....23677....23687....23689....23719....23741....23743....23747
....23753....23761....23767....23773....23789....23801....23813....23819
....23827....23831....23833....23857....23869....23873....23879....23887
....23893....23899....23909....23911....23917....23929....23957....23971
....23977....23981....23993....24001....24007....24019....24023....24029
....24043....24049....24061....24071....24077....24083....24091....24097
....24103....24107....24109....24113....24121....24133....24137....24151
....24169....24179....24181....24197....24203....24223....24229....24239
....24247....24251....24281....24317....24329....24337....24359....24371
....24373....24379....24391....24407....24413....24419....24421....24439
....24443....24469....24473....24481....24499....24509....24517....24527
....24533....24547....24551....24571....24593....24611....24623....24631
....24659....24671....24677....24683....24691....24697....24709....24733
....24749....24763....24767....24781....24793....24799....24809....24821
....24841....24847....24851....24859....24877....24889....24907....24917
....24919....24923....24943....24953....24967....24971....24977....24979
....24989....25013....25031....25033....25037....25057....25073....25087
....25097....25111....25117....25121....25127....25147....25153....25163
....25169....25171....25183....25189....25219....25229....25237....25243
....25247....25253....25261....25301....25303....25307....25309....25321
....25339....25343....25349....25357....25367....25373....25391....25409
....25411....25423....25439....25447....25453....25457....25463....25469
....25471....25523....25537....25541....25561....25577....25579....25583
....25589....25601....25603....25609....25621....25633....25639....25643
....25657....25667....25673....25679....25693....25703....25717....25733
....25741....25747....25759....25763....25771....25793....25799....25801
....25819....25841....25847....25849....25867....25873....25889....25903
....25913....25919....25931....25933....25939....25943....25951....25969
....25981....25997....25999....26003....26017....26021....26029....26041
....26053....26083....26099....26107....26111....26113....26119....26141
....26153....26161....26171....26177....26183....26189....26203....26209
....26227....26237....26249....26251....26261....26263....26267....26293
....26297....26309....26317....26321....26339....26347....26357....26371
....26387....26393....26399....26407....26417....26423....26431....26437
....26449....26459....26479....26489....26497....26501....26513....26539
....26557....26561....26573....26591....26597....26627....26633....26641
....26647....26669....26681....26683....26687....26693....26699....26701
....26711....26713....26717....26723....26729....26731....26737....26759
....26777....26783....26801....26813....26821....26833....26839....26849
....26861....26863....26879....26881....26891....26893....26903....26921
....26927....26947....26951....26953....26959....26981....26987....26993
....27011....27017....27031....27043....27059....27061....27067....27073
....27077....27091....27103....27107....27109....27127....27143....27179
....27191....27197....27211....27239....27241....27253....27259....27271
....27277....27281....27283....27299....27329....27337....27361....27367
....27397....27407....27409....27427....27431....27437....27449....27457
....27479....27481....27487....27509....27527....27529....27539....27541
....27551....27581....27583....27611....27617....27631....27647....27653
....27673....27689....27691....27697....27701....27733....27737....27739
....27743....27749....27751....27763....27767....27773....27779....27791
....27793....27799....27803....27809....27817....27823....27827....27847
....27851....27883....27893....27901....27917....27919....27941....27943
....27947....27953....27961....27967....27983....27997....28001....28019

....28027....28031....28051....28057....28069....28081....28087....28097
....28099....28109....28111....28123....28151....28163....28181....28183
....28201....28211....28219....28229....28277....28279....28283....28289
....28297....28307....28309....28319....28349....28351....28387....28393
....28403....28409....28411....28429....28433....28439....28447....28463
....28477....28493....28499....28513....28517....28537....28541....28547
....28549....28559....28571....28573....28579....28591....28597....28603
....28607....28619....28621....28627....28631....28643....28649....28657
....28661....28663....28669....28687....28697....28703....28711....28723
....28729....28751....28753....28759....28771....28789....28793....28807
....28813....28817....28837....28843....28859....28867....28871....28879
....28901....28909....28921....28927....28933....28949....28961....28979
....29009....29017....29021....29023....29027....29033....29059....29063
....29077....29101....29123....29129....29131....29137....29147....29153
....29167....29173....29179....29191....29201....29207....29209....29221
....29231....29243....29251....29269....29287....29297....29303....29311
....29327....29333....29339....29347....29363....29383....29387....29389
....29399....29401....29411....29423....29429....29437....29443....29453
....29473....29483....29501....29527....29531....29537....29567....29569
....29573....29581....29587....29599....29611....29629....29633....29641
....29663....29669....29671....29683....29717....29723....29741....29753
....29759....29761....29789....29803....29819....29833....29837....29851
....29863....29867....29873....29879....29881....29917....29921....29927
....29947....29959....29983....29989....30011....30013....30029....30047
....30059....30071....30089....30091....30097....30103....30109....30113
....30119....30133....30137....30139....30161....30169....30181....30187
....30197....30203....30211....30223....30241....30253....30259....30269
....30271....30293....30307....30313....30319....30323....30341....30347
....30367....30389....30391....30403....30427....30431....30449....30467
....30469....30491....30493....30497....30509....30517....30529....30539
....30553....30557....30559....30577....30593....30631....30637....30643
....30649....30661....30671....30677....30689....30697....30703....30707
....30713....30727....30757....30763....30773....30781....30803....30809
....30817....30829....30839....30841....30851....30853....30859....30869
....30871....30881....30893....30911....30931....30937....30941....30949
....30971....30977....30983....31013....31019....31033....31039....31051
....31063....31069....31079....31081....31091....31121....31123....31139
....31147....31151....31153....31159....31177....31181....31183....31189
....31193....31219....31223....31231....31237....31247....31249....31253
....31259....31267....31271....31277....31307....31319....31321....31327
....31333....31337....31357....31379....31387....31391....31393....31397
....31469....31477....31481....31489....31511....31513....31517....31531
....31541....31543....31547....31567....31573....31583....31601....31607
....31627....31643....31649....31657....31663....31667....31687....31699
....31721....31723....31727....31729....31741....31751....31769....31771
....31793....31799....31817....31847....31849....31859....31873....31883
....31891....31907....31957....31963....31973....31981....31991....32003
....32009....32027....32029....32051....32057....32059....32063....32069
....32077....32083....32089....32099....32117....32119....32141....32143
....32159....32173....32183....32189....32191....32203....32213....32233
....32237....32251....32257....32261....32297....32299....32303....32309
....32321....32323....32327....32341....32353....32359....32363....32369
....32371....32377....32381....32401....32411....32413....32423....32429
....32441....32443....32467....32479....32491....32497....32503....32507
....32531....32533....32537....32561....32563....32569....32573....32579
....32587....32603....32609....32611....32621....32633....32647....32653
....32687....32693....32707....32713....32717....32719....32749....32771
....32779....32783....32789....32797....32801....32803....32831....32833
....32839....32843....32869....32887....32909....32911....32917....32933
....32939....32941....32957....32969....32971....32983....32987....32993
....32999....33013....33023....33029....33037....33049....33053....33071
....33073....33083....33091....33107....33113....33119....33149....33151
....33161....33179....33181....33191....33199....33203....33211....33223
....33247....33287....33289....33301....33311....33317....33329....33331

....33343....33347....33349....33353....33359....33377....33391....33403
....33409....33413....33427....33457....33461....33469....33479....33487
....33493....33503....33521....33529....33533....33547....33563....33569
....33577....33581....33587....33589....33599....33601....33613....33617
....33619....33623....33629....33637....33641....33647....33679....33703
....33713....33721....33739....33749....33751....33757....33767....33769
....33773....33791....33797....33809....33811....33827....33829....33851
....33857....33863....33871....33889....33893....33911....33923....33931
....33937....33941....33961....33967....33997....34019....34031....34033
....34039....34057....34061....34123....34127....34129....34141....34147
....34157....34159....34171....34183....34211....34213....34217....34231
....34253....34259....34261....34267....34273....34283....34297....34301
....34303....34313....34319....34327....34337....34351....34361....34367
....34369....34381....34403....34421....34429....34439....34457....34469
....34471....34483....34487....34499....34501....34511....34513....34519
....34537....34543....34549....34583....34589....34591....34603....34607
....34613....34631....34649....34651....34667....34673....34679....34687
....34693....34703....34721....34729....34739....34747....34757....34759
....34763....34781....34807....34819....34841....34843....34847....34849
....34871....34877....34883....34897....34913....34919....34939....34949
....34961....34963....34981....35023....35027....35051....35053....35059
....35069....35081....35083....35089....35099....35107....35111....35117
....35129....35141....35149....35153....35159....35171....35201....35221
....35227....35251....35257....35267....35279....35281....35291....35311
....35317....35323....35327....35339....35353....35363....35381....35393
....35401....35407....35419....35423....35437....35447....35449....35461
....35491....35507....35509....35521....35527....35531....35533....35537
....35543....35569....35573....35591....35593....35597....35603....35617
....35671....35677....35729....35731....35747....35753....35759....35771
....35797....35801....35803....35809....35831....35837....35839....35851
....35863....35869....35879....35897....35899....35911....35923....35933
....35951....35963....35969....35977....35983....35993....35999....36007
....36011....36013....36017....36037....36061....36067....36073....36083
....36097....36107....36109....36131....36137....36151....36161....36187
....36191....36209....36217....36229....36241....36251....36263....36269
....36277....36293....36299....36307....36313....36319....36341....36343
....36353....36373....36383....36389....36433....36451....36457....36467
....36469....36473....36479....36493....36497....36523....36527....36529
....36541....36551....36559....36563....36571....36583....36587....36599
....36607....36629....36637....36643....36653....36671....36677....36683
....36691....36697....36709....36713....36721....36739....36749....36761
....36767....36779....36781....36787....36791....36793....36809....36821
....36833....36847....36857....36871....36877....36887....36899....36901
....36913....36919....36923....36929....36931....36943....36947....36973
....36979....36997....37003....37013....37019....37021....37039....37049
....37057....37061....37087....37097....37117....37123....37139....37159
....37171....37181....37189....37199....37201....37217....37223....37243
....37253....37273....37277....37307....37309....37313....37321....37337
....37339....37357....37361....37363....37369....37379....37397....37409
....37423....37441....37447....37463....37483....37489....37493....37501
....37507....37511....37517....37529....37537....37547....37549....37561
....37567....37571....37573....37579....37589....37591....37607....37619
....37633....37643....37649....37657....37663....37691....37693....37699
....37717....37747....37781....37783....37799....37811....37813....37831
....37847....37853....37861....37871....37879....37889....37897....37907
....37951....37957....37963....37967....37987....37991....37993....37997
....38011....38039....38047....38053....38069....38083....38113....38119
....38149....38153....38167....38177....38183....38189....38197....38201
....38219....38231....38237....38239....38261....38273....38281....38287
....38299....38303....38317....38321....38327....38329....38333....38351
....38371....38377....38393....38431....38447....38449....38453....38459
....38461....38501....38543....38557....38561....38567....38569....38593
....38603....38609....38611....38629....38639....38651....38653....38669
....38671....38677....38693....38699....38707....38711....38713....38723

....38729....38737....38747....38749....38767....38783....38791....38803
....38821....38833....38839....38851....38861....38867....38873....38891
....38903....38917....38921....38923....38933....38953....38959....38971
....38977....38993....39019....39023....39041....39043....39047....39079
....39089....39097....39103....39107....39113....39119....39133....39139
....39157....39161....39163....39181....39191....39199....39209....39217
....39227....39229....39233....39239....39241....39251....39293....39301
....39313....39317....39323....39341....39343....39359....39367....39371
....39373....39383....39397....39409....39419....39439....39443....39451
....39461....39499....39503....39509....39511....39521....39541....39551
....39563....39569....39581....39607....39619....39623....39631....39659
....39667....39671....39679....39703....39709....39719....39727....39733
....39749....39761....39769....39779....39791....39799....39821....39827
....39829....39839....39841....39847....39857....39863....39869....39877
....39883....39887....39901....39929....39937....39953....39971....39979
....39983....39989....40009....40013....40031....40037....40039....40063
....40087....40093....40099....40111....40123....40127....40129....40151
....40153....40163....40169....40177....40189....40193....40213....40231
....40237....40241....40253....40277....40283....40289....40343....40351
....40357....40361....40387....40423....40427....40429....40433....40459
....40471....40483....40487....40493....40499....40507....40519....40529
....40531....40543....40559....40577....40583....40591....40597....40609
....40627....40637....40639....40693....40697....40699....40709....40739
....40751....40759....40763....40771....40787....40801....40813....40819
....40823....40829....40841....40847....40849....40853....40867....40879
....40883....40897....40903....40927....40933....40939....40949....40961
....40973....40993....41011....41017....41023....41039....41047....41051
....41057....41077....41081....41113....41117....41131....41141....41143
....41149....41161....41177....41179....41183....41189....41201....41203
....41213....41221....41227....41231....41233....41243....41257....41263
....41269....41281....41299....41333....41341....41351....41357....41381
....41387....41389....41399....41411....41413....41443....41453....41467
....41479....41491....41507....41513....41519....41521....41539....41543
....41549....41579....41593....41597....41603....41609....41611....41617
....41621....41627....41641....41647....41651....41659....41669....41681
....41687....41719....41729....41737....41759....41761....41771....41777
....41801....41809....41813....41843....41849....41851....41863....41879
....41887....41893....41897....41903....41911....41927....41941....41947
....41953....41957....41959....41969....41981....41983....41999....42013
....42017....42019....42023....42043....42061....42071....42073....42083
....42089....42101....42131....42139....42157....42169....42179....42181
....42187....42193....42197....42209....42221....42223....42227....42239
....42257....42281....42283....42293....42299....42307....42323....42331
....42337....42349....42359....42373....42379....42391....42397....42403
....42407....42409....42433....42437....42443....42451....42457....42461
....42463....42467....42473....42487....42491....42499....42509....42533
....42557....42569....42571....42577....42589....42611....42641....42643
....42649....42667....42677....42683....42689....42697....42701....42703
....42709....42719....42727....42737....42743....42751....42767....42773
....42787....42793....42797....42821....42829....42839....42841....42853
....42859....42863....42899....42901....42923....42929....42937....42943
....42953....42961....42967....42979....42989....43003....43013....43019
....43037....43049....43051....43063....43067....43093....43103....43117
....43133....43151....43159....43177....43189....43201....43207....43223
....43237....43261....43271....43283....43291....43313....43319....43321
....43331....43391....43397....43399....43403....43411....43427....43441
....43451....43457....43481....43487....43499....43517....43541....43543
....43573....43577....43579....43591....43597....43607....43609....43613
....43627....43633....43649....43651....43661....43669....43691....43711
....43717....43721....43753....43759....43777....43781....43783....43787
....43789....43793....43801....43853....43867....43889....43891....43913
....43933....43943....43951....43961....43963....43969....43973....43987
....43991....43997....44017....44021....44027....44029....44041....44053
....44059....44071....44087....44089....44101....44111....44119....44123

....44129....44131....44159....44171....44179....44189....44201....44203
....44207....44221....44249....44257....44263....44267....44269....44273
....44279....44281....44293....44351....44357....44371....44381....44383
....44389....44417....44449....44453....44483....44491....44497....44501
....44507....44519....44531....44533....44537....44543....44549....44563
....44579....44587....44617....44621....44623....44633....44641....44647
....44651....44657....44683....44687....44699....44701....44711....44729
....44741....44753....44771....44773....44777....44789....44797....44809
....44819....44839....44843....44851....44867....44879....44887....44893
....44909....44917....44927....44939....44953....44959....44963....44971
....44983....44987....45007....45013....45053....45061....45077....45083
....45119....45121....45127....45131....45137....45139....45161....45179
....45181....45191....45197....45233....45247....45259....45263....45281
....45289....45293....45307....45317....45319....45329....45337....45341
....45343....45361....45377....45389....45403....45413....45427....45433
....45439....45481....45491....45497....45503....45523....45533....45541
....45553....45557....45569....45587....45589....45599....45613....45631
....45641....45659....45667....45673....45677....45691....45697....45707
....45737....45751....45757....45763....45767....45779....45817....45821
....45823....45827....45833....45841....45853....45863....45869....45887
....45893....45943....45949....45953....45959....45971....45979....45989
....46021....46027....46049....46051....46061....46073....46091....46093
....46099....46103....46133....46141....46147....46153....46171....46181
....46183....46187....46199....46219....46229....46237....46261....46271
....46273....46279....46301....46307....46309....46327....46337....46349
....46351....46381....46399....46411....46439....46441....46447....46451
....46457....46471....46477....46489....46499....46507....46511....46523
....46549....46559....46567....46573....46589....46591....46601....46619
....46633....46639....46643....46649....46663....46679....46681....46687
....46691....46703....46723....46727....46747....46751....46757....46769
....46771....46807....46811....46817....46819....46829....46831....46853
....46861....46867....46877....46889....46901....46919....46933....46957
....46993....46997....47017....47041....47051....47057....47059....47087
....47093....47111....47119....47123....47129....47137....47143....47147
....47149....47161....47189....47207....47221....47237....47251....47269
....47279....47287....47293....47297....47303....47309....47317....47339
....47351....47353....47363....47381....47387....47389....47407....47417
....47419....47431....47441....47459....47491....47497....47501....47507
....47513....47521....47527....47533....47543....47563....47569....47581
....47591....47599....47609....47623....47629....47639....47653....47657
....47659....47681....47699....47701....47711....47713....47717....47737
....47741....47743....47777....47779....47791....47797....47807....47809
....47819....47837....47843....47857....47869....47881....47903....47911
....47917....47933....47939....47947....47951....47963....47969....47977
....47981....48017....48023....48029....48049....48073....48079....48091
....48109....48119....48121....48131....48157....48163....48179....48187
....48193....48197....48221....48239....48247....48259....48271....48281
....48299....48311....48313....48337....48341....48353....48371....48383
....48397....48407....48409....48413....48437....48449....48463....48473
....48479....48481....48487....48491....48497....48523....48527....48533
....48539....48541....48563....48571....48589....48593....48611....48619
....48623....48647....48649....48661....48673....48677....48679....48731
....48733....48751....48757....48761....48767....48779....48781....48787
....48799....48809....48817....48821....48823....48847....48857....48859
....48869....48871....48883....48889....48907....48947....48953....48973
....48989....48991....49003....49009....49019....49031....49033....49037
....49043....49057....49069....49081....49103....49109....49117....49121
....49123....49139....49157....49169....49171....49177....49193....49199
....49201....49207....49211....49223....49253....49261....49277....49279
....49297....49307....49331....49333....49339....49363....49367....49369
....49391....49393....49409....49411....49417....49429....49433....49451
....49459....49463....49477....49481....49499....49523....49529....49531
....49537....49547....49549....49559....49597....49603....49613....49627
....49633....49639....49663....49667....49669....49681....49697....49711

Anlage 1 Seite 11 von

....49727....49739....49741....49747....49757....49783....49787....49789
....49801....49807....49811....49823....49831....49843....49853....49871
....49877....49891....49919....49921....49927....49937....49939....49943
....49957....49991....49993....49999....50021....50023....50033....50047
....50051....50053....50069....50077....50087....50093....50101....50111
....50119....50123....50129....50131....50147....50153....50159....50177
....50207....50221....50227....50231....50261....50263....50273....50287
....50291....50311....50321....50329....50333....50341....50359....50363
....50377....50383....50387....50411....50417....50423....50441....50459
....50461....50497....50503....50513....50527....50539....50543....50549
....50551....50581....50587....50591....50593....50599....50627....50647
....50651....50671....50683....50707....50723....50741....50753....50767
....50773....50777....50789....50821....50833....50839....50849....50857
....50867....50873....50891....50893....50909....50923....50929....50951
....50957....50969....50971....50989....50993....51001....51031....51043
....51047....51059....51061....51071....51109....51131....51133....51137
....51151....51157....51169....51193....51197....51199....51203....51217
....51229....51239....51241....51257....51263....51283....51287....51307
....51329....51341....51343....51347....51349....51361....51383....51407
....51413....51419....51421....51427....51431....51437....51439....51449
....51461....51473....51479....51481....51487....51503....51511....51517
....51521....51539....51551....51563....51577....51581....51593....51599
....51607....51613....51631....51637....51647....51659....51673....51679
....51683....51691....51713....51719....51721....51749....51767....51769
....51787....51797....51803....51817....51827....51829....51839....51853
....51859....51869....51871....51893....51899....51907....51913....51929
....51941....51949....51971....51973....51977....51991....52009....52021
....52027....52051....52057....52067....52069....52081....52103....52121
....52127....52147....52153....52163....52177....52181....52183....52189
....52201....52223....52237....52249....52253....52259....52267....52289
....52291....52301....52313....52321....52361....52363....52369....52379
....52387....52391....52433....52453....52457....52489....52501....52511
....52517....52529....52541....52543....52553....52561....52567....52571
....52579....52583....52609....52627....52631....52639....52667....52673
....52691....52697....52709....52711....52721....52727....52733....52747
....52757....52769....52783....52807....52813....52817....52837....52859
....52861....52879....52883....52889....52901....52903....52919....52937
....52951....52957....52963....52967....52973....52981....52999....53003
....53017....53047....53051....53069....53077....53087....53089....53093
....53101....53113....53117....53129....53147....53149....53161....53171
....53173....53189....53197....53201....53231....53233....53239....53267
....53269....53279....53281....53299....53309....53323....53327....53353
....53359....53377....53381....53401....53407....53411....53419....53437
....53441....53453....53479....53503....53507....53527....53549....53551
....53569....53591....53593....53597....53609....53611....53617....53623
....53629....53633....53639....53653....53657....53681....53693....53699
....53717....53719....53731....53759....53773....53777....53783....53791
....53813....53819....53831....53849....53857....53861....53881....53887
....53891....53897....53899....53917....53923....53927....53939....53951
....53959....53987....53993....54001....54011....54013....54037....54049
....54059....54083....54091....54101....54121....54133....54139....54151
....54163....54167....54181....54193....54217....54251....54269....54277
....54287....54293....54311....54319....54323....54331....54347....54361
....54367....54371....54377....54401....54403....54409....54413....54419
....54421....54437....54443....54449....54469....54493....54497....54499
....54503....54517....54521....54539....54541....54547....54559....54563
....54577....54581....54583....54601....54617....54623....54629....54631
....54647....54667....54673....54679....54709....54713....54721....54727
....54751....54767....54773....54779....54787....54799....54829....54833
....54851....54869....54877....54881....54907....54917....54919....54941
....54949....54959....54973....54979....54983....55001....55009....55021
....55049....55051....55057....55061....55073....55079....55103....55109
....55117....55127....55147....55163....55171....55201....55207....55213
....55217....55219....55229....55243....55249....55259....55291....55313

....55331....55333....55337....55339....55343....55351....55373....55381
....55399....55411....55439....55441....55457....55469....55487....55501
....55511....55529....55541....55547....55579....55589....55603....55609
....55619....55621....55631....55633....55639....55661....55663....55667
....55673....55681....55691....55697....55711....55717....55721....55733
....55763....55787....55793....55799....55807....55813....55817....55819
....55823....55829....55837....55843....55849....55871....55889....55897
....55901....55903....55921....55927....55931....55933....55949....55967
....55987....55997....56003....56009....56039....56041....56053....56081
....56087....56093....56099....56101....56113....56123....56131....56149
....56167....56171....56179....56197....56207....56209....56237....56239
....56249....56263....56267....56269....56299....56311....56333....56359
....56369....56377....56383....56393....56401....56417....56431....56437
....56443....56453....56467....56473....56477....56479....56489....56501
....56503....56509....56519....56527....56531....56533....56543....56569
....56591....56597....56599....56611....56629....56633....56659....56663
....56671....56681....56687....56701....56711....56713....56731....56737
....56747....56767....56773....56779....56783....56807....56809....56813
....56821....56827....56843....56857....56873....56891....56893....56897
....56909....56911....56921....56923....56929....56941....56951....56957
....56963....56983....56989....56993....56999....57037....57041....57047
....57059....57073....57077....57089....57097....57107....57119....57131
....57139....57143....57149....57163....57173....57179....57191....57193
....57203....57221....57223....57241....57251....57259....57269....57271
....57283....57287....57301....57329....57331....57347....57349....57367
....57373....57383....57389....57397....57413....57427....57457....57467
....57487....57493....57503....57527....57529....57557....57559....57571
....57587....57593....57601....57637....57641....57649....57653....57667
....57679....57689....57697....57709....57713....57719....57727....57731
....57737....57751....57773....57781....57787....57791....57793....57803
....57809....57829....57839....57847....57853....57859....57881....57899
....57901....57917....57923....57943....57947....57973....57977....57991
....58013....58027....58031....58043....58049....58057....58061....58067
....58073....58099....58109....58111....58129....58147....58151....58153
....58169....58171....58189....58193....58199....58207....58211....58217
....58229....58231....58237....58243....58271....58309....58313....58321
....58337....58363....58367....58369....58379....58391....58393....58403
....58411....58417....58427....58439....58441....58451....58453....58477
....58481....58511....58537....58543....58549....58567....58573....58579
....58601....58603....58613....58631....58657....58661....58679....58687
....58693....58699....58711....58727....58733....58741....58757....58763
....58771....58787....58789....58831....58889....58897....58901....58907
....58909....58913....58921....58937....58943....58963....58967....58979
....58991....58997....59009....59011....59021....59023....59029....59051
....59053....59063....59069....59077....59083....59093....59107....59113
....59119....59123....59141....59149....59159....59167....59183....59197
....59207....59209....59219....59221....59233....59239....59243....59263
....59273....59281....59333....59341....59351....59357....59359....59369
....59377....59387....59393....59399....59407....59417....59419....59441
....59443....59447....59453....59467....59471....59473....59497....59509
....59513....59539....59557....59561....59567....59581....59611....59617
....59621....59627....59629....59651....59659....59663....59669....59671
....59693....59699....59707....59723....59729....59743....59747....59753
....59771....59779....59791....59797....59809....59833....59863....59879
....59887....59921....59929....59951....59957....59971....59981....59999
....60013....60017....60029....60037....60041....60077....60083....60089
....60091....60101....60103....60107....60127....60133....60139....60149
....60161....60167....60169....60209....60217....60223....60251....60257
....60259....60271....60289....60293....60317....60331....60337....60343
....60353....60373....60383....60397....60413....60427....60443....60449
....60457....60493....60497....60509....60521....60527....60539....60589
....60601....60607....60611....60617....60623....60631....60637....60647
....60649....60659....60661....60679....60689....60703....60719....60727
....60733....60737....60757....60761....60763....60773....60779....60793

....60811....60821....60859....60869....60887....60889....60899....60901
....60913....60917....60919....60923....60937....60943....60953....60961
....61001....61007....61027....61031....61043....61051....61057....61091
....61099....61121....61129....61141....61151....61153....61169....61211
....61223....61231....61253....61261....61283....61291....61297....61331
....61333....61339....61343....61357....61363....61379....61381....61403
....61409....61417....61441....61463....61469....61471....61483....61487
....61493....61507....61511....61519....61543....61547....61553....61559
....61561....61583....61603....61609....61613....61627....61631....61637
....61643....61651....61657....61667....61673....61681....61687....61703
....61717....61723....61729....61751....61757....61781....61813....61819
....61837....61843....61861....61871....61879....61909....61927....61933
....61949....61961....61967....61979....61981....61987....61991....62003
....62011....62017....62039....62047....62053....62057....62071....62081
....62099....62119....62129....62131....62137....62141....62143....62171
....62189....62191....62201....62207....62213....62219....62233....62273
....62297....62299....62303....62311....62323....62327....62347....62351
....62383....62401....62417....62423....62459....62467....62473....62477
....62483....62497....62501....62507....62533....62539....62549....62563
....62581....62591....62597....62603....62617....62627....62633....62639
....62653....62659....62683....62687....62701....62723....62731....62743
....62753....62761....62773....62791....62801....62819....62827....62851
....62861....62869....62873....62897....62903....62921....62927....62929
....62939....62969....62971....62981....62983....62987....62989....63029
....63031....63059....63067....63073....63079....63097....63103....63113
....63127....63131....63149....63179....63197....63199....63211....63241
....63247....63277....63281....63299....63311....63313....63317....63331
....63337....63347....63353....63361....63367....63377....63389....63391
....63397....63409....63419....63421....63439....63443....63463....63467
....63473....63487....63493....63499....63521....63527....63533....63541
....63559....63577....63587....63589....63599....63601....63607....63611
....63617....63629....63647....63649....63659....63667....63671....63689
....63691....63697....63703....63709....63719....63727....63737....63743
....63761....63773....63781....63793....63799....63803....63809....63823
....63839....63841....63853....63857....63863....63901....63907....63913
....63929....63949....63977....63997....64007....64013....64019....64033
....64037....64063....64067....64081....64091....64109....64123....64151
....64153....64157....64171....64187....64189....64217....64223....64231
....64237....64271....64279....64283....64301....64303....64319....64327
....64333....64373....64381....64399....64403....64433....64439....64451
....64453....64483....64489....64499....64513....64553....64567....64577
....64579....64591....64601....64609....64613....64621....64627....64633
....64661....64663....64667....64679....64693....64709....64717....64747
....64763....64781....64783....64793....64811....64817....64849....64853
....64871....64877....64879....64891....64901....64919....64921....64927
....64937....64951....64969....64997....65003....65011....65027....65029
....65033....65053....65063....65071....65089....65099....65101....65111
....65119....65123....65129....65141....65147....65167....65171....65173
....65179....65183....65203....65213....65239....65257....65267....65269
....65287....65293....65309....65323....65327....65353....65357....65371
....65381....65393....65407....65413....65419....65423....65437....65447
....65449....65479....65497....65519....65521....65537....65539....65543
....65551....65557....65563....65579....65581....65587....65599....65609
....65617....65629....65633....65647....65651....65657....65677....65687
....65699....65701....65707....65713....65717....65719....65729....65731
....65761....65777....65789....65809....65827....65831....65837....65839
....65843....65851....65867....65881....65899....65921....65927....65929
....65951....65957....65963....65981....65983....65993....66029....66037
....66041....66047....66067....66071....66083....66089....66103....66107
....66109....66137....66161....66169....66173....66179....66191....66221
....66239....66271....66293....66301....66337....66343....66347....66359
....66361....66373....66377....66383....66403....66413....66431....66449
....66457....66463....66467....66491....66499....66509....66523....66529
....66533....66541....66553....66569....66571....66587....66593....66601

....66617....66629....66643....66653....66683....66697....66701....66713
....66721....66733....66739....66749....66751....66763....66791....66797
....66809....66821....66841....66851....66853....66863....66877....66883
....66889....66919....66923....66931....66943....66947....66949....66959
....66973....66977....67003....67021....67033....67043....67049....67057
....67061....67073....67079....67103....67121....67129....67139....67141
....67153....67157....67169....67181....67187....67189....67211....67213
....67217....67219....67231....67247....67261....67271....67273....67289
....67307....67339....67343....67349....67369....67391....67399....67409
....67411....67421....67427....67429....67433....67447....67453....67477
....67481....67489....67493....67499....67511....67523....67531....67537
....67547....67559....67567....67577....67579....67589....67601....67607
....67619....67631....67651....67679....67699....67709....67723....67733
....67741....67751....67757....67759....67763....67777....67783....67789
....67801....67807....67819....67829....67843....67853....67867....67883
....67891....67901....67927....67931....67933....67939....67943....67957
....67961....67967....67979....67987....67993....68023....68041....68053
....68059....68071....68087....68099....68111....68113....68141....68147
....68161....68171....68207....68209....68213....68219....68227....68239
....68261....68279....68281....68311....68329....68351....68371....68389
....68399....68437....68443....68447....68449....68473....68477....68483
....68489....68491....68501....68507....68521....68531....68539....68543
....68567....68581....68597....68611....68633....68639....68659....68669
....68683....68687....68699....68711....68713....68729....68737....68743
....68749....68767....68771....68777....68791....68813....68819....68821
....68863....68879....68881....68891....68897....68899....68903....68909
....68917....68927....68947....68963....68993....69001....69011....69019
....69029....69031....69061....69067....69073....69109....69119....69127
....69143....69149....69151....69163....69191....69193....69197....69203
....69221....69233....69239....69247....69257....69259....69263....69313
....69317....69337....69341....69371....69379....69383....69389....69401
....69403....69427....69431....69439....69457....69463....69467....69473
....69481....69491....69493....69497....69499....69539....69557....69593
....69623....69653....69661....69677....69691....69697....69709....69737
....69739....69761....69763....69767....69779....69809....69821....69827
....69829....69833....69847....69857....69859....69877....69899....69911
....69929....69931....69941....69959....69991....69997....70001....70003
...70009...70019...70039...70051...70061...70067...70079...70099
...70111...70117...70121...70123...70139...70141...70157...70163
...70177...70181...70183...70199...70201...70207...70223...70229
...70237...70241...70249...70271...70289...70297...70309...70313
...70321...70327...70351...70373...70379...70381...70393...70423
...70429...70439...70451...70457...70459...70481...70487...70489
...70501...70507...70529...70537...70549...70571...70573...70583
...70589...70607...70619...70621...70627...70639...70657...70663
...70667...70687...70709...70717...70729...70753...70769...70783
...70793...70823...70841...70843...70849...70853...70867...70877
...70879...70891...70901...70913...70919...70921...70937...70949
...70951...70957...70969...70979...70981...70991...70997...70999
...71011...71023...71039...71059...71069...71081...71089...71119
...71129...71143...71147...71153...71161...71167...71171...71191
...71209...71233...71237...71249...71257...71261...71263...71287
...71293...71317...71327...71329...71333...71339...71341...71347
...71353...71359...71363...71387...71389...71399...71411...71413
...71419...71429...71437...71443...71453...71471...71473...71479
...71483...71503...71527...71537...71549...71551...71563...71569
...71593...71597...71633...71647...71663...71671...71693...71699
...71707...71711...71713...71719...71741...71761...71777...71789
...71807...71809...71821...71837...71843...71849...71861...71867
...71879...71881...71887...71899...71909...71917...71933...71941
...71947...71963...71971...71983...71987...71993...71999...72019
...72031...72043...72047...72053...72073...72077...72089...72091
...72101...72103...72109...72139...72161...72167...72169...72173
...72211...72221...72223...72227...72229...72251...72253...72269

....72271....72277....72287....72307....72313....72337....72341....72353
....72367....72379....72383....72421....72431....72461....72467....72469
....72481....72493....72497....72503....72533....72547....72551....72559
....72577....72613....72617....72623....72643....72647....72649....72661
....72671....72673....72679....72689....72701....72707....72719....72727
....72733....72739....72763....72767....72797....72817....72823....72859
....72869....72871....72883....72889....72893....72901....72907....72911
....72923....72931....72937....72949....72953....72959....72973....72977
....72997....73009....73013....73019....73037....73039....73043....73061
....73063....73079....73091....73121....73127....73133....73141....73181
....73189....73237....73243....73259....73277....73291....73303....73309
....73327....73331....73351....73361....73363....73369....73379....73387
....73417....73421....73433....73453....73459....73471....73477....73483
....73517....73523....73529....73547....73553....73561....73571....73583
....73589....73597....73607....73609....73613....73637....73643....73651
....73673....73679....73681....73693....73699....73709....73721....73727
....73751....73757....73771....73783....73819....73823....73847....73849
....73859....73867....73877....73883....73897....73907....73939....73943
....73951....73961....73973....73999....74017....74021....74027....74047
....74051....74071....74077....74093....74099....74101....74131....74143
....74149....74159....74161....74167....74177....74189....74197....74201
....74203....74209....74219....74231....74257....74279....74287....74293
....74297....74311....74317....74323....74353....74357....74363....74377
....74381....74383....74411....74413....74419....74441....74449....74453
....74471....74489....74507....74509....74521....74527....74531....74551
....74561....74567....74573....74587....74597....74609....74611....74623
....74653....74687....74699....74707....74713....74717....74719....74729
....74731....74747....74759....74761....74771....74779....74797....74821
....74827....74831....74843....74857....74861....74869....74873....74887
....74891....74897....74903....74923....74929....74933....74941....74959
....75011....75013....75017....75029....75037....75041....75079....75083
....75109....75133....75149....75161....75167....75169....75181....75193
....75209....75211....75217....75223....75227....75239....75253....75269
....75277....75289....75307....75323....75329....75337....75347....75353
....75367....75377....75389....75391....75401....75403....75407....75431
....75437....75479....75503....75511....75521....75527....75533....75539
....75541....75553....75557....75571....75577....75583....75611....75617
....75619....75629....75641....75653....75659....75679....75683....75689
....75703....75707....75709....75721....75731....75743....75767....75773
....75781....75787....75793....75797....75821....75833....75853....75869
....75883....75913....75931....75937....75941....75967....75979....75983
....75989....75991....75997....76001....76003....76031....76039....76079
....76081....76091....76099....76103....76123....76129....76147....76157
....76159....76163....76207....76213....76231....76243....76249....76253
....76259....76261....76283....76289....76303....76333....76343....76367
....76369....76379....76387....76403....76421....76423....76441....76463
....76471....76481....76487....76493....76507....76511....76519....76537
....76541....76543....76561....76579....76597....76603....76607....76631
....76649....76651....76667....76673....76679....76697....76717....76733
....76753....76757....76771....76777....76781....76801....76819....76829
....76831....76837....76847....76871....76873....76883....76907....76913
....76919....76943....76949....76961....76963....76991....77003....77017
....77023....77029....77041....77047....77069....77081....77093....77101
....77137....77141....77153....77167....77171....77191....77201....77213
....77237....77239....77243....77249....77261....77263....77267....77269
....77279....77291....77317....77323....77339....77347....77351....77359
....77369....77377....77383....77417....77419....77431....77447....77471
....77477....77479....77489....77491....77509....77513....77521....77527
....77543....77549....77551....77557....77563....77569....77573....77587
....77591....77611....77617....77621....77641....77647....77659....77681
....77687....77689....77699....77711....77713....77719....77723....77731
....77743....77747....77761....77773....77783....77797....77801....77813
....77839....77849....77863....77867....77893....77899....77929....77933
....77951....77969....77977....77983....77999....78007....78017....78031

....78041....78049....78059....78079....78101....78121....78137....78139
....78157....78163....78167....78173....78179....78191....78193....78203
....78229....78233....78241....78259....78277....78283....78301....78307
....78311....78317....78341....78347....78367....78401....78427....78437
....78439....78467....78479....78487....78497....78509....78511....78517
....78539....78541....78553....78569....78571....78577....78583....78593
....78607....78623....78643....78649....78653....78691....78697....78707
....78713....78721....78737....78779....78781....78787....78791....78797
....78803....78809....78823....78839....78853....78857....78877....78887
....78889....78893....78901....78919....78929....78941....78977....78979
....78989....79031....79039....79043....79063....79087....79103....79111
....79133....79139....79147....79151....79153....79159....79181....79187
....79193....79201....79229....79231....79241....79259....79273....79279
....79283....79301....79309....79319....79333....79337....79349....79357
....79367....79379....79393....79397....79399....79411....79423....79427
....79433....79451....79481....79493....79531....79537....79549....79559
....79561....79579....79589....79601....79609....79613....79621....79627
....79631....79633....79657....79669....79687....79691....79693....79697
....79699....79757....79769....79777....79801....79811....79813....79817
....79823....79829....79841....79843....79847....79861....79867....79873
....79889....79901....79903....79907....79939....79943....79967....79973
....79979....79987....79997....79999....80021....80039....80051....80071
....80077....80107....80111....80141....80147....80149....80153....80167
....80173....80177....80191....80207....80209....80221....80231....80233
....80239....80251....80263....80273....80279....80287....80309....80317
....80329....80341....80347....80363....80369....80387....80407....80429
....80447....80449....80471....80473....80489....80491....80513....80527
....80537....80557....80567....80599....80603....80611....80621....80627
....80629....80651....80657....80669....80671....80677....80681....80683
....80687....80701....80713....80737....80747....80749....80761....80777
....80779....80783....80789....80803....80809....80819....80831....80833
....80849....80863....80897....80909....80911....80917....80923....80929
....80933....80953....80963....80989....81001....81013....81017....81019
....81023....81031....81041....81043....81047....81049....81071....81077
....81083....81097....81101....81119....81131....81157....81163....81173
....81181....81197....81199....81203....81223....81233....81239....81281
....81283....81293....81299....81307....81331....81343....81349....81353
....81359....81371....81373....81401....81409....81421....81439....81457
....81463....81509....81517....81527....81533....81547....81551....81553
....81559....81563....81569....81611....81619....81629....81637....81647
....81649....81667....81671....81677....81689....81701....81703....81707
....81727....81737....81749....81761....81769....81773....81799....81817
....81839....81847....81853....81869....81883....81899....81901....81919
....81929....81931....81937....81943....81953....81967....81971....81973
....82003....82007....82009....82013....82021....82031....82037....82039
....82051....82067....82073....82129....82139....82141....82153....82163
....82171....82183....82189....82193....82207....82217....82219....82223
....82231....82237....82241....82261....82267....82279....82301....82307
....82339....82349....82351....82361....82373....82387....82393....82421
....82457....82463....82469....82471....82483....82487....82493....82499
....82507....82529....82531....82549....82559....82561....82567....82571
....82591....82601....82609....82613....82619....82633....82651....82657
....82699....82721....82723....82727....82729....82757....82759....82763
....82781....82787....82793....82799....82811....82813....82837....82847
....82883....82889....82891....82903....82913....82939....82963....82981
....82997....83003....83009....83023....83047....83059....83063....83071
....83077....83089....83093....83101....83117....83137....83177....83203
....83207....83219....83221....83227....83231....83233....83243....83257
....83267....83269....83273....83299....83311....83339....83341....83357
....83383....83389....83399....83401....83407....83417....83423....83431
....83437....83443....83449....83459....83471....83477....83497....83537
....83557....83561....83563....83579....83591....83597....83609....83617
....83621....83639....83641....83653....83663....83689....83701....83717
....83719....83737....83761....83773....83777....83791....83813....83833

....83843....83857....83869....83873....83891....83903....83911....83921
....83933....83939....83969....83983....83987....84011....84017....84047
....84053....84059....84061....84067....84089....84121....84127....84131
....84137....84143....84163....84179....84181....84191....84199....84211
....84221....84223....84229....84239....84247....84263....84299....84307
....84313....84317....84319....84347....84349....84377....84389....84391
....84401....84407....84421....84431....84437....84443....84449....84457
....84463....84467....84481....84499....84503....84509....84521....84523
....84533....84551....84559....84589....84629....84631....84649....84653
....84659....84673....84691....84697....84701....84713....84719....84731
....84737....84751....84761....84787....84793....84809....84811....84827
....84857....84859....84869....84871....84913....84919....84947....84961
....84967....84977....84979....84991....85009....85021....85027....85037
....85049....85061....85081....85087....85091....85093....85103....85109
....85121....85133....85147....85159....85193....85199....85201....85213
....85223....85229....85237....85243....85247....85259....85297....85303
....85313....85331....85333....85361....85363....85369....85381....85411
....85427....85429....85439....85447....85451....85453....85469....85487
....85513....85517....85523....85531....85549....85571....85577....85597
....85601....85607....85619....85621....85627....85639....85643....85661
....85667....85669....85691....85703....85711....85717....85733....85751
....85781....85793....85817....85819....85829....85831....85837....85843
....85847....85853....85889....85903....85909....85931....85933....85991
....85999....86011....86017....86027....86029....86069....86077....86083
....86111....86113....86117....86131....86137....86143....86161....86171
....86179....86183....86197....86201....86209....86239....86243....86249
....86257....86263....86269....86287....86291....86293....86297....86311
....86323....86341....86351....86353....86357....86369....86371....86381
....86389....86399....86413....86423....86441....86453....86461....86467
....86477....86491....86501....86509....86531....86533....86539....86561
....86573....86579....86587....86599....86627....86629....86677....86689
....86693....86711....86719....86729....86743....86753....86767....86771
....86783....86813....86837....86843....86851....86857....86861....86869
....86923....86927....86929....86939....86951....86959....86969....86981
....86993....87011....87013....87037....87041....87049....87071....87083
....87103....87107....87119....87121....87133....87149....87151....87179
....87181....87187....87211....87221....87223....87251....87253....87257
....87277....87281....87293....87299....87313....87317....87323....87337
....87359....87383....87403....87407....87421....87427....87433....87443
....87473....87481....87491....87509....87511....87517....87523....87539
....87541....87547....87553....87557....87559....87583....87587....87589
....87613....87623....87629....87631....87641....87643....87649....87671
....87679....87683....87691....87697....87701....87719....87721....87739
....87743....87751....87767....87793....87797....87803....87811....87833
....87853....87869....87877....87881....87887....87911....87917....87931
....87943....87959....87961....87973....87977....87991....88001....88003
....88007....88019....88037....88069....88079....88093....88117....88129
....88169....88177....88211....88223....88237....88241....88259....88261
....88289....88301....88321....88327....88337....88339....88379....88397
....88411....88423....88427....88463....88469....88471....88493....88499
....88513....88523....88547....88589....88591....88607....88609....88643
....88651....88657....88661....88663....88667....88681....88721....88729
....88741....88747....88771....88789....88793....88799....88801....88807
....88811....88813....88817....88819....88843....88853....88861....88867
....88873....88883....88897....88903....88919....88937....88951....88969
....88993....88997....89003....89009....89017....89021....89041....89051
....89057....89069....89071....89083....89087....89101....89107....89113
....89119....89123....89137....89153....89189....89203....89209....89213
....89227....89231....89237....89261....89269....89273....89293....89303
....89317....89329....89363....89371....89381....89387....89393....89399
....89413....89417....89431....89443....89449....89459....89477....89491
....89501....89513....89519....89521....89527....89533....89561....89563
....89567....89591....89597....89599....89603....89611....89627....89633
....89653....89657....89659....89669....89671....89681....89689....89753

....89759....89767....89779....89783....89797....89809....89819....89821
....89833....89839....89849....89867....89891....89897....89899....89909
....89917....89923....89939....89959....89963....89977....89983....89989
....90001....90007....90011....90017....90019....90023....90031....90053
....90059....90067....90071....90073....90089....90107....90121....90127
....90149....90163....90173....90187....90191....90197....90199....90203
....90217....90227....90239....90247....90263....90271....90281....90289
....90313....90353....90359....90371....90373....90379....90397....90401
....90403....90407....90437....90439....90469....90473....90481....90499
....90511....90523....90527....90529....90533....90547....90583....90599
....90617....90619....90631....90641....90647....90659....90677....90679
....90697....90703....90709....90731....90749....90787....90793....90803
....90821....90823....90833....90841....90847....90863....90887....90901
....90907....90911....90917....90931....90947....90971....90977....90989
....90997....91009....91019....91033....91079....91081....91097....91099
....91121....91127....91129....91139....91141....91151....91153....91159
....91163....91183....91193....91199....91229....91237....91243....91249
....91253....91283....91291....91297....91303....91309....91331....91367
....91369....91373....91381....91387....91393....91397....91411....91423
....91433....91453....91457....91459....91463....91493....91499....91513
....91529....91541....91571....91573....91577....91583....91591....91621
....91631....91639....91673....91691....91703....91711....91733....91753
....91757....91771....91781....91801....91807....91811....91813....91823
....91837....91841....91867....91873....91909....91921....91939....91943
....91951....91957....91961....91967....91969....91997....92003....92009
....92033....92041....92051....92077....92083....92107....92111....92119
....92143....92153....92173....92177....92179....92189....92203....92219
....92221....92227....92233....92237....92243....92251....92269....92297
....92311....92317....92333....92347....92353....92357....92363....92369
....92377....92381....92383....92387....92399....92401....92413....92419
....92431....92459....92461....92467....92479....92489....92503....92507
....92551....92557....92567....92569....92581....92593....92623....92627
....92639....92641....92647....92657....92669....92671....92681....92683
....92693....92699....92707....92717....92723....92737....92753....92761
....92767....92779....92789....92791....92801....92809....92821....92831
....92849....92857....92861....92863....92867....92893....92899....92921
....92927....92941....92951....92957....92959....92987....92993....93001
....93047....93053....93059....93077....93083....93089....93097....93103
....93113....93131....93133....93139....93151....93169....93179....93187
....93199....93229....93239....93241....93251....93253....93257....93263
....93281....93283....93287....93307....93319....93323....93329....93337
....93371....93377....93383....93407....93419....93427....93463....93479
....93481....93487....93491....93493....93497....93503....93523....93529
....93553....93557....93559....93563....93581....93601....93607....93629
....93637....93683....93701....93703....93719....93739....93761....93763
....93787....93809....93811....93827....93851....93871....93887....93889
....93893....93901....93911....93913....93923....93937....93941....93949
....93967....93971....93979....93983....93997....94007....94009....94033
....94049....94057....94063....94079....94099....94109....94111....94117
....94121....94151....94153....94169....94201....94207....94219....94229
....94253....94261....94273....94291....94307....94309....94321....94327
....94331....94343....94349....94351....94379....94397....94399....94421
....94427....94433....94439....94441....94447....94463....94477....94483
....94513....94529....94531....94541....94543....94547....94559....94561
....94573....94583....94597....94603....94613....94621....94649....94651
....94687....94693....94709....94723....94727....94747....94771....94777
....94781....94789....94793....94811....94819....94823....94837....94841
....94847....94849....94873....94889....94903....94907....94933....94949
....94951....94961....94993....94999....95003....95009....95021....95027
....95063....95071....95083....95087....95089....95093....95101....95107
....95111....95131....95143....95153....95177....95189....95191....95203
....95213....95219....95231....95233....95239....95257....95261....95267
....95273....95279....95287....95311....95317....95327....95339....95369
....95383....95393....95401....95413....95419....95429....95441....95443

....95461....95467....95471....95479....95483....95507....95527....95531
....95539....95549....95561....95569....95581....95597....95603....95617
....95621....95629....95633....95651....95701....95707....95713....95717
....95723....95731....95737....95747....95773....95783....95789....95791
....95801....95803....95813....95819....95857....95869....95873....95881
....95891....95911....95917....95923....95929....95947....95957....95959
....95971....95987....95989....96001....96013....96017....96043....96053
....96059....96079....96097....96137....96149....96157....96167....96179
....96181....96199....96211....96221....96223....96233....96259....96263
....96269....96281....96289....96293....96323....96329....96331....96337
....96353....96377....96401....96419....96431....96443....96451....96457
....96461....96469....96479....96487....96493....96497....96517....96527
....96553....96557....96581....96587....96589....96601....96643....96661
....96667....96671....96697....96703....96731....96737....96739....96749
....96757....96763....96769....96779....96787....96797....96799....96821
....96823....96827....96847....96851....96857....96893....96907....96911
....96931....96953....96959....96973....96979....96989....96997....97001
....97003....97007....97021....97039....97073....97081....97103....97117
....97127....97151....97157....97159....97169....97171....97177....97187
....97213....97231....97241....97259....97283....97301....97303....97327
....97367....97369....97373....97379....97381....97387....97397....97423
....97429....97441....97453....97459....97463....97499....97501....97511
....97523....97547....97549....97553....97561....97571....97577....97579
....97583....97607....97609....97613....97649....97651....97673....97687
....97711....97729....97771....97777....97787....97789....97813....97829
....97841....97843....97847....97849....97859....97861....97871....97879
....97883....97919....97927....97931....97943....97961....97967....97973
....97987....98009....98011....98017....98041....98047....98057....98081
....98101....98123....98129....98143....98179....98207....98213....98221
....98227....98251....98257....98269....98297....98299....98317....98321
....98323....98327....98347....98369....98377....98387....98389....98407
....98411....98419....98429....98443....98453....98459....98467....98473
....98479....98491....98507....98519....98533....98543....98561....98563
....98573....98597....98621....98627....98639....98641....98663....98669
....98689....98711....98713....98717....98729....98731....98737....98773
....98779....98801....98807....98809....98837....98849....98867....98869
....98873....98887....98893....98897....98899....98909....98911....98927
....98929....98939....98947....98953....98963....98981....98993....98999
....99013....99017....99023....99041....99053....99079....99083....99089
....99103....99109....99119....99131....99133....99137....99139....99149
....99173....99181....99191....99223....99233....99241....99251....99257
....99259....99277....99289....99317....99347....99349....99367....99371
....99377....99391....99397....99401....99409....99431....99439....99469
....99487....99497....99523....99527....99529....99551....99559....99563
....99571....99577....99581....99607....99611....99623....99643....99661
....99667....99679....99689....99707....99709....99713....99719....99721
....99733....99761....99767....99787....99793....99809....99817....99823
....99829....99833....99839....99859....99871....99877....99881....99901
....99907....99923....99929....99961....99971....99989....99991

Von 3 bis 100000 sind 9591 Primzahlen zu finden sowie 1224 Zwillinge mod 2

Anmerkungen:

1. Die Ordnungsfolge der ungeraden Primzahlen von 3 bis 99991 erweist sich für das Verständnis des in dieser Monographie deduzierten Beweises der Goldbach'schen Vermutung als wichtige Basis, auf die der Leser zurückgreifen kann. Die hier deduzierten - in der Primzahlforschung noch unbekannten Gesetze - finden bei Betrachtung dieser Anlage eine volle Bestätigung und werden in Anlage 20 vertieft und bestätig
2. Die Primzahlzwillinge mod 2 wurden in Anlage 1 schattiert markiert. Daraus ist ersichtlich, dass sie auch in einer verketteten Sequenz vorkommen können, ohne dass zwischen ihnen (unbeteiligte) Primzahlen auftreten. Dieser Aspekt wurde in der Primzahlforschung bisher nicht näher analysiert. Beachtlich ist des weiteren, dass in diesen Zwillingen stets nur Primzahlen mit den Endziffern: 1&3, 1&9, 7&9 eingehen können und dass eine dieser Primzahlen zu den Klassen: 1,2,3,4 - die andere aber - zu den Klassen: 5,6,7,8 gehören muss.

Anlage 2 7 gekoppelte Goldbach-Zerlegungen im Intervall: 0 bis 5*10^7

-

-
- ```
for i from 0 to 5*10^7 do: if isprime (i) and isprime (20!-i)and
isprime (30!-i)and isprime (40!-i)and isprime (50!-i)
and isprime (60!-i) then print ("i",i,20!-i,30!-i,40!-i,50!-i,60!-i)
   end_if: end_for:
```

```
"i", 18625661, 2432902008158014339, 265252859812191058636308461374339,

   815915283247897734345611269596115894271981374339,

   30414093201713378043612608166064768844377641568960511999981374339,

832098711274139014427634118322336438075417260636124595244927769640959999981374\
339

"i", 22343099, 2432902008154296901, 265252859812191058636308457656901,

   815915283247897734345611269596115894271977656901,

   30414093201713378043612608166064768844377641568960511999977656901,

832098711274139014427634118322336438075417260636124595244927769640959999977656\
901

"i", 26437973, 2432902008150202027, 265252859812191058636308453562027,

   815915283247897734345611269596115894271973562027,

   30414093201713378043612608166064768844377641568960511999973562027,

832098711274139014427634118322336438075417260636124595244927769640959999973562\
027

"i", 26473543, 2432902008150166457, 265252859812191058636308453526457,

   815915283247897734345611269596115894271973526457,

   30414093201713378043612608166064768844377641568960511999973526457,

832098711274139014427634118322336438075417260636124595244927769640959999973526\
457

"i", 39958493, 2432902008136681507, 265252859812191058636308440041507,

   815915283247897734345611269596115894271960041507,

   30414093201713378043612608166064768844377641568960511999960041507,

832098711274139014427634118322336438075417260636124595244927769640959999960041\
507

"i", 43929499, 2432902008132710501, 265252859812191058636308436070501,

   815915283247897734345611269596115894271956070501,

   30414093201713378043612608166064768844377641568960511999956070501,

832098711274139014427634118322336438075417260636124595244927769640959999956070\
501

"i", 48461767, 2432902008128178233, 265252859812191058636308431538233,

   815915283247897734345611269596115894271951538233,

   30414093201713378043612608166064768844377641568960511999951538233,

832098711274139014427634118322336438075417260636124595244927769640959999951538\
233
```

<u>Anmerkung:</u>
Die hier bestimmten "Primzahl-Wellen" bestehen aus Primzahlen die (zeilenweise) in den (großen) Abständen: (30!-20!), 40!-30!), (50!-40!), (69!-50!) auftreten.
```

Anlage 3 3 gekoppelte Anti-Goldbach-Zerlegungen im Intervall: 0 bis 5*10^7

-

•

```
• for i from 0 to 5*10^7 do: if isprime (i)and isprime (20!+i)and
  isprime (30!+i)and isprime (40!+i)and isprime (50!+i)
  and isprime (60!+i) then print ("i",i,20!+i,30!+i,40!+i,50!+i,60!+i)
    end_if: end_for:

"i", 2187971, 2432902008178827971, 265252859812191058636308482187971,

   815915283247897734345611269596115894272002187971,

   30414093201713378043612608166064768844377641568960512000002187971,

8320987112741390144276341183223364380754172606361245952449277696409600000002187\
971

"i", 4594817, 2432902008181234817, 265252859812191058636308484594817,

   815915283247897734345611269596115894272004594817,

   30414093201713378043612608166064768844377641568960512000004594817,

8320987112741390144276341183223364380754172606361245952449277696409600000004594\
817

"i", 14359591, 2432902008190999591, 265252859812191058636308494359591,

   815915283247897734345611269596115894272014359591,

   30414093201713378043612608166064768844377641568960512000014359591,

8320987112741390144276341183223364380754172606361245952449277696409600000014359\
591
```

•

<u>Anmerkung:</u>
Die hier bestimmten "Primzahl-Wellen" bestehen aus Primzahlen die (zeilenweise) in den (großen) Abständen: (30!-20!), 40!-30!), (50!-40!), (69!-50!) auftreten.

Anlage 4 3 Quasi-Goldbach-Zerlegungen im Intervall: 0 bis 2*10^7

•

•

```
• for i from 0 to 2*10^7 do: if isprime (20!-i)and isprime (30!-
  i)and isprime (40!-i)and isprime (50!-i)and isprime (60!-i)
  then print ("i",i,20!-i,30!-i,40!-i,50!-i,60!-i) end_if: end_for:

"i", 248663, 2432902008176391337, 265252859812191058636308479751337,
    815915283247897734345611269596115894271999751337,
    30414093201713378043612608166064768844377641568960511999999751337,
8320987112741390144276341183223364380754172606361245952449277696409599999999999751\
337
"i", 382799, 2432902008176257201, 265252859812191058636308479617201,
    815915283247897734345611269596115894271999617201,
    30414093201713378043612608166064768844377641568960511999999617201,
8320987112741390144276341183223364380754172606361245952449277696409599999999999617\
201
"i", 18625661, 2432902008158014339, 265252859812191058636308461374339,
    815915283247897734345611269596115894271981374339,
    30414093201713378043612608166064768844377641568960511999981374339,
8320987112741390144276341183223364380754172606361245952449277696409599999999981374\
339
```

• `ifactor(248663)`

$167 \cdot 1489$

• `ifactor(382799)`

$313 \cdot 1223$

• `ifactor(18625661)`

18625661

•

Anmerkung:

Die hier bestimmten "Primzahl-Wellen" bestehen aus Primzahlen die (zeilenweise) in den (großen) Abständen: (20!-10!), (30!-20!), (40!-30!) auftreten. Bei den "Quasi-Goldbach-Zerlegungen" können die i - Werte prime oder nicht prime Zahlen sein.

Anlage 5 7 Quasi-Anti-Goldbach-Zerlegungen im Intervall: 0 bis 5*10^7

-
-
- for i from 0 to 5*10^7 do: if isprime (20!+i)and isprime (30!+i)and isprime (40!+i)and isprime (50!+i)and isprime (60!+i) then print ("i,",i,20!+i,30!+i,40!+i,50!+i,60!+i) end_if: end_for:

```
"i,", 2187971, 2432902008178827971, 265252859812191058636308482187971,
   815915283247897734345611269596115894272002187971,
   30414093201713378043612608166064768844377641568960512000002187971,
8320987112741390144276341183223364380754172606361245952449277696409600000002187\
971
"i,", 4594817, 2432902008181234817, 265252859812191058636308484594817,
   815915283247897734345611269596115894272004594817,
   30414093201713378043612608166064768844377641568960512000004594817,
8320987112741390144276341183223364380754172606361245952449277696409600000004594\
817
"i,", 14359591, 2432902008190999591, 265252859812191058636308494359591,
   815915283247897734345611269596115894272014359591,
   30414093201713378043612608166064768844377641568960512000014359591,
8320987112741390144276341183223364380754172606361245952449277696409600000014359\
591
"i,", 24406721, 2432902008201046721, 265252859812191058636308504406721,
   815915283247897734345611269596115894272024406721,
   30414093201713378043612608166064768844377641568960512000024406721,
8320987112741390144276341183223364380754172606361245952449277696409600000024406\
721
"i,", 29601977, 2432902008206241977, 265252859812191058636308509601977,
   815915283247897734345611269596115894272029601977,
   30414093201713378043612608166064768844377641568960512000029601977,
8320987112741390144276341183223364380754172606361245952449277696409600000029601\
977
"i,", 37476811, 2432902008214116811, 265252859812191058636308517476811,
   815915283247897734345611269596115894272037476811,
   30414093201713378043612608166064768844377641568960512000037476811,
8320987112741390144276341183223364380754172606361245952449277696409600000037476\
811
"i,", 49096429, 2432902008225736429, 265252859812191058636308529096429,
   815915283247897734345611269596115894272049096429,
   30414093201713378043612608166064768844377641568960512000049096429,
8320987112741390144276341183223364380754172606361245952449277696409600000049096\
429
```

-
-

Anmerkung:
Die hier bestimmten "Primzahl-Wellen" bestehen aus Primzahlen die (zeilenweise) in den (großen) Abständen: (20!-10!), (30!-20!), (40!-30!) auftreten. Bei den "Quasi-Goldbach-Zerlegungen" können die i - Werte prime oder nicht prime Zahlen sein.

Anlage 6 11 Goldbach-Zerlegungen, vermischt mit 11 Anti-Goldbach-Zerlegungen im Intervall 0 bis 10^6

-
-
- for i from 0 to 10^6 do: if isprime (i)and isprime (10!+i)and isprime (20!+i)and isprime (30!-i)and isprime (40!-i) then print ("i",i,10!+i,20!+i,30!-i,40!-i) end_if: end_for:

```
"i", 103087, 3731887, 2432902008176743087, 265252859812191058636308479896913,
  815915283247897734345611269596115894271999896913
"i", 136711, 3765511, 2432902008176776711, 265252859812191058636308479863289,
  815915283247897734345611269596115894271999863289
"i", 286733, 3915533, 2432902008176926733, 265252859812191058636308479713267,
  815915283247897734345611269596115894271999713267
"i", 309521, 3938321, 2432902008176949521, 265252859812191058636308479690479,
  815915283247897734345611269596115894271999690479
"i", 371639, 4000439, 2432902008177011639, 265252859812191058636308479628361,
  815915283247897734345611269596115894271999628361
"i", 405611, 4034411, 2432902008177045611, 265252859812191058636308479594389,
  815915283247897734345611269596115894271999594389
"i", 418909, 4047709, 2432902008177058909, 265252859812191058636308479581091,
  815915283247897734345611269596115894271999581091
"i", 425869, 4054669, 2432902008177065869, 265252859812191058636308479574131,
  815915283247897734345611269596115894271999574131
"i", 725321, 4354121, 2432902008177365321, 265252859812191058636308479274679,
  815915283247897734345611269596115894271999274679
"i", 762791, 4391591, 2432902008177402791, 265252859812191058636308479237209,
  815915283247897734345611269596115894271999237209
"i", 841447, 4470247, 2432902008177481447, 265252859812191058636308479158553,
  815915283247897734345611269596115894271999158553
```

- forprise(20!-10!)

  forprise(2432902008173011200)

- forprise(30!-20!)

  forprise(265252859812188625734300303360000)

- forprise(40!-30!)

  forprise(815915283247897469092751457405057257963520000000)

<u>Anmerkung:</u> Die hier bestimmten "Primzahl-Wellen" bestehen aus Primzahlen die (zeilenweise) in gleichen Abständen auftreten.
Diese (sehr großen) Abstände ergeben sich aus den Differenzen: (20!-10!), (30!-20!), 40!-30!) - die in dieser Anlage ermittelt wurden.

Anlage 7

10 gekoppelte Goldbach-und Anti-Goldbach-Zerlegungen
- im Intervall: 0 bis 10^6
-
- `for i from 0 to 10^6 do: if isprime (i)and isprime (10!+i)and isprime (20!-i)and isprime (30!-i)and isprime (40!-i) then print ("i",i,10!+i,20!+i,30!-i,40!-i) end_if: end_for:`

```
"i", 103087, 3731887, 2432902008176743087, 265252859812191058636308479896913,
   815915283247897734345611269596115894271999896913
"i", 136711, 3765511, 2432902008176776711, 265252859812191058636308479863289,
   815915283247897734345611269596115894271999863289
"i", 286733, 3915533, 2432902008176926733, 265252859812191058636308479713267,
   815915283247897734345611269596115894271999713267
"i", 309521, 3938321, 2432902008176949521, 265252859812191058636308479690479,
   815915283247897734345611269596115894271999690479
"i", 371639, 4000439, 2432902008177011639, 265252859812191058636308479628361,
   815915283247897734345611269596115894271999628361
"i", 405611, 4034411, 2432902008177045611, 265252859812191058636308479594389,
   815915283247897734345611269596115894271999594389
"i", 418909, 4047709, 2432902008177058909, 265252859812191058636308479581091,
   815915283247897734345611269596115894271999581091
"i", 425869, 4054669, 2432902008177065869, 265252859812191058636308479574131,
   815915283247897734345611269596115894271999574131
"i", 725321, 4354121, 2432902008177365321, 265252859812191058636308479274679,
   815915283247897734345611269596115894271999274679
"i", 762791, 4391591, 2432902008177402791, 265252859812191058636308479237209,
   815915283247897734345611269596115894271999237209
"i", 841447, 4470247, 2432902008177481447, 265252859812191058636308479158553,
   815915283247897734345611269596115894271999158553
```

- `forprise(20!-10!)`

  forprise(2432902008173011200)

- `forprise(30!-20!)`

  forprise(265252859812188625734300303360000)

- `forprise(40!-30!)`

  forprise(815915283247897469092751457405057257963520000000)

Anmerkung:

Die hier bestimmten "Primzahl-Wellen" bestehen aus Primzahlen die (zeilenweise) in den (großen) Abständen: (20!-10!), (30!-20!), (40!-30!) auftreten.

Anlage 8 22 Quasi-Goldbach-Zerlegungen im Intervall 10! bis 0

-
-
- ```
for i from 0 to 10! do: if isprime (10!+12-i)and isprime (10!+14-
i)and isprime (10!+18-i)and isprime
(10!+20-i)and isprime (10!+30-i)and isprime (10!+32-i) then print
  ("i",i,10!+12-i,10!+14-i,10!+18-i,10!+20-i,
10!+30-i,10!+32-i) end_if: end_for:
```

```
"i", 212761, 3416051, 3416053, 3416057, 3416059, 3416069, 3416071
"i", 333271, 3295541, 3295543, 3295547, 3295549, 3295559, 3295561
"i", 413071, 3215741, 3215743, 3215747, 3215749, 3215759, 3215761
"i", 1032151, 2596661, 2596663, 2596667, 2596669, 2596679, 2596681
"i", 1170151, 2458661, 2458663, 2458667, 2458669, 2458679, 2458681
"i", 1451311, 2177501, 2177503, 2177507, 2177509, 2177519, 2177521
"i", 1460161, 2168651, 2168653, 2168657, 2168659, 2168669, 2168671
"i", 1469581, 2159231, 2159233, 2159237, 2159239, 2159249, 2159251
"i", 1880341, 1748471, 1748473, 1748477, 1748479, 1748489, 1748491
"i", 1909951, 1718861, 1718863, 1718867, 1718869, 1718879, 1718881
"i", 1950061, 1678751, 1678753, 1678757, 1678759, 1678769, 1678771
"i", 2035951, 1592861, 1592863, 1592867, 1592869, 1592879, 1592881
"i", 2299111, 1329701, 1329703, 1329707, 1329709, 1329719, 1329721
"i", 2473201, 1155611, 1155613, 1155617, 1155619, 1155629, 1155631
"i", 2560111, 1068701, 1068703, 1068707, 1068709, 1068719, 1068721
"i", 2626471, 1002341, 1002343, 1002347, 1002349, 1002359, 1002361
"i", 2965231, 663581, 663583, 663587, 663589, 663599, 663601
"i", 3236551, 392261, 392263, 392267, 392269, 392279, 392281
"i", 3463111, 165701, 165703, 165707, 165709, 165719, 165721
"i", 3530971, 97841, 97843, 97847, 97849, 97859, 97861
"i", 3610771, 18041, 18043, 18047, 18049, 18059, 18061
"i", 3628801, 11, 13, 17, 19, 29, 31
```

<u>Anmerkung:</u> Die hier bestimmten "Primzahl-Wellen" bestehen aus Zwillingen mod 2, die (zeilenweise) in gleichen Abständen auftreten und zu den Folgen: *Folge 1*, *Folge 2* und *Folge 3* gehören.
```

Anlage 9 22 Quasi-Anti- Goldbach - Zerlegungen im Intervall 0 bis 10!

-
-
-
- for i from 0 to 10! do: if isprime (10+i)and isprime (12+i)and isprime (16+i)and isprime (18+i)and isprime (28+i)and isprime (30+i) then print ("i",i,10+i,12+i,16+i,18+i,28+i,30+i) end_if: end_for:

```
"i", 1, 11, 13, 17, 19, 29, 31
"i", 18031, 18041, 18043, 18047, 18049, 18059, 18061
"i", 97831, 97841, 97843, 97847, 97849, 97859, 97861
"i", 165691, 165701, 165703, 165707, 165709, 165719, 165721
"i", 392251, 392261, 392263, 392267, 392269, 392279, 392281
"i", 663571, 663581, 663583, 663587, 663589, 663599, 663601
"i", 1002331, 1002341, 1002343, 1002347, 1002349, 1002359, 1002361
"i", 1068691, 1068701, 1068703, 1068707, 1068709, 1068719, 1068721
"i", 1155601, 1155611, 1155613, 1155617, 1155619, 1155629, 1155631
"i", 1329691, 1329701, 1329703, 1329707, 1329709, 1329719, 1329721
"i", 1592851, 1592861, 1592863, 1592867, 1592869, 1592879, 1592881
"i", 1678741, 1678751, 1678753, 1678757, 1678759, 1678769, 1678771
"i", 1718851, 1718861, 1718863, 1718867, 1718869, 1718879, 1718881
"i", 1748461, 1748471, 1748473, 1748477, 1748479, 1748489, 1748491
"i", 2159221, 2159231, 2159233, 2159237, 2159239, 2159249, 2159251
"i", 2168641, 2168651, 2168653, 2168657, 2168659, 2168669, 2168671
"i", 2177491, 2177501, 2177503, 2177507, 2177509, 2177519, 2177521
"i", 2458651, 2458661, 2458663, 2458667, 2458669, 2458679, 2458681
"i", 2596651, 2596661, 2596663, 2596667, 2596669, 2596679, 2596681
"i", 3215731, 3215741, 3215743, 3215747, 3215749, 3215759, 3215761
"i", 3295531, 3295541, 3295543, 3295547, 3295549, 3295559, 3295561
"i", 3416041, 3416051, 3416053, 3416057, 3416059, 3416069, 3416071
```

-

<u>Anmerkung:</u> Die hier bestimmten "Primzahl-Wellen" bestehen aus Zwillingen mod 2, die (zeilenweise) in gleichen Abständen auftreten und zu den Folgen: *Folge 1, Folge* 2 *und Folge* 3 gehören.
Ein Vergleich mit **Anlage 7** zeigt die gleiche Zahl (22) der Zerlegungen, die bei den Quasi-Goldbach-Zerlegungen auftreten.

Anlage 10 12 Goldbach-Zerlegungen im Intervall 10! bis 0

-
-
- for i from 0 to 10! do: if isprime (i) and isprime (10!+12-i)and isprime (10!+14-i)and isprime (10!+18-i)and isprime (10!+20-i)and isprime (10!+30-i)and isprime (10!+32-i) then print ("i",i,10!+12-i,10!+14-i,10!+18-i,10!+20-i, 10!+30-i,10!+32-i) end_if: end_for:

```
"i", 333271, 3295541, 3295543, 3295547, 3295549, 3295559, 3295561
"i", 413071, 3215741, 3215743, 3215747, 3215749, 3215759, 3215761
"i", 1032151, 2596661, 2596663, 2596667, 2596669, 2596679, 2596681
"i", 1460161, 2168651, 2168653, 2168657, 2168659, 2168669, 2168671
"i", 1469581, 2159231, 2159233, 2159237, 2159239, 2159249, 2159251
"i", 1880341, 1748471, 1748473, 1748477, 1748479, 1748489, 1748491
"i", 1909951, 1718861, 1718863, 1718867, 1718869, 1718879, 1718881
"i", 1950061, 1678751, 1678753, 1678757, 1678759, 1678769, 1678771
"i", 2299111, 1329701, 1329703, 1329707, 1329709, 1329719, 1329721
"i", 2626471, 1002341, 1002343, 1002347, 1002349, 1002359, 1002361
"i", 3530971, 97841, 97843, 97847, 97849, 97859, 97861
"i", 3610771, 18041, 18043, 18047, 18049, 18059, 18061
```

-

<u>Anmerkung:</u> Die hier bestimmten "Primzahl-Wellen" bestehen aus Zwillingen mod 2, die (zeilenweise) in gleichen Abständen auftreten und zu den Folgen: *Folge 1*, *Folge 2* und *Folge 3* gehören.
Die i-Werte sind (hier) ausschließlich prime Zahlen und bilden eine Teilmenge der Zerlegungen, die in **Anlage 7** abgeleitete wurden.

Anlage 11 3 Anti- Goldbach-Zerlegungen im Intervall 10! bis 0

-
-
-
- for i from 0 to 10! do: if isprime (i)and isprime (10+i)and isprime (12+i)and isprime (16+i)and isprime (18+i)and isprime (28+i)and isprime (30+i) then print ("i",i,10+i,12+i,16+i,18+i,28+i,30+i) end_if: end_for:

```
"i", 392251, 392261, 392263, 392267, 392269, 392279, 392281

"i", 663571, 663581, 663583, 663587, 663589, 663599, 663601

"i", 1155601, 1155611, 1155613, 1155617, 1155619, 1155629, 1155631
```

-

Anmerkung: Die hier bestimmten "Primzahl-Wellen" bestehen aus Zwillingen mod 2, die (zeilenweise) in gleichen Abständen auftreten und zu den Folgen: *Folge 1*, *Folge 2* und *Folge 3* gehören.
Die i-Werte sind (hier) ausschließlich prime Zahlen und bilden eine Teilmenge der Zerlegungen, die in **Anlage 8** abgeleitete wurden.

Anlage 12 62 (achtfach kombinierte) Anti-Goldbach-Zerlegungen im Intervall: 0 bis 10^8 - 5 Seiten

•

```
for i from 0 to 10^8 do: if isprime (i)and isprime (4!+i)and isprime
(5!+i)and isprime (6!+i)and isprime (7!+i)and isprime
(8!+i)and isprime (100!+i) then print
  ("i",i,4!+i,5!+i,6!+i,7!+i,8!+i,100!+i) end_if: end_for:
```

```
"i", 69739, 69763, 69859, 70459, 74779, 110059,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000000069739

"i", 124679, 124703, 124799, 125399, 129719, 164999,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000000124679

"i", 171559, 171583, 171679, 172279, 176599, 211879,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000000171559

"i", 217397, 217421, 217517, 218117, 222437, 257717,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000000217397

"i", 407249, 407273, 407369, 407969, 412289, 447569,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000000407249

"i", 585317, 585341, 585437, 586037, 590357, 625637,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000000585317

"i", 1874833, 1874857, 1874953, 1875553, 1879873, 1915153,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000001874833

"i", 6442697, 6442721, 6442817, 6443417, 6447737, 6483017,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000006442697

"i", 6626623, 6626647, 6626743, 6627343, 6631663, 6666943,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000006626623

"i", 7637093, 7637117, 7637213, 7637813, 7642133, 7677413,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000007637093

"i", 9365029, 9365053, 9365149, 9365749, 9370069, 9405349,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000009365029

"i", 9375767, 9375791, 9375887, 9376487, 9380807, 9416087,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000009375767

"i", 10028329, 10028353, 10028449, 10029049, 10033369, 10068649,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000010028329
```

"i", 11260237, 11260261, 11260357, 11260957, 11265277, 11300557,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000011260237

"i", 12068009, 12068033, 12068129, 12068729, 12073049, 12108329,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000012068009

"i", 12529403, 12529427, 12529523, 12530123, 12534443, 12569723,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000012529403

"i", 14755009, 14755033, 14755129, 14755729, 14760049, 14795329,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000014755009

"i", 15223889, 15223913, 15224009, 15224609, 15228929, 15264209,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000015223889

"i", 18200339, 18200363, 18200459, 18201059, 18205379, 18240659,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000018200339

"i", 18396149, 18396173, 18396269, 18396869, 18401189, 18436469,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000018396149

"i", 20161577, 20161601, 20161697, 20162297, 20166617, 20201897,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000020161577

"i", 20807383, 20807407, 20807503, 20808103, 20812423, 20847703,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000020807383

"i", 20967389, 20967413, 20967509, 20968109, 20972429, 21007709,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000020967389

"i", 21066323, 21066347, 21066443, 21067043, 21071363, 21106643,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000021066323

"i", 24370873, 24370897, 24370993, 24371593, 24375913, 24411193,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000024370873

"i", 25468847, 25468871, 25468967, 25469567, 25473887, 25509167,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000025468847

"i", 25831417, 25831441, 25831537, 25832137, 25836457, 25871737,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000025831417

"i", 25900607, 25900631, 25900727, 25901327, 25905647, 25940927,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000025900607

"i", 26167993, 26168017, 26168113, 26168713, 26173033, 26208313,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000026167993

"i", 29254657, 29254681, 29254777, 29255377, 29259697, 29294977,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000029254657

"i", 31249459, 31249483, 31249579, 31250179, 31254499, 31289779,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000031249459

"i", 33037007, 33037031, 33037127, 33037727, 33042047, 33077327,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000033037007

"i", 33637397, 33637421, 33637517, 33638117, 33642437, 33677717,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000033637397

"i", 33843449, 33843473, 33843569, 33844169, 33848489, 33883769,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000033843449

"i", 37395773, 37395797, 37395893, 37396493, 37400813, 37436093,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000037395773

"i", 39058477, 39058501, 39058597, 39059197, 39063517, 39098797,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000039058477

"i", 45095857, 45095881, 45095977, 45096577, 45100897, 45136177,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000045095857

"i", 47809897, 47809921, 47810017, 47810617, 47814937, 47850217,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000047809897

"i", 49331567, 49331591, 49331687, 49332287, 49336607, 49371887,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000049331567

"i", 57690467, 57690491, 57690587, 57691187, 57695507, 57730787,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000057690467

"i", 57814717, 57814741, 57814837, 57815437, 57819757, 57855037,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000057814717

"i", 58154693, 58154717, 58154813, 58155413, 58159733, 58195013,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000058154693

"i", 59872237, 59872261, 59872357, 59872957, 59877277, 59912557,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000059872237

"i", 59877793, 59877817, 59877913, 59878513, 59882833, 59918113,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000059877793

"i", 62004749, 62004773, 62004869, 62005469, 62009789, 62045069,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000062004749

"i", 62416049, 62416073, 62416169, 62416769, 62421089, 62456369,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000062416049

"i", 63161807, 63161831, 63161927, 63162527, 63166847, 63202127,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000063161807

"i", 68280733, 68280757, 68280853, 68281453, 68285773, 68321053,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000068280733

"i", 73118713, 73118737, 73118833, 73119433, 73123753, 73159033,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000073118713

"i", 74050919, 74050943, 74051039, 74051639, 74055959, 74091239,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000074050919

"i", 74800267, 74800291, 74800387, 74800987, 74805307, 74840587,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000074800267

"i", 74979193, 74979217, 74979313, 74979913, 74984233, 75019513,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000074979193

"i", 79232333, 79232357, 79232453, 79233053, 79237373, 79272653,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000079232333

"i", 82758343, 82758367, 82758463, 82759063, 82763383, 82798663,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000082758343

"i", 84185617, 84185641, 84185737, 84186337, 84190657, 84225937,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000084185617

"i", 84676853, 84676877, 84676973, 84677573, 84681893, 84717173,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000084676853

"i", 88674283, 88674307, 88674403, 88675003, 88679323, 88714603,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000088674283

"i", 89597567, 89597591, 89597687, 89598287, 89602607, 89637887,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000089597567

"i", 92361733, 92361757, 92361853, 92362453, 92366773, 92402053,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000092361733

"i", 93781823, 93781847, 93781943, 93782543, 93786863, 93822143,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000093781823

"i", 93881273, 93881297, 93881393, 93881993, 93886313, 93921593,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000093881273

"i", 95163617, 95163641, 95163737, 95164337, 95168657, 95203937,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000095163617

"i", 97962853, 97962877, 97962973, 97963573, 97967893, 98003173,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916864000000000000000097962853

•

<u>Anmerkung:</u>
Diese kombinierten Anti-Goldbach-Zerlegungen bilden "Primzahl - Wellen - Tsunamis", die sich in wachsenden Zahlen-Intervallen fortpflanzen, wobei die Abstände zwischen den Primzahlen: p_1 bis p_6 in allen 62 Zeilen konstant bleiben.

Anlage 13 36 (kombinierte) Goldbach - Zerlegungen im Intervall: 0 bis 10^8 - 4 Seiten ~~Seite 1 von 4~~

- ```
  for i from 0 to 10^8 do: if isprime (i)and isprime (12!-i)and
  isprime (13!-i)and isprime (14!-i)and isprime (15!-i)and
  isprime (16!-i)and isprime (100!-i) then print ("i",i,12!-i,13!-
  i,14!-i,15!-i,16!-i,100!-i) end_if: end_for:
  ```

```
"i", 4408861, 474592739, 6222611939, 87173882339, 1307669959139,

   20922785479139,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999995591139

"i", 4574441, 474427159, 6222446359, 87173716759, 1307669793559,

   20922785313559,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999995425559

"i", 4923953, 474077647, 6222096847, 87173367247, 1307669444047,

   20922784964047,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999995076047

"i", 5570927, 473430673, 6221449873, 87172720273, 1307668797073,

   20922784317073,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999994429073

"i", 6296707, 472704893, 6220724093, 87171994493, 1307668071293,

   20922783591293,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999993703293

"i", 6594109, 472407491, 6220426691, 87171697091, 1307667773891,

   20922783293891,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999993405891

"i", 7200811, 471800789, 6219819989, 87171090389, 1307667167189,

   20922782687189,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999992799189

"i", 12790577, 466211023, 6214230223, 87165500623, 1307661577423,

   20922777097423,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999987209423

"i", 15299041, 463702559, 6211721759, 87162992159, 1307659068959,

   20922774588959,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999984700959

"i", 15814333, 463187267, 6211206467, 87162476867, 1307658553667,

   20922774073667,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999984185667
```
```

"i", 24664609, 454336991, 6202356191, 87153626591, 1307649703391,

20922765223391,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999975335391

"i", 25198711, 453802889, 6201822089, 87153092489, 1307649169289,

20922764689289,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999974801289

"i", 25252831, 453748769, 6201767969, 87153038369, 1307649115169,

20922764635169,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999974747169

"i", 33238589, 445763011, 6193782211, 87145052611, 1307641129411,

20922756649411,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999966761411

"i", 35233507, 443768093, 6191787293, 87143057693, 1307639134493,

20922754654493,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999964766493

"i", 42905843, 436095757, 6184114957, 87135385357, 1307631462157,

20922746982157,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999957094157

"i", 43270991, 435730609, 6183749809, 87135020209, 1307631097009,

20922746617009,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999956729009

"i", 45934379, 433067221, 6181086421, 87132356821, 1307628433621,

20922743953621,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999954065621

"i", 46488443, 432513157, 6180532357, 87131802757, 1307627879557,

20922743399557,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999953511557

"i", 48789607, 430211993, 6178231193, 87129501593, 1307625578393,

20922741098393,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999951210393

"i", 49713959, 429287641, 6177306841, 87128577241, 1307624654041,

20922740174041,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999950286041

"i", 50706289, 428295311, 6176314511, 87127584911, 1307623661711,

20922739181711,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
560894146397615651828625369792082722375825118521091686399999999999999949293711

"i", 52159543, 426842057, 6174861257, 87126131657, 1307622208457,

20922737728457,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
560894146397615651828625369792082722375825118521091686399999999999999947840457

"i", 57959549, 421042051, 6169061251, 87120331651, 1307616408451,

20922731928451,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
560894146397615651828625369792082722375825118521091686399999999999999942040451

"i", 59622007, 419379593, 6167398793, 87118669193, 1307614745993,

20922730265993,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
560894146397615651828625369792082722375825118521091686399999999999999940377993

"i", 61604899, 417396701, 6165415901, 87116686301, 1307612763101,

20922728283101,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
560894146397615651828625369792082722375825118521091686399999999999999938395101

"i", 70520357, 408481243, 6156500443, 87107770843, 1307603847643,

20922719367643,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
560894146397615651828625369792082722375825118521091686399999999999999929479643

"i", 77121907, 401879693, 6149898893, 87101169293, 1307597246093,

20922712766093,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
560894146397615651828625369792082722375825118521091686399999999999999922878093

"i", 78286477, 400715123, 6148734323, 87100004723, 1307596081523,

20922711601523,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
560894146397615651828625369792082722375825118521091686399999999999999921713523

"i", 83483707, 395517893, 6143537093, 87094807493, 1307590884293,

20922706404293,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
560894146397615651828625369792082722375825118521091686399999999999999916516293

"i", 84365131, 394636469, 6142655669, 87093926069, 1307590002869,

20922705522869,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
560894146397615651828625369792082722375825118521091686399999999999999915634869

"i", 93945847, 385055753, 6133074953, 87084345353, 1307580422153,

20922695942153,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
560894146397615651828625369792082722375825118521091686399999999999999906054153

"i", 95608559, 383393041, 6131412241, 87082682641, 1307578759441,

20922694279441,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999999904391441

"i", 95644931, 383356669, 6131375869, 87082646269, 1307578723069,

20922694243069,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999999904355069

"i", 98770079, 380231521, 6128250721, 87079521121, 1307575597921,

20922691117921,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999999901229921

"i", 99562823, 379438777, 6127457977, 87078728377, 1307574805177,

20922690325177,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999999900437177

•

Anmerkung:
Diese kombinierten Goldbach- und Anti - Goldbach- Zerlegungen bilden "Primzahl - Wellen - Tsunamis", die sich in wachsenden Zahlen-Intervallen fortpflanzen, wobei die Abstände zwischen den Primzahlen: p_1 bis p_3 und p_4 bis p_6 in allen 36 Zeilen konstant bleiben. Die Abstände zwischen den Primzahlen: p_3 und p_4 wachsen hingegen zunehmend.

Anlage 14 107 (kombinierte) Anti - Goldbach - und Goldbach - Zerlegungen im Intervall: 0 bis 10^8 - 11 Seiten

-
- for i from 0 to 10^8 do: if isprime (i)and isprime (14!+i)and isprime (15!+i)and isprime (16!+i)and isprime (17!-i) (18!-i)and isprime (100!-i) then print ("i",i,14!+i,15!+i,16!+i,17!-i,18!-i,100!-i) end_if: end_for:

```
"i", 264871, 87178556071, 1307674632871, 20922790152871, 355687427831129,

   6402373705463129,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999999735129

"i", 321553, 87178612753, 1307674689553, 20922790209553, 355687427774447,

   6402373705406447,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999999678447

"i", 1415569, 87179706769, 1307675783569, 20922791303569, 355687426680431,

   6402373704312431,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999998584431

"i", 1644571, 87179935771, 1307676012571, 20922791532571, 355687426451429,

   6402373704083429,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999998355429

"i", 1824331, 87180115531, 1307676192331, 20922791712331, 355687426271669,

   6402373703903669,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999998175669

"i", 2632477, 87180923677, 1307677000477, 20922792520477, 355687425463523,

   6402373703095523,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999997367523

"i", 3370669, 87181661869, 1307677738669, 20922793258669, 355687424725331,

   6402373702357331,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999996629331

"i", 3989233, 87182280433, 1307678357233, 20922793877233, 355687424106767,

   6402373701738767,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999996010767

"i", 4946323, 87183237523, 1307679314323, 20922794834323, 355687423149677,

   6402373700781677,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999995053677

"i", 5747341, 87184038541, 1307680115341, 20922795635341, 355687422348659,

   6402373699980659,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999994252659

"i", 8148557, 87186439757, 1307682516557, 20922798036557, 355687419947443,
```

6402373697579443,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999999991851443

"i", 8565217, 87186856417, 1307682933217, 20922798453217, 355687419530783,

6402373697162783,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999999991434783

"i", 9343993, 87187635193, 1307683711993, 20922799231993, 355687418752007,

6402373696384007,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999999990656007

"i", 9402917, 87187694117, 1307683770917, 20922799290917, 355687418693083,

6402373696325083,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999999990597083

"i", 10133857, 87188425057, 1307684501857, 20922800021857, 355687417962143,

6402373695594143,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999999989866143

"i", 10635343, 87188926543, 1307685003343, 20922800523343, 355687417460657,

6402373695092657,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999999989364657

"i", 12309503, 87190600703, 1307686677503, 20922802197503, 355687415786497,

6402373693418497,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999999987690497

"i", 12639083, 87190930283, 1307687007083, 20922802527083, 355687415456917,

6402373693088917,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999999987360917

"i", 12680593, 87190971793, 1307687048593, 20922802568593, 355687415415407,

6402373693047407,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999999987319407

"i", 13282697, 87191573897, 1307687650697, 20922803170697, 355687414813303,

6402373692445303,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999999986717303

"i", 14585629, 87192876829, 1307688953629, 20922804473629, 355687413510371,

6402373691142371,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999999985414371

"i", 15010883, 87193302083, 1307689378883, 20922804898883, 355687413085117,

6402373690717117,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999999984989117

"i", 15816329, 87194107529, 1307690184329, 20922805704329, 355687412279671,

6402373689911671,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999984183671

"i", 15817831, 87194109031, 1307690185831, 20922805705831, 355687412278169,

6402373689910169,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999984182169

"i", 17163799, 87195454999, 1307691531799, 20922807051799, 355687410932201,

6402373688564201,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999982836201

"i", 17215687, 87195506887, 1307691583687, 20922807103687, 355687410880313,

6402373688512313,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999982784313

"i", 17313187, 87195604387, 1307691681187, 20922807201187, 355687410782813,

6402373688414813,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999982686813

"i", 17505743, 87195796943, 1307691873743, 20922807393743, 355687410590257,

6402373688222257,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999982494257

"i", 18445501, 87196736701, 1307692813501, 20922808333501, 355687409650499,

6402373687282499,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999981554499

"i", 18501359, 87196792559, 1307692869359, 20922808389359, 355687409594641,

6402373687226641,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999981498641

"i", 20746013, 87199037213, 1307695114013, 20922810634013, 355687407349987,

6402373684981987,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999979253987

"i", 20878537, 87199169737, 1307695246537, 20922810766537, 355687407217463,

6402373684849463,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999979121463

"i", 21024271, 87199315471, 1307695392271, 20922810912271, 355687407071729,

6402373684703729,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999978975729

"i", 22837747, 87201128947, 1307697205747, 20922812725747, 355687405258253,

6402373682890253,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999977162253

"i", 23524187, 87201815387, 1307697892187, 20922813412187, 355687404571813,

6402373682203813,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999976475813

"i", 25010599, 87203301799, 1307699378599, 20922814898599, 355687403085401,

6402373680717401,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999974989401

"i", 25300937, 87203592137, 1307699668937, 20922815188937, 355687402795063,

6402373680427063,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999974699063

"i", 25920151, 87204211351, 1307700288151, 20922815808151, 355687402175849,

6402373679807849,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999974079849

"i", 26206727, 87204497927, 1307700574727, 20922816094727, 355687401889273,

6402373679521273,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999973793273

"i", 27473857, 87205765057, 1307701841857, 20922817361857, 355687400622143,

6402373678254143,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999972526143

"i", 28473491, 87206764691, 1307702841491, 20922818361491, 355687399622509,

6402373677254509,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999971526509

"i", 29635943, 87207927143, 1307704003943, 20922819523943, 355687398460057,

6402373676092057,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999970364057

"i", 29707303, 87207998503, 1307704075303, 20922819595303, 355687398388697,

6402373676020697,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999970292697

"i", 29752969, 87208044169, 1307704120969, 20922819640969, 355687398343031,

6402373675975031,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999970247031

"i", 30533771, 87208824971, 1307704901771, 20922820421771, 355687397562229,

6402373675194229,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999969466229

"i", 31467847, 87209759047, 1307705835847, 20922821355847, 355687396628153,

6402373674260153,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\

5608941463976156518286253697920827222375825118521091686399999999999999999968532153

"i", 31740307, 87210031507, 1307706108307, 20922821628307, 355687396355693,

6402373673987693,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827222375825118521091686399999999999999999968259693

"i", 31796803, 87210088003, 1307706164803, 20922821684803, 355687396299197,

6402373673931197,

93326215443944152681699238856266700490715968264381621468592963895217599993229 91\
5608941463976156518286253697920827222375825118521091686399999999999999999968203197

"i", 34035817, 87212327017, 1307708403817, 20922823923817, 355687394060183,

6402373671692183,

93326215443944152681699238856266700490715968264381621468592963895217599993229 91\
5608941463976156518286253697920827222375825118521091686399999999999999999965964183

"i", 34217263, 87212508463, 1307708585263, 20922824105263, 355687393878737,

6402373671510737,

93326215443944152681699238856266700490715968264381621468592963895217599993229 91\
5608941463976156518286253697920827222375825118521091686399999999999999999965782737

"i", 34651637, 87212942837, 1307709019637, 20922824539637, 355687393444363,

6402373671076363,

93326215443944152681699238856266700490715968264381621468592963895217599993229 91\
5608941463976156518286253697920827222375825118521091686399999999999999999965348363

"i", 34828873, 87213120073, 1307709196873, 20922824716873, 355687393267127,

6402373670899127,

93326215443944152681699238856266700490715968264381621468592963895217599993229 91\
5608941463976156518286253697920827222375825118521091686399999999999999999965171127

"i", 35005843, 87213297043, 1307709373843, 20922824893843, 355687393090157,

6402373670722157,

93326215443944152681699238856266700490715968264381621468592963895217599993229 91\
5608941463976156518286253697920827222375825118521091686399999999999999999964994157

"i", 35321311, 87213612511, 1307709689311, 20922825209311, 355687392774689,

6402373670406689,

93326215443944152681699238856266700490715968264381621468592963895217599993229 91\
5608941463976156518286253697920827222375825118521091686399999999999999999964678689

"i", 35348843, 87213640043, 1307709716843, 20922825236843, 355687392747157,

6402373670379157,

93326215443944152681699238856266700490715968264381621468592963895217599993229 91\
5608941463976156518286253697920827222375825118521091686399999999999999999964651157

"i", 36924841, 87215216041, 1307711292841, 20922826812841, 355687391171159,

6402373668803159,

93326215443944152681699238856266700490715968264381621468592963895217599993229 91\
5608941463976156518286253697920827222375825118521091686399999999999999999963075159

"i", 37735669, 87216026869, 1307712103669, 20922827623669, 355687390360331,

6402373667992331,

93326215443944152681699238856266700490715968264381621468592963895217599993229 91\
5608941463976156518286253697920827222375825118521091686399999999999999999962264331

"i", 38681729, 87216972929, 1307713049729, 20922828569729, 355687389414271,

6402373667046271,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999961318271

"i", 38743619, 87217034819, 1307713111619, 20922828631619, 355687389352381,

6402373666984381,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999961256381

"i", 39639233, 87217930433, 1307714007233, 20922829527233, 355687388456767,

6402373666088767,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999960360767

"i", 40187977, 87218479177, 1307714555977, 20922830075977, 355687387908023,

6402373665540023,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999959812023

"i", 41459149, 87219750349, 1307715827149, 20922831347149, 355687386636851,

6402373664268851,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999958540851

"i", 41805769, 87220096969, 1307716173769, 20922831693769, 355687386290231,

6402373663922231,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999958194231

"i", 42812873, 87221104073, 1307717180873, 20922832700873, 355687385283127,

6402373662915127,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999957187127

"i", 43180337, 87221471537, 1307717548337, 20922833068337, 355687384915663,

6402373662547663,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999956819663

"i", 44445833, 87222737033, 1307718813833, 20922834333833, 355687383650167,

6402373661282167,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999955554167

"i", 44455157, 87222746357, 1307718823157, 20922834343157, 355687383640843,

6402373661272843,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999955544843

"i", 45043679, 87223334879, 1307719411679, 20922834931679, 355687383052321,

6402373660684321,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999954956321

"i", 46316351, 87224607551, 1307720684351, 20922836204351, 355687381779649,

6402373659411649,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999953683649

"i", 47007497, 87225298697, 1307721375497, 20922836895497, 355687381088503,

6402373658720503,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999952992503

"i", 48532067, 87226823267, 1307722900067, 20922838420067, 355687379563933,

6402373657195933,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999951467933

"i", 49186447, 87227477647, 1307723554447, 20922839074447, 355687378909553,

6402373656541553,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999950813553

"i", 49238993, 87227530193, 1307723606993, 20922839126993, 355687378857007,

6402373656489007,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999950761007

"i", 49969897, 87228261097, 1307724337897, 20922839857897, 355687378126103,

6402373655758103,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999950030103

"i", 50496191, 87228787391, 1307724864191, 20922840384191, 355687377599809,

6402373655231809,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999949503809

"i", 50642569, 87228933769, 1307725010569, 20922840530569, 355687377453431,

6402373655085431,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999949357431

"i", 51196991, 87229488191, 1307725564991, 20922841084991, 355687376899009,

6402373654531009,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999948803009

"i", 52911473, 87231202673, 1307727279473, 20922842799473, 355687375184527,

6402373652816527,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999947088527

"i", 53674787, 87231965987, 1307728042787, 20922843562787, 355687374421213,

6402373652053213,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999946325213

"i", 55872919, 87234164119, 1307730240919, 20922845760919, 355687372223081,

6402373649855081,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999944127081

"i", 56455307, 87234746507, 1307730823307, 20922846343307, 355687371640693,

6402373649272693,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999943544693

"i", 56690611, 87234981811, 1307731058611, 20922846578611, 355687371405389,

6402373649037389,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999943309389

"i", 57305459, 87235596659, 1307731673459, 20922847193459, 355687370790541,

6402373648422541,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999942694541

"i", 58882723, 87237173923, 1307733250723, 20922848770723, 355687369213277,

6402373646845277,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999941117277

"i", 59615359, 87237906559, 1307733983359, 20922849503359, 355687368480641,

6402373646112641,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999940384641

"i", 60074563, 87238365763, 1307734442563, 20922849962563, 355687368021437,

6402373645653437,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999939925437

"i", 61317631, 87239608831, 1307735685631, 20922851205631, 355687366778369,

6402373644410369,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999938682369

"i", 61572253, 87239863453, 1307735940253, 20922851460253, 355687366523747,

6402373644155747,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999938427747

"i", 61792957, 87240084157, 1307736160957, 20922851680957, 355687366303043,

6402373643935043,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999938207043

"i", 62978819, 87241270019, 1307737346819, 20922852866819, 355687365117181,

6402373642749181,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999937021181

"i", 64400551, 87242691751, 1307738768551, 20922854288551, 355687363695449,

6402373641327449,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999935599449

"i", 65189183, 87243480383, 1307739557183, 20922855077183, 355687362906817,

6402373640538817,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999934810817

"i", 68829323, 87247120523, 1307743197323, 20922858717323, 355687359266677,

6402373636898677,

93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916863999999999999999931170677

"i", 70932887, 87249224087, 1307745300887, 20922860820887, 355687357163113,

6402373634795113,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999929067113

"i", 71238701, 87249529901, 1307745606701, 20922861126701, 355687356857299,

6402373634489299,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999928761299

"i", 72880243, 87251171443, 1307747248243, 20922862768243, 355687355215757,

6402373632847757,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999927119757

"i", 73445573, 87251736773, 1307747813573, 20922863333573, 355687354650427,

6402373632282427,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999926554427

"i", 73838231, 87252129431, 1307748206231, 20922863726231, 355687354257769,

6402373631889769,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999926161769

"i", 73884929, 87252176129, 1307748252929, 20922863772929, 355687354211071,

6402373631843071,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999926115071

"i", 74031707, 87252322907, 1307748399707, 20922863919707, 355687354064293,

6402373631696293,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999925968293

"i", 75648359, 87253939559, 1307750016359, 20922865536359, 355687352447641,

6402373630079641,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999924351641

"i", 79269529, 87257560729, 1307753637529, 20922869157529, 355687348826471,

6402373626458471,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999920730471

"i", 79554721, 87257845921, 1307753922721, 20922869442721, 355687348541279,

6402373626173279,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999920445279

"i", 79863401, 87258154601, 1307754231401, 20922869751401, 355687348232599,

6402373625864599,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999920136599

"i", 81896987, 87260188187, 1307756264987, 20922871784987, 355687346199013,

6402373623831013,

9332621544394415268169923885626670049071596826438162146859296389521759999322991\
5608941463976156518286253697920827223758251185210916863999999999999999918103013

"i", 82179919, 87260471119, 1307756547919, 20922872067919, 355687345916081,

6402373623548081,

93326215443944152681699238856266700490715968264381621468592963895217599993229915\
608941463976156518286253697920827223758251185210916863999999999999999917820081

"i", 82425433, 87260716633, 1307756793433, 20922872313433, 355687345670567,

6402373623302567,

93326215443944152681699238856266700490715968264381621468592963895217599993229915\
608941463976156518286253697920827223758251185210916863999999999999999917574567

"i", 82562773, 87260853973, 1307756930773, 20922872450773, 355687345533227,

6402373623165227,

93326215443944152681699238856266700490715968264381621468592963895217599993229915\
608941463976156518286253697920827223758251185210916863999999999999999917437227

"i", 84444637, 87262735837, 1307758812637, 20922874332637, 355687343651363,

6402373621283363,

93326215443944152681699238856266700490715968264381621468592963895217599993229915\
608941463976156518286253697920827223758251185210916863999999999999999915555363

"i", 86620817, 87264912017, 1307760988817, 20922876508817, 355687341475183,

6402373619107183,

93326215443944152681699238856266700490715968264381621468592963895217599993229915\
608941463976156518286253697920827223758251185210916863999999999999999913379183

"i", 87412727, 87265703927, 1307761780727, 20922877300727, 355687340683273,

6402373618315273,

93326215443944152681699238856266700490715968264381621468592963895217599993229915\
608941463976156518286253697920827223758251185210916863999999999999999912587273

"i", 87445507, 87265736707, 1307761813507, 20922877333507, 355687340650493,

6402373618282493,

93326215443944152681699238856266700490715968264381621468592963895217599993229915\
608941463976156518286253697920827223758251185210916863999999999999999912554493

"i", 89803069, 87268094269, 1307764171069, 20922879691069, 355687338292931,

6402373615924931,

93326215443944152681699238856266700490715968264381621468592963895217599993229915\
608941463976156518286253697920827223758251185210916863999999999999999910196931

"i", 89927183, 87268218383, 1307764295183, 20922879815183, 355687338168817,

6402373615800817,

93326215443944152681699238856266700490715968264381621468592963895217599993229915\
608941463976156518286253697920827223758251185210916863999999999999999910072817

"i", 94469449, 87272760649, 1307768837449, 20922884357449, 355687333626551,

6402373611258551,

93326215443944152681699238856266700490715968264381621468592963895217599993229915\
608941463976156518286253697920827223758251185210916863999999999999999905530551

"i", 99258553, 87277549753, 1307773626553, 20922889146553, 355687328837447,

6402373606469447,

93326215443944152681699238856266700490715968264381621468592963895217599993229915\
608941463976156518286253697920827223758251185210916863999999999999999900741447

"i", 99272659, 87277563859, 1307773640659, 20922889160659, 355687328823341,

6402373606455341,

93326215443944152681699238856266700490715968264381621468592963895217599993229915\
608941463976156518286253697920827223758251185210916863999999999999999900727341

```
"i", 99698563, 87277989763, 1307774066563, 20922889586563, 355687328397437,

   6402373606029437,

93326215443944152681699238856266700490715968264381621468592963895217599993229915\
56089414639761565182862536979208272237582511852109168639999999999999999900301437
```

•

Anmerkung:

Diese kombinierten Anti - Goldbach - und Goldbach - Zerlegungen bilden "Primzahl - Wellen - Tsunamis", die sich in wachsenden Zahlen-Intervallen fortpflanzen, wobei die Abstände zwischen den Primzahlen: p_1 bis p_3 und p_4 bis p_6 in allen 107 Zeilen konstant bleiben. Die Abstände zwischen den Primzahlen: p_3 und p_4 wachsen hingegen zunehmend in definierten Stufen (mit dem Anstieg der primen Zahl:= i)

Patentansprüche

1. Alle Ansprüche, die aus der additiven Zerlegung von Primzahlen herrühren und implementierbarer Bestandteil statischer, dynamischer Tabelleninhalte zur Verschlüsselung nach Goldbach'schen und Anti-Goldbach'schen Gesetzmäßigkeiten führen.

2. Schutzbedürftig ist das Prinzip der synchron rollierenden Authentisierung, welches auf der Grundlage des Austausches von verschlüsselten Kernzahlen eines dedizierten Algorithmus bestehen und auf der Grundlage additiver Primzahlzerlegungen beruhen.

3. Alle Ansprüche, die sich aus den Bildungsgesetzen der beigefügten Monographien des Dr. Cwienk ergeben.

4. Die Art und Weise der Verfahrens- und Implementierungsformen des prototypisch beschriebenen Austausches von Informationen, die von einer oder mehreren Quellen zu einer oder mehreren Senken dediziert, gesichert über die additiven Zerlegungsformen der Primzahlen oder nicht primen Zahlen erfolgt.

5. Besonders schützenswürdig ist die Möglichkeit der hardwarenahen Implementierungsformen der beschriebenen Verfahrensformen von additiven Primzahlzerlegungen zur Bildung von Schlüsseln beim Datenaustausch zwischen beliebigen authorisierten und sich gegenseitig authentifizierenden Quellen und Senken.

6. Schützenswürdig ist jedwede Möglichkeit eines Schlüsselverfahrensaufbaus, der Datenspeicher bezogenen Implementierung von additiv errechneten Algorithmen zur Authentisierung, Authenifizierung und Datenstromchiffrierung führen kann und auf den erwähnten additiven Zerlegungen von Primzahlen und nicht primen Zahlen beruht.

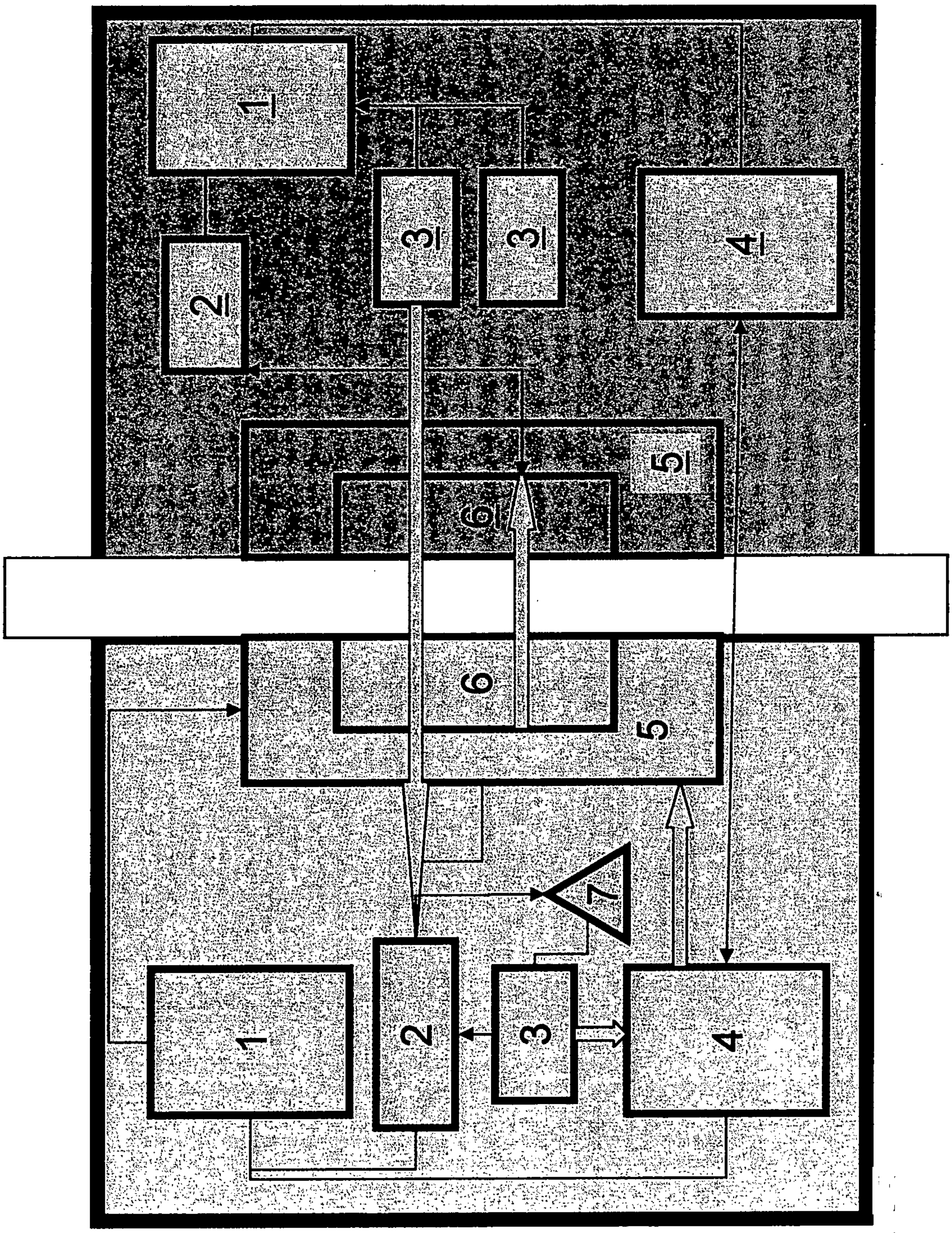
7. Als schutzwürdig wird die Nutzung des ICMP-Protokolls gesehen, um gesichert Kryptomittel in verschiedenen Authentisierungsformen auszutauschen.

8. Schutzwürdig ist die systemimmanente Blocksatzchiffrierung und die Zuordnung zu unterschiedlich gebildeten, sich gegenseitig ergänzenden Zerlegungsformen der additiven Bildungsgesetzte wie in den Monographien des Dr. Cwienk beschrieben.

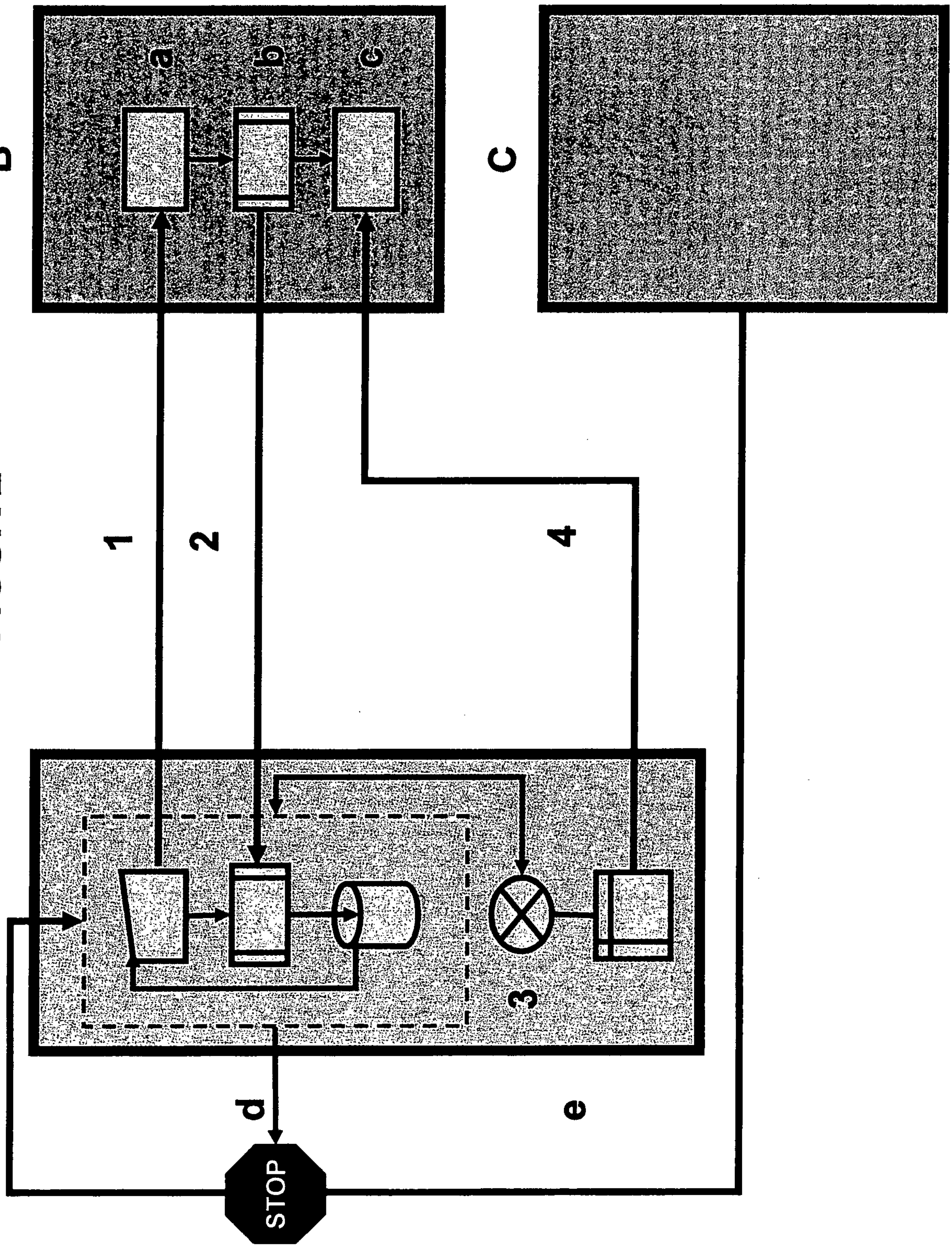
Es folgen 4 Blatt Zeichnungen

Anhängende Zeichnungen

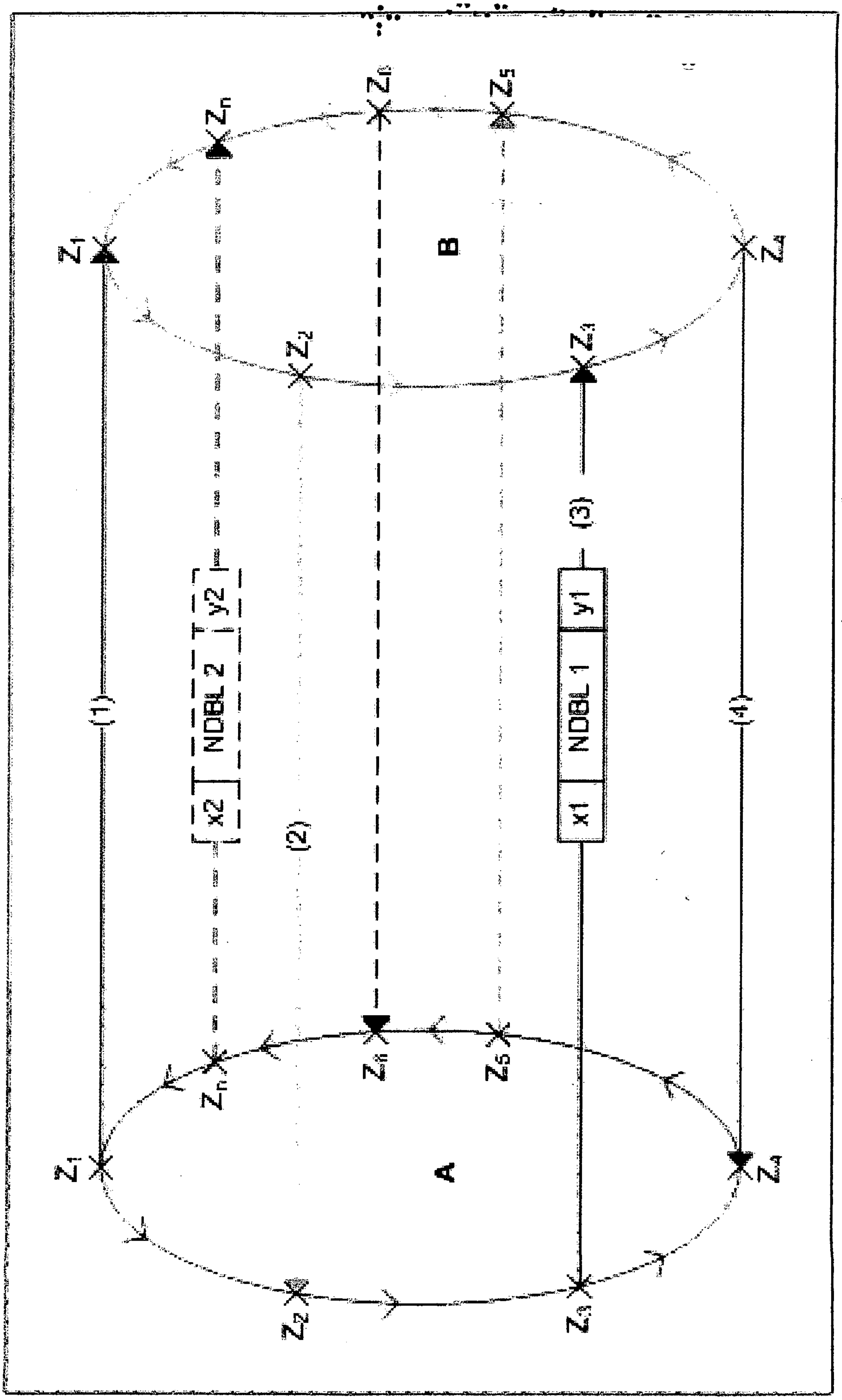
FIGUR 1



FIGUR 2



FIGUR 3



Figur 4
Prinzip Ablauf Content-Krypto
SynD