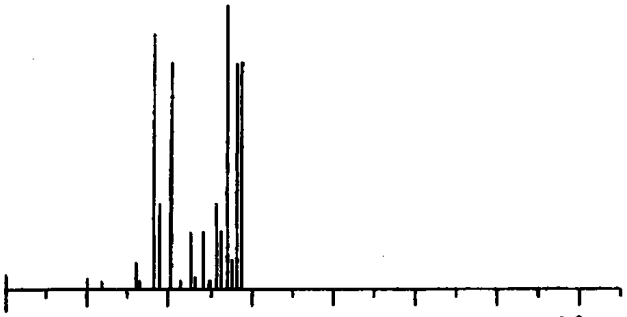
Ref.
1. Williams, S. A. and Anthony, J. W. (1970), Hemihedrite, A New Mineral From Arizona, *Am. Min.* 55, 1088–1102.
2. McLean, W. J. and Anthony, J. W. (1970), The Crystal Structure of Hemihedrite, *Am. Min.* 55, 1103–1114.



American Mineralogist 55, 1100, (1970) $2\theta \rightarrow$

Powd. Pat. Debye–Scherrer (114.6 mm; CuK_α; Visual *I*).

STRONGEST LINES		LARGEST *d* SPACINGS					
3.301	100	8.765	4	4.364	80	3.478	30
4.872	90	7.481	3	4.136	3	3.399	20
4.364	80	5.512	9	3.909	20	3.301	100
3.164	80	5.384	3	3.820	4	3.234	10
3.102	80	4.872	90	3.676	20	3.164	80
2.924	55	4.675	30	3.584	3	3.102	80

Struct. Cell. $P1 - C_1^1$; $Z = 1$;

$a = 9.497(1)$, $b = 11.443(2)$, $c = 10.841(2)$
$\alpha = 120°30(1)'$, $\beta = 92°06(1)'$, $\gamma = 55°50(1)'$
$a : b : c = 0.830 : 1 : 0.947$

Chem. Substitution of Zn for Pb noted in some samples.

	1	2
ZnO	2.7	3.93
PbO	73.0	70.5
Cr_2O_3	19.7	19.5
SiO_2	3.9	3.2
F	1.2	5.1
$-O \equiv F$	−0.5	−2.1
Total	100.0	100.0

1. $ZnF_2[Pb_5(CrO_4)_3SiO_4]_2$.
2. Average of several partial analyses.

Opt. Thin section shows feeble pleochroism with $Z > Y > X$. Relief extreme; dispersion resembles horizontal dispersion.

$\alpha = 2.105(5)$ yellow
$\beta = 2.32$ (2) yellow $\quad (2V_z)_{meas} = 92°(-)$
$\gamma = 2.65$ (2) orange $\quad (2V_z)_{calc} = 88°(+)$

* * * * * * * * * * *

Problems

* * * * * * * * * * *

Send Submissions to:
Lynne A. Price
TUG Macro Coordinator
Calma R&D
212 Gibraltar Dr.
Sunnyvale, CA 94086

Balancing Columns of Text and Translation

In the last issue, Johnny Stovall asked about a macro that could adjust the width of each column of two-column output so that the lengths of the two columns will be equal. His application involves typesetting original texts in parallel with translations. As long as reliable estimates of the relative length of the two segments are available, a simple technique can be used. `\varunit` can be set to the width available for text in both columns (excluding margins), and the actual width of each column can then be set in terms of a percentage of vu.

The following macro illustrates this approach using `\hbox par`:

```
\input basic
\varunit 6in % Space available for both
             % columns, exclusive of margins
\def\intercolumnspacing{\hskip .5in}
% Arguments to \trans are percentages
% of total width for first column,
% contents of first column, percentage
% of total width for second column, and
% contents of second column, respectively.
% The two numbers should sum to 100.
\def\trans#1#2#3#4{
  \hbox{\hbox par 0.#1vu
  {#2}\intercolumnspacing
  \hbox par 0.#3vu{#4}}
}
```

Here is a simple example:

Now is the time for all good men to come to the aid of their party. Now is the time for all good men to come to the aid of their party. Now is the time for all good men to come to the aid of their party. Now is the time for all good men to come to the aid of their party. Now is the time for all good men to come to the aid of their party. Now is the time for all good men to come to the aid of their party.

Now is the time for all good men to come to the aid of their party. Now is the time for all good men to come to the aid of their party. Now is the time for all good men to come to the aid of their party. Now is the time for all good men to come to the aid of their party. Now is the time for all good men to come to the aid of their party. Now is the time for all good men to come to the aid of their party.

And continue ...

Here, it is assumed that there is text extending across the full page width above or below the translations. This macro must be modified if the text segments include multiple paragraphs. A similar macro

could be used to redefine \hsize if the entire text has this form.

Input-Dependent Macro Redefinition

Also in the last issue, Mike Spivak asked about a pair of macros \data and \list. Successive calls to \data would save text that could later be retrieved by calling \list, so that \list{1} would produce the parameter passed on the first call to \data, \list{2} would produce the parameter passed on the second call to \data, and so on.

Macros to solve this problem are fairly straightforward—\data can concatenate its successive arguments on a single string, separating them with some delimiter, and then \list can pick off the necessary text string. The concatenation of the passed strings, however, is created with \xdef and a problem arises when some of the arguments themselves contain \xdefs that the user does not want evaluated until \list is invoked.

A solution is possible using the technique illustrated by Patrick Milligan's \DefineFont macro published in TUGboat, Vol. 2, No. 2. The macro \macrolist is defined to contain a sequence of otherwise unused macro names (e.g., \aa, \bb, \cc ...). Each time \data is called, it removes the first item from \macrolist, saves it on a list \listitems, and then defines the specified macro to invoke the argument text. Thus, if \macrolist starts \aa\bb\cc..., the first call to \data sets \listitems to \aa, sets \macrolist to \bb\cc\dd..., and defines \aa to be whatever the argument to \data was. The \xdef to modify \listitems does not cause \aa to be expanded, because \listitems is redefined before \aa is defined. To retrieve the stored macros, \list must pick off the appropriate token from \listitems and execute it.

The algorithm just sketched is the heart of the macros shown here. The situation is somewhat more complicated, however. \list works by using \xdefs applied to \listitems. To prevent \list from expanding any \xdefs that might be contained in the argument text \data has so far received, \listitems is constructed with the undefined macros \a, \b, \c, etc. After \list has selected the appropriate item, \a is locally defined (i.e., defined within a group) to invoke \aa, \b to invoke \bb, and so on through the ugly macro \equivalences. The selected macro is then invoked within the group that defines these equivalences. To allow all this, \macrolist is really initialized to \aa\a\bb\b\cc\c\dd\d....

These macros also use the \Apply, \First, and \Rest macros used by Patrick in \DefineFont.

\list uses the recursion macros Brendan M described in the same issue of TUGboat. Th vious restriction that these macros are limit 26 calls to \data can, of course, be extend reinitializing \macrolist. A less obvious restr comes from the local equivalence of \a and \a A macro defined with \def within an argumen is usable only from within the same argument, the argument is invoked within a group. \gde \xdef, of course, have more permanent effects. restriction can be surmounted by redefining to explicitly test its argument and to call \aa argument is 1, \bb if it is 2, etc. Unaesthetic as an approach may be, it is not much worse tha current definition of \equivalences. Note t "groupless if" such as McKay and Spivak als cuss in the last issue must be used to avoid the problem.

```
\xdef\listitems{}

\def\data#1{
  \if !\macrolist!{
    \send9{Error: Maximum number of
                  data items exceeded}}
  \else{
    \Apply {\First} to {\macrolist!} ->
          {\macro} % Get macro name
    \Apply {\Rest} to {\macrolist!} ->
          {\macrolist} % Remove code from
    \Apply {\First} to {\macrolist!} ->
          {\macroprime} % Get macro name
    \Apply {\Rest} to {\macrolist!} ->
          {\macrolist} % Remove code from
    \xdef\listitems{\listitems\macroprime}
    \define\macro{#1}
  }
}

\def\define#1#2{
  \let \Gdef=\let
  \xdef\Define{\Gdef #1}
  \let \Gdef=\gdef
  \Define{#2}
  }

\def\list#1{{
  \xdef\temp{\listitems}
  \repeat #1\times
    \if !\temp! {\setcount9 0\gdef\macro{}}
    \else{\Apply {\First} to {\temp!} ->
                 {\macro}
          \Apply {\Rest} to {\temp!} ->
                 {\temp}}\endrepeat
  \equivalences
  \macro}}

% The macro \macrolist describes the set of
% macro names available to \list and \data.
% These macros should never be explicitly
% invoked by the user.
```

```
\def\macrolist{\aa\a\bb\b\cc\c\dd\d\ee\e
  \ff\f\gg\g\hh\h\ii\i\jj\j\kk\k\ll\l
  \mm\m\nn\n\oo\o\pp\p\qq\q\rr\r\ss\s
  \tt\t\uu\u\vv\v\ww\w\xx\x\yy\y\zz\z}

% \equivalences equates pairs of consecutive
% macros declared in \macrolist

\def\equivalences{\def\a{\aa}\def\b{\bb}\def
  \c{\cc}\def\d{\dd}\def\e{\ee}\def\f{\ff}\def
  \g{\gg}\def\h{\hh}\def\i{\ii}\def\j{\jj}\def
  \k{\kk}\def\l{\ll}\def\m{\mm}\def\n{\nn}\def
  \o{\oo}\def\p{\pp}\def\q{\qq}\def\r{\rr}\def
  \s{\ss}\def\t{\tt}\def\u{\uu}\def\v{\vv}\def
  \w{\ww}\def\x{\xx}\def\y{\yy}\def\z{\zz}}
```

A Macro That Prints Its Name

Anthony Siegman asks whether it is possible to devise a TEX macro that will process a one-word argument as both a word of text and a control sequence.

Suppose the argument is entered as {name}, or alternatively as {\name} (one or the other, not both). He wants a macro that will:

(a) Accomplish the result \xdef\name{\count 4}

(b) Put both the numerical value of \name and the word "name" into the output text.

(c) Send both the numerical value of \name and the word "name" to an external file.

Mike Spivak has written the following sequence

```
\def\findname{\chcode'134 12 \findit}
\def\findit #1{\finditt#1\endd}
\def\finditt #1#2\endd{\gdef
        \thename{#2}\chcode'134=0 }
```

which allows the input

```
\findname{\foo}
```

to define \thename to be "foo". However, despite appearances, \findname has no parameters; {\foo} is the text following \findname. This technique illustrates that TEX can be made to treat what appears to be a control sequence as two tokens: the text character \ followed by a word of text. It can not be extended to solve the complete problem presented by Siegman, since once TEX has determined the character class of a symbol, the character cannot be reinterpreted. Thus, any one occurrence of the four characters "\foo" can be interpreted either as four text characters or as one control sequence, but can not sequentially be interpreted both ways.

Nevertheless, macros approximating what has been asked for can be written. Instead of defining a macro \foo, the following \setname macro, given the argument "foo", allows another macro to return the value of \count4 at the time \setname{foo} was called. This new macro is invoked by a single character, @, so, in effect, the control sequence defined by \setname is @foo rather than \foo. Of course, "@foo" is not really a control sequence, but it may be used as such. Actually, "foo" is an argument to the macro @. @ expects its parameters to be terminated by a space, the user must remember to include such a space even in situations (e.g., before punctuation) where it would not be required for an actual control sequence.

The \setname macro defined below defines a list of its arguments on successive calls, separated by semicolons and a similar list of the values of \count4 each time it was called. The macro @ then looks through the list of names until it finds the one matching its argument and returns the corresponding \count4 value.

```
% Used to put a space between the number
% and name output by \send
\def\space{ }

% Initialize lists
\def\namelist{}
\def\numberlist{}

% Define a new "name"
\def\setname#1{
     \xdef\namelist{\namelist #1;}
     \xdef\numberlist{\numberlist\count4;}
     \xdef\number  % Number followed by
          {\count4 } % space #1; Put numerical
     \number\ #1  % value & word into text
     \send9{\number\space #1}% Send number
     }              % & name to external file

% Retrieve a previously saved value
\def\name#1 {\xdef\nametemp{\namelist}
             \def\numbertemp{\numberlist}
             \xdef\lookfor{#1}
             \gdef\action{\next}
             \next
            }

% Test if next name on temporary list is one
% needed. Note, if the name is not on the list,
% an error message will be generated since
% \nosuchname has not been defined
\def\next{\if !\nametemp!{\gdef\action
                    {}\nosuchname}\else{}
       \Apply {\first} to {\nametemp!} ->
                    {\nextname}
       \Apply {\first} to {\numbertemp!} ->
                    {\nextnumber}
       \Apply {\rest} to {\nametemp!} ->
                    {\nametemp}
       \Apply {\rest} to {\numbertemp!} ->
                    {\numbertemp}
       \stringeq{\nextname}{\lookfor}
       \if T\stringeqv {\nextnumber
                    \gdef\action{}}\else{}
       \action
     }

\def\first #1;#2!{#1}
```

```
%       Returns first token from list
\def\rest #1;#2!{#2}
%       Returns list with first token removed

% Allow \name to be invoked by the
% single character @
\def\ {\name}
\chcode`100=13
```

Testing the Equivalence of Strings

To test the equivalence of strings, the macro \stringeq is called. Shown below, this macro uses the "groupless if" techiques described independently by McKay and Spivak in Vol. 2, No. 2.

```
% String equivalence--evaluates to T
% if first argument is same string
% as second, F otherwise

% \stringeq returns its answer by
% setting \stringeqv
\def\true{\gdef\stringeqv{T}}
\def\false{\gdef\stringeqv{F}}

% Main "routine"  Save arguments in \stringa
% and \stringb and start testing
\def\stringeq#1#2{\xdef\stringa{#1}
  \xdef\stringb{#2}
  \gdef\endap{\enda}
  \gdef\endbp{\endb}
  \gdef\compcharp{\compchar}
  \enda
  }

% Test if at end of first string
\def\enda{\if !\stringa!{\gdef\endbp
                {\nulltest}}\else{}\endbp}

% Test if at end of second string
\def\endb{\if !\stringb!{\false
        \gdef\compcharp{}}\else{}\compcharp}

% Compare next characters
\def\compchar{\Apply {\First} to {\stringa!} ->
 {\firsta}\Apply {\First} to {\stringb!} ->
 {\firstb}\if \firsta\firstb
{\Apply {\Rest} to {\stringa!} -> {\stringa
}\Apply {\Rest} to {\stringb!} -> {\stringb}}
\else{\false\gdef\endap{}}
\endap}

% When at end of first string,
% test if also at end of second
\def\nulltest{\if !\stringb!{\true}\else
                          {\false}}
```

* * * * * * * * * * *

PROBLEMS FROM THE TEXARCANA CL ANSWERS, AND ANOTHER PROBLEM

These problems are from the video TEXarcana Class taught by Don Knuth last M The solutions given here are based on the one tributed in class. Some parts of the solutions depend on features which, although installed i current version of TEX at Stanford, may not yet been implemented elsewhere; an attempt has made to note such features.

Problem no. 1:

Type:

```
\vskip 12pt
\noindent\hide{--}Allan Temko

\vskip 2pt
\noindent Architecture Critic
```

To get:

–Allan Temko
Architecture Critic

```
\def \hide#1{\hbox to 0pt{\hss #1}}
```

Problem no: 2:

Type:

```
\fancy Senator and Mrs.\Stanford had reserved to themselves control of the
University's affairs during their lifetimes, including the parceling
out of ``all the money that could be wisely used.'' Mrs.\Stanford had remained in
her husband's shadow---on opening day she could not bring herself to deliver
the short speech she had written out. But following the death of the Senator
she, at age 65, took on full responsibility for the University with
unsuspected strength.
```

To get:

Senator and Mrs. Stanford had reserved to themselves control of the University's affairs during their lifetimes, including the parceling out of "all the money that could be wisely used." Mrs. Stanford had remained in her husband's shadow—on opening day she could not bring herself to deliver the short speech she had written out. But following the death of the Senator she, at age 65, took on full responsibility for the University with unsuspected strength.

```
\font H=cmr10 at 30pt
\def \fancy#1{{\:H\save0\hbox{\chop to 0pt{\hbox{\lower 12pt
     \hbox{#1}\hskip .1em}}}\hangindent 1wd0 for 2\noindent
     \hide\box0\hskip -.1em}}
```

The font specified here requires that magnification be implemented; it is also possible to use cmr30, if that is available. (At the AMS, neither is available for the Alphatype, on which this page was prepared, and the large "S" has been obtained from another source and pasted in.)

Note the use of \hide in the last line of this solution; alternatively, this line might begin "\hskip -1wd0 ...".

Problem no. 3:

Type:

```
\hsize 25em
\noindent This is a case where the name and address fit in nicely
with the review.\signed{A. Reviewer}{Ann Arbor, Mich.}

\vskip 8pt
\noindent But sometimes an extra line must be added.\signed{N. Bourbaki}{Paris}
```

To get:

This is a case where the name and address fit in nicely with the review. *A. Reviewer* (Ann Arbor, Mich.)

But sometimes an extra line must be added.
N. Bourbaki (Paris)

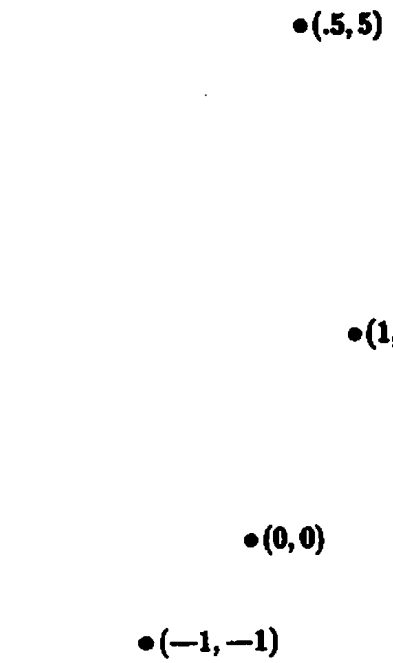
```
\def \signed#1#2{\parfillskip 0pt{\unskip\penalty 1000\hfil\penalty 200
\hskip 2em\hbox{}\penalty 1000\hfil{\sl#1\/} (#2)\par}}
```

Problem no. 4:

Type:

```
\point 0 0
\point 1 2
\point 2 1
\point .5 5
\point -1 -1
```

To get:




```
\lineskip 0pt
\baselineskip 0pt
\varunit .5in
\topspace 6vu \def \point#1#2 {\vbox to 0pt
    {\vss\vbox to #2vu{
     \hbox{\hskip 2vu\hskip #1vu
      $\bullet\,(#1,#2)$}\vss}}}
```

Problem no. 5:

Type:

```
\hsize 20em
End of a paragraph.\par
\rightjustifythefollowing
This is the first line
{\it This is the second line.}
{\sl The third.}
{\bf The last.}
\endrightjustify
Beginning of another paragraph.
```

To get:

End of a paragraph.

This is the fir
This is the secon
The
Th

Beginning of another paragraph.

```
\def \rightjustifythefollowing{\par
   \chcode`15=12\penalty1000
   \vskip-12pt\let\rjn=\rj\rj}
\chcode`15=12
\def \rj#1
{\rjustline{#1}\rjn}\chcode`15=5 %
\def \endrightjustify{\gdef\rjn{\endrj}}
\def \endrj{\chcode`15=5\penalty1000\vskip-1
```

Problem no. 6:

Type:

```
How do you do this?
$$\lineskip 2pt
\baselineskip 1.3ex
\vcenter{\halign{\hfil#\hfil\cr
\linedown{Look at this {strange} pile.}}}\qquad
\vcenter{\halign{\hfil#\cr
\lineup{And at this {stranger} one.}}}$$
```

To get:

How do you do this?

Look at this strange pile. And at this stranger one.

```
\chcode`44=12
\def\linedown#1{\gdef\ans{}\Linedown#1$}
\def\Linedown#1{\if $#1{\gdef\next{\ans}}
    \else{\xdef\ans{\ans#1\cr}\gdef\next{\Linedown}}
    \next}
\def\lineup#1{\gdef\ans{}\Lineup#1$}
\def\Lineup#1{\if $#1{\gdef\next{\ans}}
    \else{\xdef\ans{#1\cr\ans}\gdef\next{\Lineup}}
    \next}
\chcode`44=3
```

How do you do it faster?

```
$$\vcenter{\halign{\hfil#\hfil\cr
\linedn{Look at this {strange}pile.}}}$$

\def\linedn#1{\gdef\ans{}\Linedn#1$}
\def\Linedn#1{\if $#1{\gdef\next{\ans}}
    \else{\gdef\nans{\#1\cr}\gdef\next{\ans\Lndn}}
    \next}
\def\Lndn#1{\if $#1{\gdef\next{\nans}}
    \else{\gdef\ans{\#1\cr}\gdef\next{\nans\Linedn}}
    \next}
```

On the next page appears the "Challenge problem" presented to the TEXArcana attendees. The solution will appear in the next issue.

First page of output:

•If I have all the eloquence of men or of angels, but speak without love, I am 1
simply a gong booming or a cymbal clashing. •If I have the gift of prophecy, 2
understanding all the mysteries there are, and knowing everything, and if I have
faith in all its fulness, to move mountains, but without love, then I am nothing at
all. •If I give away all that I possess, piece by piece, and if I even let them take 3
my body to burn it, but am without love, it will do me no good whatever.

•Love is always patient and kind; it is never jealous; love is never boastful or 4
conceited; •it is never rude or selfish; it does not take offence, and is not resentful. 5
•Love takes no pleasure in other people's sins but delights in the truth; •it is always 6
ready to excuse, to trust, to hope, and to endure whatever comes. 7

•Love does not come to an end. But if there are gifts of prophecy, the time 8
will come when they must fail; or the gift of languages, it will not continue for

Second page of output:

ever; and knowledge—for this, too, the time will come when it must fail. •For our 9
knowledge is imperfect and our prophesying is imperfect; •but once perfection 10
comes, all imperfect things will disappear. •When I was a child, I used to talk 11
like a child, and think like a child, and argue like a child, but now I am a man,
all childish ways are put behind me. •Now we are seeing a dim reflection in a 12
mirror; but then we shall be seeing face to face. The knowledge that I have now is
imperfect; but then I shall know as fully as I am known.

•In short, there are three things that last: faith, hope and love; and the greatest 13
of these is love.

You should use this text file →

```
3 Mar 1981    21:06        LOVE.TEX[ 1,DEK]          PAGE
@ If I have all the eloquence of men or of
     angels, but speak without love, I am simply a gong boomin
     or a cymbal clashing.
@ If I have the gift of prophecy, understanding all the myst
     there are, and knowing everything, and if I have faith in
     fulness, to move mountains, but without love,
          nothing at all.
@ For our [illegible]hat I possess, piece by piece, and if I
@ but once perfection burn it, but am without love, it
@ When I was a child,
     child, and argue like a
          ways are put behind me.
@ Now we are seeing a dim reflection jealous;
     be seeing face to face. The knowledge
     imperfect; but then I shall know as fully

@ In short, there are three things that last: faith
     love; and the greatest of these is love.

\flushpage
```

The driver file says:

```
\input basic
\def\pagesize{1.8 in}
<secret code to do
this formatting>

\setcount1 1 % starting verse number
\input love [1,dek]
\end
```