

`\advance\pageno by 1`

The backslash serves as an `{\bf escape character}`, i.e., as a message to `\TeX\` that what immediately follows the backslash is to be taken as a command (rather than as text to be typeset).

In particular, you get all the symbols and signs `{\it not}` on your keyboard with the aid of such `{\bf command}`s. See Sam Bent's `\TeX\ Reference Card` in the Appendix for a complete, ordered listing.

`\heading{odds and ends}`

`\TeX\` will automatically leave more space after a period at the end of a sentence than between words in a sentence. But since `\TeX\` has no way of understanding what you write, it has to make a guess at what is 'a period at the end of a sentence'. It guesses that it is any period followed by one or more blanks as long as it is not preceded by a capital letter, as in D. E. Knuth. Note that `\TeX\` provides an ordinary interword space after each of the two initials, and a larger, intersentence, space after the final period.

This rule of thumb causes difficulty when a sentence ends in a capital letter, such as this ONE.

See? You can overcome it by inserting the `{\bf do nothing}` command `\ttb{null}` just prior to the `PERIOD\`. There. That does it.

This rule of thumb also causes difficulty with abbreviations, such as `refs. to comp.lit. or phys.ed. courses`. You overcome this by following such abbreviation periods by a `{\bf forced blank}`, e.g., as in `refs.\ to comp.lit.\ or phys.ed.\ courses`. See the difference?

This rule of thumb causes difficulty with the unhappy habit and unfortunate standard rule of 'putting the period at the end of a quote `{\it inside}` the quote.' The remedy is simple: 'Put the period outside'. But if you must keep it inside, 'follow the quote with two forced blanks.' `\ \ There`.

Note that the beginning quotes in the preceding paragraph were typed differently from what you might have expected. Just to get used to this, also type the closing double quote as `'` even though `"` will give the same thing.

`\heading{simple math}`

All math within text is enclosed within dollar signs, even when it is just one letter or symbol, such as `a`, `b`, or `c`, and certainly for things like `$E=mc^2$`, or `$x_i+y_i={3\over4}$`, or `a^{b^c}`, or `x^{α_1}`, or or `$\int_{-1}^1x\,dx=0$`, or `$\sum_{i=0}^nc^i=(c^{n+1}-1)/(c-1)$` for all `$n\ge0$`.

It is worth studying these examples in some detail. I have not left any spaces around symbols; `\TeX\` takes care of that, usually. But `\TeX\` cannot understand math, so sometimes you may have to control spacing, as I have done in the integral, by putting a bit of space (via `{\tt\char'134,}`) between the `x` and the `dx`. Also, one must use braces to indicate the extent of subscripts and superscripts, but only if they involve more than one character. Braces are also required to control what `\ttb{over}` puts on top and below the bar.

For a complete, ordered list of all the available math symbols, see Sam Bent's `\TeX\ Reference Card` in the appendix.

The inexperienced typist will bemoan having to type all those dollar signs. Although I do touch-typing, sort of, I have found it more convenient to type the dollar sign (and some other signs) piano-fashion, i.e., by hitting appropriate keys simultaneously with the `\hfill\break\vfill\ejct`

\advance\pageno by 1
 \noindent same hand; in this example, it's the left hand, with the little finger hitting the shift key while the middle finger simultaneously hits the 4/\\$ key.

\heading{display math}
 Displayed math is enclosed within {\bf double} dollar signs. Here are some of the earlier math examples, but in double dollar signs:

\$\$
 $E=mc^2$, or $x_i+y_i=\frac{3}{4}$, or a^{b^c} , or x^{α_1} ,
 \$\$ or

\$\$\int_{-1}^1 x \, dx=0, \$\$ or
 \$\$\sum_{i=0}^n c^i = (c^{n+1}-1)/(c-1), \text{ for all } n \ge 0. \$\$

You'll notice that some of the display has become more expansive. For example, the summation sign is bigger, and its limits are above and below it, rather than to the right of it. That's nice since \TeX takes care of such things; you type it the same way, whether in math mode or in display mode, and \TeX makes the appropriate adjustments. You'll also notice that the last comma in the first and second display, and the period in the third display, are now typed before the dollar sign(s) rather than after. Can you guess what would have happened otherwise?

On the other hand, notice what happened to the ordinary word "or" and the text "for all". \TeX treats them, not as words, but as a sequence of math symbols, hence puts them into italics, gives them a funny spacing and ignores entirely the interword space. To have them typeset as ordinary text, you need to switch temporarily to that font, as in the following:

\$\$\sum_{i=0}^n c^i = (c^{n+1}-1)/(c-1), \{\rm for all\} n \ge 0. \$\$

This now uses ordinary (roman) type for these two words, but still isn't right since, being inside display mode, the interword spacing is ignored. So you have to enforce it by using the {\bf forced blank}:

\$\$\sum_{i=0}^n c^i = (c^{n+1}-1)/(c-1), \{\rm for\ all\ } n \ge 0. \$\$

This is still not quite right since it is nicer to set off that last quantifier from the actual formula. This you do by inserting some (standard) space that \TeX provides with the commands \ttb{quad} and \ttb{qqquad}. \ttb{qqquad} is the right one here:

\$\$\sum_{i=0}^n c^i = (c^{n+1}-1)/(c-1), \qqquad \{\rm for\ all\ } n \ge 0. \$\$

```
\vfill\eject
%%
% here, for the record, is the list of the macros from the file
% format.tex (input initially) that are actually used here.
%
```

```
\def\heading#1{\bigskip\centerline{#1}\smallskip}
\def\sub_{\bf|}
\def\openZ{{\rm Z}\kern-.28em{\rm Z}}
\def\ga{\alpha}
\def\g0{\Omega}
\def\strutdepth{\dp\strutbox}
\def\marginal#1{\strut\vadjust{\kern-\strutdepth%
\top to \strutdepth{
\baselineskip\strutdepth
\vss\llap{\fiverm#1\ } \null}}}}
\def\inpro#1{\langle#1\rangle}
\def\emember#1{\ifrememberchapter % subsequent percents prevent unwanted blanks
\immediate\write\chpaux{\def\string#1\chapterno{\chapterno.\formeq}}\fi%
\immediate\write\aux{\def\string#1\formeq}}%
\global\edef#1{\formeq}}
\newif\ifrememberchapter\newread\testfl
\def\inputifthere#1{\immediate\openin\testfl=#1
\ifeof\testfl\message{(#1 does not yet exist)}\else\input#1\fi\closein\testfl}
\inputifthere{\jobname.aux}\newwrite\aux\immediate\openout\aux=\jobname.aux
```

```
%
%%
```

\advance\pageno by 1

\heading{alignment}

\TeX\ really begins to pay off when it comes to the alignment of text. Here are some standard {\it examples}:

```
$$
H(x):=\cases{x&if $0\le x\le 1$;\cr
            2-x&if $1\le x\le 2$;\cr
            0&otherwise.\cr}
$$
```

This is an example of the `\ttb{cases}` statement. The description of each line has two {\bf fields}. The first field goes from the ‘beginning’ of the description to the {\bf ampersand} `\tt&`; it is typeset automatically as {\it math}. The second field goes from the ampersand to the {\bf carriage return} `\ttb{cr}`, and it is typeset automatically as {\it text}; hence any math in it must be embraced by dollar signs. The description of the first line begins right at the opening brace of the `\ttb{cases}` statement. The description of subsequent lines begins right after the `\ttb{cr}` of the preceding line. (It doesn’t matter how many actual lines you use in your {\tt.tex} file to describe the lines to be displayed in the cases statement.)

If you prefer somewhat more space between the cases, use `\tt\char'134 noalign`, as in this modification, which also shows how you could get the material in the first fields centered:

```
$$
H(x):=\cases{\hfil x\hfil&if $0\le x\le 1$;\cr\noalign{\vskip6pt}
            2-x&if $1\le x\le 2$;\cr\noalign{\vskip6pt}
            \hfil0\hfil&otherwise.\cr}
$$
```

The next example shows how to align several lines of displayed equations. The description of each line is terminated again by the carriage return, and, in each description, the point of alignment is again marked by an ampersand.

```
$$\eqalign{M{*}'f&=M{*}(M\sub{*})c\cr
&=M{*}(c{*})M\sub}\cr
&=(M{*})c{*}M\sub=f{*}'M.\cr}$$
```

Incidentally, the asterisks have all been em‘brace’d here to prevent \TeX\ from embedding them in extra space (a thing it would do to any symbol it recognizes as a binary operation sign).

Many math alignments can be handled by using `\bf matrices`. Here is the standard example:

```
$$
A=\pmatrix{a_{11}&a_{12}&\ldots&a_{1n}\cr
            a_{21}&a_{22}&\ldots&a_{2n}\cr
            \vdots&\vdots&\ddots&\vdots\cr
            a_{m1}&a_{m2}&\ldots&a_{mn}\cr}
\leqno{(14)}$$
```

in which you also see various ways of entering the equivalent of `\bf etc` into math formulas as well as an equation number.

The macro `\ttb{pmatrix}` provides its own fences. If you prefer some other kind, you use `\ttb{matrix}` together with `\ttb{left}` and `\ttb{right}`, as in

```
$$\left[\matrix{\rm A&\cal B&\it C\cr
                \bf D&\sl F&\Gamma\cr}\right],\leqno{(14)'}$$
```

\vfill\ejct


```

\advance\pageno by 1
\def\sumZ{\sum_{j\in\openZ^d}} % <<<< locally useful abbreviation
\bigskip\hbox to \hsize{\hfill\vbox{
\halign{#\&\quad##$\quad\&\hfil$\hfil&\$;\#;\$&\hfil$\hfil\cr
for $\f\in\Pi_M$,&M{*}'f=f{*}'M&=&f\sumZ M(j)&-\&\sumZ (f-f(\cdot-j))M(j)\cr
\noalign{\medskip}
&&\in&f&+\&\Pi_{\langle\{\rm deg\}f\rangle}\cr}}
\hfill}\bigskip

```

I have to admit that getting an alignment of such complexity right takes some doing and some patience, and some struggle with TeX's often inscrutable error messages.

This particular example might actually be easier to handle with the `\ttb{matrix}` command, which gives:

```

$$\matrix{
\hbox{for\ }f\in\Pi_M,\ \ \&M{*}'f=f{*}'M&=&f\sumZ M(j)&-\&\sumZ\big(f-f
(\cdot-j)\big)M(j)\cr \cr &&\in&f&+\&\Pi_{\langle\{\rm deg\}f\rangle}\cr}$$

```

Here is a simple use of `\ttb{halign}` which also demonstrates the use of `\ttb{hbox}` and `\ttb{vbox}`, well, of its cousin `\ttb{vtop}` (see the section on boxes for details).

```

\medskip
\hbox to \hsize{\hfill\vtop{
\halign{#\hfill\cr Michael G. Crandall\cr Department of
Mathematics\cr University of Wisconsin-Madison\cr Madison WI 53706\cr}}
\quad and \quad \vtop{
\halign{#\hfill\cr Pierre-Louis Lions\cr Ceremade\cr
Universit'e Paris-Dauphine\cr Place de Lattre de Tassigny\cr
75775 Paris Cedex 16\cr}}\hfill}
\medskip

```

The function served here by `\ttb{halign}` is to permit the name-and-address information to be entered in line, separated only by the carriage returns, and to make sure that all these lines are left-adjusted, and to supply a box that is exactly large enough to contain all that information, hence makes it possible to align these two boxes properly on the page. But it makes it also very easy to `{\bf center}` the information instead, by adding another `\ttb{hfill}` to the template in `\ttb{halign}`:

```

\medskip
\hbox to \hsize{\hfill\vtop{
\halign{\hfill#\hfill\cr Michael G. Crandall\cr Department of
Mathematics\cr University of Wisconsin-Madison\cr Madison WI 53706\cr}}
\quad and \quad \vtop{
\halign{\hfill#\hfill\cr Pierre-Louis Lions\cr Ceremade\cr
Universit'e Paris-Dauphine\cr Place de Lattre de Tassigny\cr
75775 Paris Cedex 16\cr}}\hfill}
\medskip

```

Table construction is also easily done with the aid of `\ttb{halign}`. Here is a simple `{\bf table}`.

```

\bigskip \centerline{\bf Table 1}
\medskip
\hbox to \hsize{\hfill
\vbox{\halign{\hfill#\hfill&\vrule#\&\hfill#\hfill\cr
$x$$$&\sin(x)$\cr \noalign{\hrule}
0&0\cr
$\pi/2$$$&1\cr}}\hfill} \bigskip

```

Note the use of rules to draw lines. Note in particular how that vertical line is drawn for each line with the aid of the second field in the pattern. Note that the pattern description for that second field must contain a sharp even though you have no intention of ever putting anything into that field (other than the `\ttb{vrule}` specified by the pattern).

But it's a miserable table otherwise. The heading line and the first line in the table are right on top of each other, yet there's some skip between the two lines in the table, and the columns are just big enough to contain the widest item in each.

```
\vfill\ejct
```



```
\advance\pageno by 1
\hoffset=2cm\voffset=-1cm
{\hsize=15truecm\vsize=24truecm
\heading{page layout}
The size of the overall area on a page to be filled with print can be
controlled by setting \ttb{hsize} and \ttb{vsize}.
```

The placement of this rectangle of printed text is controlled by setting \ttb{hoffset} and \ttb{voffset}.

```
{
\parskip=10pt The space between paragraphs is
controlled by \ttb{parskip}.
```

```
\parindent=3cm The amount of indentation is
controlled by \ttb{parindent}.
```

```
\baselineskip=18pt
The amount of space between lines can be
controlled by setting \ttb{baselineskip}, as I have just done. All
these parameters have default values. You set them only to change those default
values. Their new values are used as soon as you set them and until you come to
the end of the current grouping. After that, they revert to what they were
before entering that grouping.\par
}
```

```
{\narrower\indent % will increase the margins on either side of a grouping
If you want to restrict these changes to a particular part of the text, put that
text into braces that also embrace those changes, as I have done for the
preceding three paragraphs (and also for the entire material on this page).
\par}
```

```
{\narrower\narrower\noindent The preceding paragraph has had its margins
widened by use of \ttb{narrower}. Also, the indentation for the present
paragraph has been suppressed by a \ttb{noindent}, and the paragraph
made yet narrower by a \ttb{narrower\char'134 narrower}. But, in
each case, the effect of the narrowing command has been restricted to the
paragraph by embracing the {\it completed} paragraph.\par}
```

It is important to note that parameters that only affect entire paragraphs (such as \ttb{baselineskip} or \ttb{narrower}) only affect those paragraphs that are completed before the end of the grouping within which they occur is reached. For {\it example}, leaving off the \ttb{par} near the end of the description of the preceding paragraph would have prevented the \ttb{narrower\char'134 narrower} at its beginning from taking effect.

```
\parindent=5pt
\hang \ttb{parindent} has just been made quite small. Also,
\ttb{hang} has been used to indent the entire paragraph. The amount of
indenting is determined by the current value of \ttb{parindent}.
```

```
\item{${\bullet}$} You can get the same effect with \ttb{item}, except that
\ttb{item} gives you the opportunity to put something to the left
of the indent on the first line, as I have done here.
```

Note the effects of

```
\leftline{a \ttb{leftline}}
\centerline{a \ttb{centerline}}
\rightline{and a \ttb{rightline}}
\line{and of a \ttb{line}\hfill with \hfill some\hfill \ttb{hfill}}
}
\vfill\eject
```


`\advance\pageno by 1`
`\noindent` use an argument because you see the symbols `{\tt\#1}` right between the name of the macro and the opening brace of its 'body'. Also, you see within the body the symbols `{\tt\#1}` repeated right at the spot where the material that makes up the argument is to be placed.

Here is how such a macro is understood by `\TeX`. After `\TeX` has read the macro's name (`\TeX` will know that the name is `{\tt heading}` and not something longer because the character following the `{\tt g}` in `\ttb{heading}` either is a blank or else a non-letter), `\TeX` will look up the definition, find that it requires an argument and now look for it. It expects to find an opening brace as the next character. If that is indeed the case, it will take everything between this opening brace and the `{\it corresponding}` closing brace as the argument. In this particular example, this means that everything between that opening and corresponding closing brace will end up centered, with the centerline preceded and followed by some vertical space.

`\TeX` also allows for a shortcut. If an argument consists of just one character, then it is not necessary to enclose it in braces. In other words, if `\TeX` does not find an opening brace as the next character, then it takes that next character as the whole argument.

A macro may have up to nine separate arguments. In the definition of the macro, they are listed, between the macro's name and the opening brace of the macro's 'body', as `{\tt\#1\#2...}`. They should also (but don't have to) appear at least once inside the macro's body, exactly at the spot at which the material that forms the argument is to appear. The intent is to have the macro provide a template with certain places left open, to be filled in with particulars when the macro is actually used.

At first glance, this list of `{\tt\#1\#2\#3...}` between macro name and body looks a bit silly; why not simply say `{\tt 5}` if five arguments are expected? But this is yet another cleverness of `\TeX`. For, you are permitted to put any one character after each of those numbers, for example, `{\tt \#1,\#2\|#3\#4/...}`, and these very characters are used later by `\TeX` to decide when one argument ends and the next one begins. For example, I have a friend (not a piano player) who hates typing dollar signs. He has a macro that he defined as follows:

```
\ttb{def\ttb{m\#1:\ttb{\$ \#1 \$}}
```

`\noindent` If he has to type something in math-mode, e.g. $\alpha=1/\gamma$, he would type

```
\ttb{m\ttb{alpha=1/\char'134gamma}:}
```

`\noindent` `\TeX` will pick up the `{\tt m}` as the name of a macro, look it up and find that it has one argument and that the extent of the argument is all the stuff following that `\ttb{m}` until it comes across a colon. So it picks up all that stuff and, following instructions, puts it between dollar signs and then processes it in the usual way. The colon itself will not be printed; it was used up as the `{\bf delimiter}` or `{\bf terminator}` of the first (and only) argument of the macro.

What is he going to do when his math-stuff contains a colon, e.g., he wants to type $\{x: f(x)=0\}$? Then he hides that colon in braces! I.e., he types

```
\ttb{m\char'134\char'173x\ttb{:} f(x)=0\char'134\char'175:}
```

`\noindent` For, as `\TeX` reads the stuff following `\ttb{m}`, it skips over any groupings, i.e. over any stuff between braces in its search for the delimiting or terminating colon.

```
\vfill\ejct
```

```
\advance\pageno by 1
\beginsection{including and labelling figures}
```

Here are the definitions of `\ttb{gridbox}` and `\ttb{point}`. These macros are useful for the precise placement of material. I illustrate their use in labelling a simple graph. The actual graph is specified with the aid of PostScript, the same language that is used on many Laser writers to print TeX. Different printers or different converters from dvi-file to printer file would require different statements to get the graphic combined with text. A popular (free) means for placing figures into TeX text is `\ttb{epsf}`.

```
\def\tick#1/{\vrule width\gridwidth heightOpt depth#1truecm}
\def\nexttick#1#2/{\hbox to #1truecm{\hfil\tick#2/}}
\newcount\tickcount
\newdimen\gridwidth \gridwidth=0.4pt
\newif\ifshowgrid \showgridtrue
\def\point(#1,#2)#3{\vbox to Opt{\kern#1truecm
  \hbox{\kern#2truecm{#3}}\vss}\nointerlineskip}

\def\gridbox#1/#2/#3{
\hbox to #1truecm{
#3
\ifshowgrid\tickcount=0
  \loop
  \vbox to Opt{\kern\tickcount truecm\hrule width#2truecm height\gridwidth\vss}
  \nointerlineskip \advance\tickcount by 1
  \ifdim\tickcount pt<#1pt\repeat % Note: #1, #2 need not be integers
  \hbox to Opt{\tickcount=1\tick#1/
  \loop\ifdim\tickcount pt<#2pt\nexttick1 #1/\advance\tickcount by 1\repeat\hss}
\else \vbox to Opt{\hrule width#2truecm heightOpt\vss}
\fi\vfil}\vfil}
```

We start with the graphic, put into the gridbox, with the grid drawn, to help us later on to place the labels.

```
\bigskip\hbox{
  \vbox{ \gridbox3.5/6.6/{\point(0,0){\special{ps: plotfile intsamp.pls}}
  }}\bigskip
```

Next we take a stab at placing the labels.

```
\bigskip\hbox{
  \vbox{ \gridbox3.5/6.6/{\point(0,0){\special{ps: plotfile intsamp.pls}}
  \point(3,.5){$A$}\point(3.2,6.1){$C$}\point(.2,2.45){$R$} %<<<<<< LABELS
  }}\bigskip
```

That looks ok, except that the `R` should be a little bit more to right and down, say .35cm down and .4 cm to the right. So we change these 'coordinates'. We also get rid of the grid (which we can do by saying `\ttb{showgridfalse}` or by saying `\ttb{gridwidth=0pt}`), center the whole figure, and put a Figure description underneath.

```
\showgridfalse
\bigskip\hbox to \hsize{
  \hfil\vbox{ \gridbox3.5/6.6/{\point(0,0){\special{ps: plotfile intsamp.pls}}
  \point(3,.5){$A$}\point(3.2,6.1){$C$}\point(.55,2.9){$R$}%<<<<<< LABELS
  }}\hfil }
\centerline{Figure 3. A labelled graph} \bigskip
```

There, that wasn't so bad. -- One would usually work on this in a separate file and only insert the finished material appropriately, perhaps using `\ttb{midinsert}` or `\ttb{topinsert}`.

```
\vfill\eject
```

```
% Here is the plotfile intsamp.pls used on this page. you'd have to remove
% the leading percent signs before using it.
```

```
%% save
%% % 23.625 23.625 scale% this is the right scale (with dvlaser) to make the
%% % given (x,y) exactly cm when using \magstep=1200.
%% 236.25 -236.25 scale % is the right scale for dvips.
%% currentpoint translate
%% .02 setlinewidth 1 setlinejoin 1 setlinecap
%% newpath 1 -3 moveto 3 -1 lineto 6 -3.5 lineto stroke
%% restore
```

```

\advance\pageno by 1

\def\boxit#1{$$\vbox{\hrule\hbox{\vrule\kern10pt
\vbox{\kern10pt\vbox{\hspace 15.6truecm \noindent \strut
#1\strut}\kern10pt}\kern10pt\vrule}\hrule}$$}

\boxit{At times, it is convenient to pack it all in. This is one of those
times, I think. (You might try to modify the macro \ttb{boxit} used here to
get a double box line.)}

\def\keyword#1{${\inpro{#1}$}
\beginsection{sequencing equations and other items}

\newcount\equationno \equationno=0
\def\formeq{\the\equationno}

\def\label#1/{\elabel{#1}////}\newif\ifshowlabel
\def\elabel#1/#2/#3/{\global\advance\equationno by 1 %
\ifx#1\empty\else\emember#1%
\ifshowlabel\marginal{\string#1}\fi\fi%
\ifmmode\leqno{#3(\formeq#2)}\else#3\formeq#2\fi} %<----- switch to \eqno ???

(\label/) Here is the macro \ttb{label}
which is useful for an automatic sequencing of equations
and other items. It is so simple that it wouldn't be hard for you to
modify them to fit the particular needs of a particular paper.

(\label\listequ/) \ttb{label} will increment the value of \ttb{equationno} by
one and then print it out, and will enclose it in parentheses if called in
display-math mode, as in the following.  $$e=mc^{1/2}\label{e}$$

(\label/) In order to refer to these numbers later, you give them names.
For example, I gave the name \ttb{listequ} to the number that starts the
paragraph (\listequ) ({\bf check how I typed the preceding number})
by saying \ttb{label}\ttb{listequ} there.
\equationno=37

(\label/) You can always change \ttb{equationno} to any value you like. For
example, I just changed it to 37 by typing \ttb{equationno=37} prior to this
paragraph.

\showlabeltrue
(\label\memory/) If you find that you have trouble remembering the names
you gave to earlier items (and you don't want to search for it in the {\tt.tex}
file), you can say \ttb{showlabeltrue} and from then on the name you used will
appear, quite small, nearby in the left margin of the printed document.

\showlabelfalse
(\label\gone/) Guess what you type to turn off this feature?

(\label/), (\label/), $\ldots$.

($\infty$) Finally, you may wish to refer to the very equation- or
item-numbers named in this {\tt.tex} file in other files. You would want to
write them as definitions into a file \keyword{filename} which you would then
\ttb{input}\ \keyword{filename} in the other file. For this to work here,
you would first say something like\par
\ttb{rememberchaptertrue} \ttb{newwrite}\ttb{chpaux}\par
\ttb{immediate}\ttb{openout}\ttb{chpaux}=\keyword{filename}\par
\noindent when you make all those other definitions in this section.
The macro \ttb{label} then takes care of the rest.

($\infty+1$) Actually, the way it is set up in this section, the name
saved is given the suffix \ttb{chapterno}, with the assumption that
\ttb{chapterno} is some word identifying this particular file. Furthermore,
the number saved is prefixed by \ttb{chapterno.} . This means that you can
safely use names in the present file without worrying about the fact that
you might use the same name for something else in the other file. Enough
already.
{\def\bye{\vfill\ejct}
\bye
\input boxes
\input errors
}
\bye

```