

# kAri OJ（北邮 OJ）常见问题

## 1. 问：我的程序是如何输入又是怎样输出的？

答：你的程序应该用 `stdio` 输入(标准的输入流)，输出应该用 `stdout` 输出(标准的输出流)。比如说：你可以使用 C 里面的 `scanf` 和 C++ 里面的 `cin` 来读入，而使用 C 里面 `printf` 或 C++ 里面的 `cout` 来写出。

你提交的程序将不允许进行任何文件的操作。你有可能为此而得到[运行时错误\(Runtime Error\)](#)或者[答案错误\(Wrong Answer\)](#)。

G++ 中的 I/O 操作，是基于比较复杂的实现模式，相对于 `scanf` 和 `printf` 而言，`cin` 和 `cout` 要慢很多。经验表明他们之间的差异在当你使用 G++ 作为编译器的时候显得尤为突出。所以当一个问题要求有比较大量的输入和输出时，`cin` 和 `cout` 将会导致[超出时间限制\(Time Limit Exceed\)!](#)

## 2. 问：这个评判系统使用的是什么编译器？

答：当前的编译器支持三种语言环境。有 GNU C 4.6.3(GCC) 和 GNU C++ 4.6.3(G++) 以及 Java (OpenJDK 1.6.0\_27)。

各语言的编译命令如下：

1. gcc: `gcc -o dest source_code.c -DONLINE-JUDGE -lm -std=c99`
2. g++: `g++ -o dest source_code.c -DONLINE-JUDGE`
3. Java: `javac Main.java -d .`

对于经典的 [A+B Problem](#)，我们可以用以下代码通过题目。

C:

```
1. #include <stdio.h>
2.
3. int main() {
4.     int a, b;
5.     scanf("%d%d", &a, &b);
6.     printf("%d\n", a + b);
7.     return 0;
```

```
8. }
```

C++:

```
1. #include <iostream>
2. using namespace std;
3.
4. int main() {
5.     int a, b;
6.     cin >> a >> b;
7.     cout << a + b << endl;
8.     return 0;
9. }
```

对于 Java，你需要实现一个名为 Main 的类，其中包含一个静态方法 main，如下所示：

```
1. import java.util.*;
2.
3. public class Main {
4.     public static void main(String args[]) {
5.         Scanner in = new Scanner(System.in);
6.         int a = in.nextInt();
7.         int b = in.nextInt();
8.         System.out.println(a + b);
9.     }
10. }
```

值得注意的是：对 64 位整数，请使用 `long long` 进行申明，GNU 不支持 `__int64`，输入输出使用 `lld`。例如：

```
1. long long a, b;
2. scanf("%lld%lld",&a, &b);
3. printf("%lld\n", a + b);
```

### 3. 问：那我的程序是怎样被判的呢？

答：评判系统首先是将你提交的代码保存起来，然后按照你选择的语言进行编译。如果编译失败，那么你将得到[编译错误](#)作为评判结果。

如果通过了编译这一关，你的代码编译生成一个可执行文件，接下来 judge 会将标准的输入数据作为输入，输入到这个产生的文件中然后将输出放到一个输出文件中。在你的程序执行标准的输入文件的过程中，如果发生**运行时错误**，**超过时间限制**，**超过输出限制**，你将会得到相应的评判结果。一旦出现了以上的问题，你的程序将不会被继续执行。

之后，评判系统就将这个输出文件和正确的输出文件进行严格的比较。如果与输出文件不一致，可能是答案错误，你将得到相应的评判结果。

如果你的程序没有符合上面任何一种情况，那么祝贺你，你将获得**通过**作为最终评判结果。

需要注意的是，OJ 判题程序仅仅允许提交的代码进行"正当的"操作，这意味着你的程序只能从标准输入读入数据，用某些算法处理完毕之后把结果输出到屏幕。**任何类似运行其他命令(例如 `system("pause");`)、创建子进程、创建 socket 和读取硬盘文件的操作都是不允许的**。这样的操作被判题程序检测到后会立即终止提交代码的运行，并返回运行时错误。

#### 4. 问：Judge 所返回的回复是什么意思？

答：以下是所有的回复所对应的含义：

- **等待评测(Pending)** 表示你的代码正在等待评判系统的评判。
- **通过(Accepted)** 祝贺你！你的程序是正确的！
- **格式错误(Presentation Error)** 虽然你程序的结果是正确的，但是输出结果的格式不对，请你再仔细检查，是不是你在某些位置多输出了空格或者空行。
- **编译错误(Compile Error)** 你程序有语法错误，没有通过编译，请仔细检查。
- **答案错误(Wrong Answer)** 你的程序并没有得到正确的输出结果。
- **运行时错误(Runtime Error)** 在你的程序的执行过程中，可能出现了以下情况：非法的文件访问，堆栈溢出，或者是非法访问了内存单元，数组越界，计算中 0 作为除数等导致异常结束的情况。
- **超过时间限制(Time Limit Exceed)** 你的程序没有在限定时间内执行完。这里有两种限定时间--总限定时间(total time limit)和单组限定时间(case time limit)，前者是对你的程序执行完全部输入文件的总时间的限定。后者是对程序执行如数文件中的每一组输入的时间的限定。当你的程序超过了两者中任意一个的限定那么将导致该错误。如果你的程序的耗时少于题目给的限定，那么你的程序一定是超过了单组限定时间。一般情况下题目没有特殊的要求的话，题目的单组限定时间就是题目的所给的时限。

- **超出内存限制(Memory Limit Exceed)** 你使用了过大的内存，请减小数组或者其它消耗内存的东西。
- **超过输出限制(Output Limit Exceed)** 你的程序产生了过大的输出文件。如果你产生的输出文件达到了标准输出文件的三倍大小那么会有这样的结果，一般是由于死循环导致的，注意：死循环也有可能被判定为 OLE。

## 5. 问：在有些题目名称旁边的那个红色的 **Special Judge** 表示什么呢？

答：当一个题目有多个可以正确的输出结果时，一个 **Special Judge** 程序将被使用。它将用来检查你的输出结果是否符合题目的要求。

## 6. 问：我怎样才能判断一个输入的开始？

答：一般题目会给你会有一些特别的信息作为输入及结束，比如输入 0 作为结束。但是很多情况都是用 EOF 来结束输入，你可以通过测试 `scanf` 和 `cin` 接受 EOF 时的返回值来判断结束。

例如，对于 C/C++ 可以使用如下代码：

```
1. while(scanf("%d", &n) != EOF) {  
2.     ...;  
3. }
```

对于 C++，也可以这么写：

```
1. while(cin >> n) {  
2.     ...;  
3. }
```

对于 Java：

```
1. Scanner in = new Scanner(System.in);  
2. while (in.hasNextInt()) {  
3.     ...;  
4. }
```

当然，要完成相同的操作还可以有更多的方法，这里就不再赘述了。

**7. 问：我明明已经通过了题目中给的样例，但为什么我不管怎么交都是答案错误？**

答：样例仅仅是用于帮助理解题意，事实上，样例可能只是测试数据中的冰山一角。因此请再仔细检查代码中的细节，查看代码中存在的问题以应付样例之外的测试点。

**8. 问：我的超过时间限制的代码只是超过时限 10ms。我怎样才能提高那 10ms？**

答：评判系统会在你程序运行时间超过时限后就马上停止你程序的运行，一般是在超过时限 10ms 后停止执行，你程序的实际运行时间可能远远超过这个数。唯一的解决办法是降低你的程序的时间复杂度，使你的程序更快更有效。

**9. 问：考试/比赛时是按什么规则排名？**

首先按照过题数从高到低。如果过题数相同，则按罚时总和从低到高排序。罚时计算规则为：单题罚时 = (通过每题时间+错误提交的次数×20) 分钟

通过每题后，排行榜上对应题目显示的数字即为该题的罚时

**10. 问：我还有很多疑问，请问在哪里可以得到很好的回复呢？**

答：请到我们的 [BBS\(北邮人论坛 ACM/ICPC 版\)](#)上发帖或者通过 [mail](#) 和我们联系，那里的管理员和用户将会很乐意解决你的问题！^\_^