EXPLORATIONS

Chapter 6

Beginning Programs— Sequential Processing

Now that you have written and revised a home screen program, you are ready to create your first program with the TI-86. This first program is a *sequential* program in which each statement is carried out in order, from first to last.

In this chapter, you will begin using the program editor and learn some shortcuts for entering information into the program editor.

Using the program editor

Before creating a new program, you need to become familiar with the program editor. When you press the [PRGM] key, you will see the menu shown in Figure 6.1. To display existing program names for editing or to create a new program, press [F2] (**EDIT**). (To quit the editor, press [EXIT].)

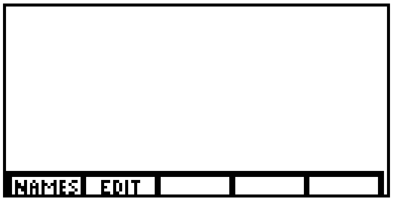

Figure 6.1
Program Editor Starting Menu

Note: If you already have programs in your TI-86, those names are in the menu at the bottom of the screen (see Figure 6.2). To look at or change one of these programs, press the menu key beneath its name. The **MORE** symbol (▸) at the lower right indicates that there are additional items (more programs) in this menu. Press [MORE] to display additional program names.

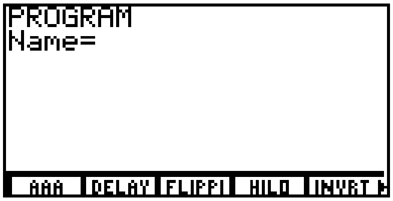
PROGRAM
Name=

Figure 6.2
Names of First Five Programs

Naming a program

To create a new program, type the name of the program at the **Name=** prompt. Type up to eight characters, and then press ENTER.

If you select a name already in use elsewhere, you will get an error message, such as the one in Figure 6.3. Just press F1 (**GOTO**) to go back to the invalid name you entered. You can edit the name already there, or press CLEAR to enter an entirely new name.

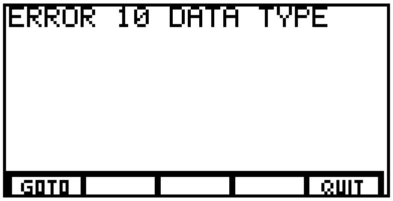
ERROR 10 DATA TYPE

Figure 6.3
Error Message Dialog

For now, create a new program called **MODES** (see Figure 6.4) and press ENTER to begin typing the program.

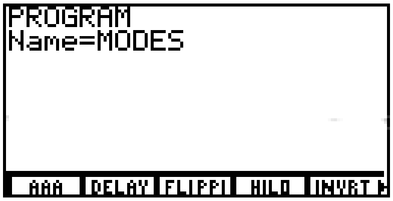
PROGRAM
Name=MODES

Figure 6.4
Naming a New Program

Creating your first program

You are now ready to create a program—a really simple one. Many programs set modes that are very different from what you are accustomed to. You will now write a program (the one you just named **MODES**) to set several modes to your preferences.

The first nine lines of the program below let you customize the default settings. The italicized lines substitute a different value for the defaults. The line `2*yMax→yMax` instructs the calculator to double the default value for **yMax** and store the result into **yMax**. The next line, `-yMax→yMin`, stores the opposite of **yMax** (-6.2) into **yMin**.

This program is a simple example of sequential processing, in which each statement is executed in order, from first to last.

```
:Normal
:LabelOff
:Float
:Degree
:RectC
:Func
:Dec
:RectV
:dxDer1
:ZDecm
:AxesOn
:CoordOn
:2*yMax→yMax
: -yMax→yMin
```

Some novices are reluctant to use the same variable on both sides of the store arrow (→), but you should get used to it, since you will need to do it often. Just keep in mind that there is a logical separation and order to the calculation-and-storing cycle. It is a two-step process: calculate, then store. It is perfectly legal to use the same variable on both sides.

Press [2nd] [MEM] [F3] (**RESET**) [F3] (**DEFAULTS**) to get the TI-86 back into factory mode.

Pasting program keystrokes and keywords

When creating your **MODES** program, you can paste almost every statement into the program from the [MODE] menu.

Once you are in the program editor, press [2nd] [MODE] and observe (Figure 6.5) that nothing is highlighted. When you are in the program editor, you can highlight what you want to type and press [ENTER] to paste it into the program editor.

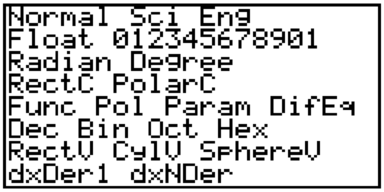

Figure 6.5
Pasting Modes in the Program Editor

It is a lot like cutting and pasting except that you do not have to cut. Whatever is highlighted is considered to be ready to copy. Press [ENTER] to paste.

This is the same procedure you use with the TI-Graph Link at your PC. (See Figure 6.6.)

If you have the TI-Graph Link software, by all means use it as soon as you tire of the TI-86 program editor. But *do* use the editor for a little while just to learn how it works. You might have to make last-second changes away from your PC at some time.

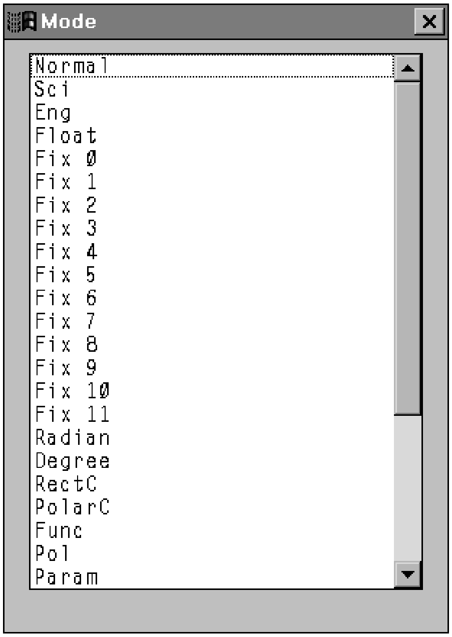

Figure 6.6
Pasting Modes in the TI-Graph Link Software for Windows 95

There are other shortcut menus available to you in the program editor. These also can be pasted. The menu at the bottom of the screen offers thirty-one keywords often used in programming. Use of the menu keys (F1, F2, and so on) makes editing easier, especially pasting from secondary menus.

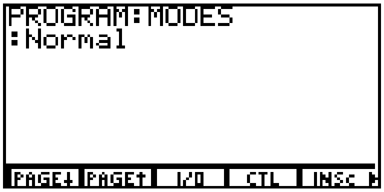
PROGRAM:MODES
:Normal

Figure 6.7
MORE Gives More Editing Options

Press MORE to display **DELc** and **UNDEL** (Figure 6.8), which afford a way to cut and repeatedly paste entire lines. Pressing F3 allows you to paste in a colon.

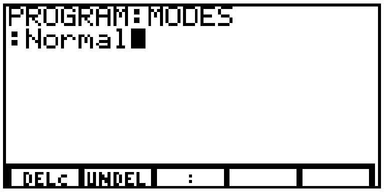
PROGRAM:MODES
:Normal

Figure 6.8
Use F1 and F2 to Cut and Paste

Press F3 (the **Input/Output** key) to see the first five shortcut keywords. You will see a few familiar keywords (Figure 6.9) as well as some less familiar ones that will be discussed later. Press MORE twice to see seven more **I/O** keywords as well as a key for entering the quotation mark (").

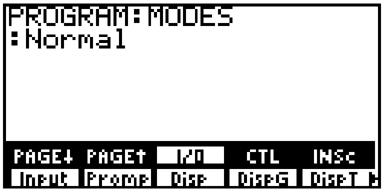
PROGRAM:MODES
:Normal

Figure 6.9
Press MORE Twice to See More Keywords

Note that when you press F3 (**I/O**) (Figure 6.9), the **I/O** menu forces the program editor main menu up a line, giving two tiers of menus. This is good because, if you have to page up or down (**PAGE↑** or **PAGE↓**) to jump toward the top or bottom of your program, you can just press 2nd F1 or 2nd F2 and still have immediate access to the **I/O** menu. More important, you can press 2nd F4 to activate the **CTL** (control) keywords (see Figure 6.10). If you then press MORE, you will see the **Goto** and **Lbl** keywords as well as sixteen other control keywords.

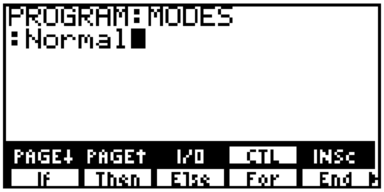
PROGRAM:MODES
:Normal

Figure 6.10
CTL Menu Includes Eighteen Keywords

When you need a keyword that you cannot find in the menus, just type it. Case does not matter except in variable names. You can also press 2nd [CATLG-VARS] to display a menu that offers access to an alphabetized list of all keywords and symbols on the TI-86 keyboard (see Figure 6.11).

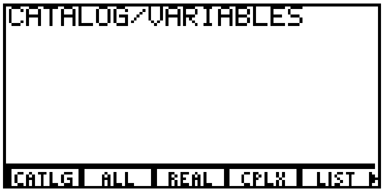
CATALOG/VARIABLES

Figure 6.11
[CATLG-VARS] Menu

Press F1 (**CATLG**) to see this list (Figure 6.12).

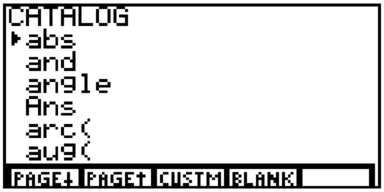

Figure 6.12
The TI-86 Catalog

If you press a key with a blue alphabetic letter above it (for example the 7 key), the TI-86 jumps to that part of the alphabetized catalog (since **L** is the alpha letter above the 7 key, pressing 7 takes you to the **L**s in the catalog, as shown in Figure 6.13).

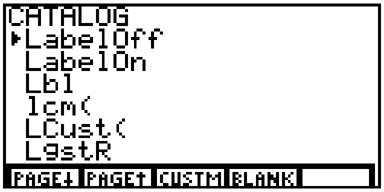

Figure 6.13
Jump to Ls

By using the cursor keys and page up (F2) and page down (F1) keys, you can easily find whatever you are looking for. A handy trick: press **A** and then press ▲ and possibly the F2 (**PAGE↑**) key to find any symbol (for example, the *greater than or equal to* key, ≥).

Creating custom menus

The TI-86 has a customizable menu feature, into whose slots you may put almost any keyword or keystroke you want and use it later for pasting, as you just did from the **CATALOG**. To activate it, press CUSTOM. This works almost anywhere you happen to be, including in the program editor, home screen, graph screen, and so on.

You will see nothing at all in the bottom line of menu keys, unless you have already defined some custom menu items for yourself.

In Figure 6.14, the first five key slots of a custom menu are displayed. To see five more, press MORE. You can make or alter your custom menu to fit your programming needs.

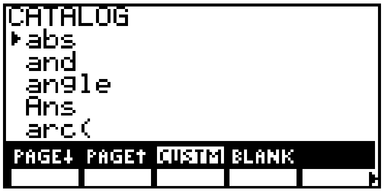

Figure 6.14
Editing Your CUSTM Menu

To edit your **CUSTM** menu, press 2nd [CATLG-VARS], then F1 (**CATLG**), then F3 (**CUSTM**). Move ▸ (the cursor indicator) to the keyword or symbol you want to have in your custom menu and press the menu key (F1 through F5) that you want to assign to it. Use MORE if needed.

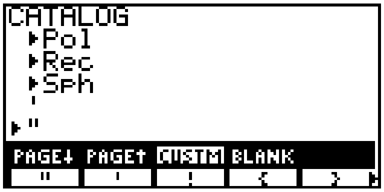

Figure 6.15
Assigning Custom Menu Keys

In Figure 6.15, the first five custom keys have been assigned to special characters not found on the TI-86 keyboard.

After the first five slots are filled, you may press $\boxed{\text{MORE}}$ to fill five more, and again for a total of fifteen. You will want to put commonly-used, hard-to-find items in your menu. The ≠ symbol (*not equal to*) should probably be somewhere in the first five slots, as should ≥ (*greater than or equal to*) and ≤ (*less than or equal to*), ! (the exclamation point, used to compute factorials and add emphasis in quoted strings), and maybe even " (the quotation mark) and ? (question mark). All can easily be found using $\boxed{\text{2nd}}$ [CATLG-VARS] [A] $\boxed{\blacktriangle}$ (catalog, A, cursor up).

The $\boxed{\text{CUSTOM}}$ menu may be adjusted from the home screen easily enough, but if you learn how to do it in a program, you could conceivably have several $\boxed{\text{CUSTOM}}$ menus at your disposal. Only one fifteen-item menu may be active at a time, but with programming, you can easily choose the one you want for the current task. Here is how to make one to assign meaning to the first eight menu keys in the $\boxed{\text{CUSTOM}}$ menu. Note that special characters, program names (**Cust2**), and messages (**Hi!**) may be assigned to the menu keys.

```
    :LCust(1,"{",2,"}",3,"≤",4,"≥",5,"≠",6,"!",7,"Cust2",8,"Hi!")
```

That is easy enough, particularly with TI-Graph Link software, to almost discourage using $\boxed{\text{CUSTOM}}$ from the home screen! When the program is run, the $\boxed{\text{CUSTOM}}$ menu will look like Figure 6.16.

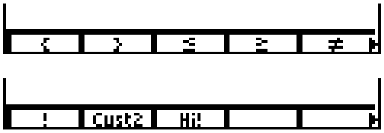

Figure 6.16
The Program-Generated $\boxed{\text{CUSTOM}}$ Menu

Summary

If you do not have the TI-Graph Link software, you will find the pasting feature from the function key menus and the catalog to be indispensable tools in writing programs. They work so well that you might convince yourself that you do not need the Graph Link. You do!

You have finished your first real program, an example of a sequential program. Such programs can be useful, but they do not begin to tap the true power of programming. This will occur in the next chapter, when you learn about *conditional processing* and *looping*. But you have not necessarily written your last sequential program. Later you may find a need for *initialization routines*, which get the calculator ready to run a new program (or system of programs) for the first time. Such programs are likely to be sequential.