

## Bonus Chapter 7

# Controlling Program Input and Output

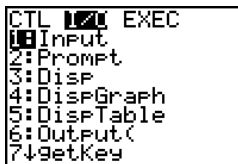
---

### *In This Chapter*

- ▶ Entering program input and output commands
  - ▶ Using input commands (Input, Prompt)
  - ▶ Using output commands (Disp, Output)
  - ▶ Using a program to graph functions or stat plots
- 

**P**rogram *input* is information that the program requests from the program user. *Program output* is information passed from the program back to the program user. This chapter tells you how to get a program to shuttle information back and forth between the program and the program user.

The Program I/O menu, which houses the input and output commands, is available only when you're using the Program editor to create a new program or to edit an existing program. A picture of the Program I/O menu appears in Figure B7-1. Creating and editing programs are explained in Chapters B5 and B6.



**Figure B7-1:** The Program I/O menu.

---

## *Using Input Commands*

The **Input** and **Prompt** commands are used in a program to solicit information from the program user. The **Input** command asks the

user for the value of only one variable and allows the program to briefly describe the variable it is requesting. The **Prompt** command asks the user for the value of one or more variables, but does not allow for a description of the variable other than its name.

## Using the **Input** command

The syntax for using the **Input** command to request the program user to assign a value to a *variable* is: **Input** "text",*variable*. The *text*, which must be in quotes, offers the program user a description of what is being requested by this command. The *text* and the *variable* must be separated by a comma. Note that there is no space between the comma and the *variable*, as in the first picture in Figure B7-2.



Press **[ALPHA][+]** to insert a quotation mark.

When the program is executed, the program displays the *text* on the Home screen and waits for the program user to enter a number and press **[ENTER]**. This is illustrated at the top of the third picture in Figure B7-2. The number entered by the user is then stored in the *variable* specified by the **Input** command.

Each line of the Home screen can accommodate a maximum of 16 characters. Often, this isn't enough space for the **Input** command to display the *text* and for the program user to enter the value of the *variable*. If this is the case, you may want to precede the **Input** command with a **Disp** command describing the value that the user must enter. When you do so, the syntax for the **Input** command is simply: **Input** *variable*, as in the second picture in Figure B7-2.

When the program is executed, it displays the description given in the **Disp** command, and then prompt the program user for a value by displaying a question mark, as in the second half of the third picture in Figure B7-2. Using the **Disp** command is discussed later in this chapter.

<pre>PROGRAM: INPUT : Input "NUM=",N :</pre>	<pre>PROGRAM: INPUT2 :Disp "ENTER INT &lt; 20" : Input N :</pre>	<pre>PrgrmINPUT NUM=10           Done PrgrmINPUT2 ENTER INT &lt; 20 ?10             Done</pre>
--	--	--

**Figure B7-2:** Using the **Input** command.

## Using the Prompt command

The **Prompt** command asks the program user to assign values to one or more *variables*. The syntax for using the **Prompt** command is: **Prompt** *variable1,variable2,...,variable n*. Commas separate the *variables* and there is no space between the comma and the next *variable*, as in the first picture in Figure B7-3.

When the program is executed, the program displays the first *variable* followed by an equal sign and a question mark and waits for the program user to enter a number. It then does the same for the next *variable*, and so on, as in the second picture in Figure B7-3. The numbers entered by the user are then stored in the *variable* specified by the **Prompt** command.



The Window variables **Xmin**, **Xmax**, **Ymin**, and **Ymax** are housed in the Variables Window menu. To access this menu, press **VAR****S****1**.

<pre>PROGRAM:PROMPT :Disp "SET THE W INDOW" :Prompt Xmin,Xma x,Ymin,Ymax :</pre>	<pre>PRGM:PROMPT SET THE WINDOW Xmin=?-5 Xmax=?5 Ymin=?-5 Ymax=?5 Done</pre>
--	--

**Figure B7-3:** Using the **Prompt** command.

## Using Output Commands

The **Disp** and **Output** commands are used by a program to display text messages and values. The **Disp** command is capable of displaying more than one piece of information, and the **Output** command allows the program to place text or a value, but not both, at a predetermined location on the Home screen. The syntax for using these commands is explained later in this chapter.

### Using a program to write text

The **Disp** and **Output** commands, which are explained in the next two subsections, are used to get a program to display text on the Home screen. Because each line of the Home screen can accommodate up to 16 characters, the wise programmer will limit all text items to no more than 16 characters. A space counts as one character.

The first picture in Figure B7-4 shows an example of a program that displays the text “PRESS ENTER TO CONTINUE” in two ways. The first **Disp** command displays the whole text, in spite of the fact that it contains more than 16 characters. The next two **Disp** commands break the text into two parts, each of which contains less than 16 characters.

The output of the program in the first picture in Figure B7-4 is shown in the second picture in Figure B7-4. The ellipsis at the end of the second line in this picture indicates that the calculator could not display the whole line. (The calculator does not understand “wrap around.”) And worse than that, you cannot use  $\leftarrow$   $\rightarrow$   $\uparrow$   $\downarrow$  to see what comes after that ellipsis. The remaining lines of this picture illustrate the solution to this problem.



When programming the calculator to output text, limit all text items to 16 characters. A space counts as one character. If necessary, break the text into two or more text items that are consecutively displayed.

PROGRAM:TEXT :Disp "PRESS ENT ER TO CONTINUE" :Disp "PRESS ENT ER" :Disp "TO CONTIN UE"	PRGMTEXT PRESS ENTER TO ... PRESS ENTER TO CONTINUE Done
---	--

**Figure B7-4:** Limiting text items to 16 characters.

## Using the Disp command

The syntax for using the **Disp** command to have a program display *text* and *values* is: **Disp** *item1,item2,...,item n* where *item* is either *text* or a *value*. The *items* appearing after this command are separated by commas with no spaces inserted after each comma. *Text items* must be in quotes, and *value items* can be arithmetic expressions, as in the last two lines of the first picture in Figure B7-5.

When a program executes a **Disp** command, it places each *item* following the command on a separate line; *text items* are left justified and *value items* are right justified, as in the second picture in Figure B7-5.



The Home screen, where program output is displayed, can accommodate up to eight lines. If the **disp** command in your program is going to result in more than eight lines, consider breaking it into several **disp** commands separated by the **Pause** command. The **Pause** command is explained in the previous chapter.



The **Disp** command followed by an empty *text item* can be used to make a program skip a line on the Home screen, as in the fifth line of the program in the first picture in Figure B7-5.

<pre>PROGRAM:DISPLAY :ClrHome :Disp "ENTER INT &lt; 20" :Input N :Disp "" :Disp "INTEGER + 5 = ",N+5</pre>	<pre>ENTER INT &lt; 20 ?10  INTEGER + 5 =      15                    Done</pre>
--	---

**Figure B7-5:** Using the **Disp** command.

---

## *Using the Output command*

The syntax for using the **Output** command to have a program display *text* or a *value* at a specified location on the Home screen is: **Output**(*line,column,item*). The calculator supplies the first parenthesis; you must supply the last parenthesis. There are no spaces inserted after the commas.

The Home screen contains 8 lines and 16 columns. The *item* displayed by this command can be a *text item* or a *value item*. *Text items* must be in quotes, and a *value item* can be an arithmetic expression, as in the last four lines of the first picture in Figure B7-6. The program output resulting from executing this program is illustrated in the second picture in Figure B7-6.

<pre>PROGRAM:OUTPUT :Disp "ENTER INT &lt; 20" :Input N :Output(5,1,"INT EGER + 5 = ") :Output(5,15,N+5 )</pre>	<pre>ENTER INT &lt; 20 ?10  INTEGER + 5 = 15</pre>
--	--

**Figure B7-6:** Using the **Input** command.

---

## *Using a Program to Display a Graph*

When the **Input** command isn't followed by a variable name, it graphs the functions that are turned on in the Y= editor.

An example of a program that uses the **Input** command to graph functions appears in the first picture in Figure B7-7. In this program,

the **PlotsOff** command turns off all stat plots and the **FnOff** command turns off all functions in the Y= editor. The **FnOn 1,2** command turns on the first two functions in the Y= editor so that only these two functions are graphed. The **ZStandard** command tells the calculator to graph these two functions in the standard viewing window where  $-10 \leq x \leq 10$  and  $-10 \leq y \leq 10$ . The **Input** command, because it isn't followed by a variable, tells the calculator to graph these two functions.

When the program in the first picture in Figure B7-7 reaches the **Input** command, it displays the graph of the first two functions housed in the Y= editor, places the free moving cursor at the center of the graph, and displays the coordinates of the cursor, provided that **CoordOn** is selected in the Format menu. You can use the  $\leftarrow$   $\rightarrow$   $\uparrow$   $\downarrow$  to move the cursor, but it doesn't trace the graph in this program, as in the second picture in Figure B7-7.

After a program graphs functions using the **Input** command, the program user will see a moving, broken vertical line in the upper-right corner of the graph, as is illustrated in the second picture in Figure B7-7. This indicates that the program user must press **ENTER** in order to tell the program that he or she is finished looking at the graph. If you don't think that the program user will realize that he or she must do so, then before the **Input** command, have your program tell the user to press **ENTER** after viewing the graph. The next section tells you how to do so.

When a program user presses **ENTER** after viewing the graphs displayed by the **Input** command, control will be returned to the program. If the **Input** command was the last command in the program, then control will be returned to the calculator where the user can, if desired, use any of the techniques discussed in Parts IV and V to investigate the graph. If the **Input** command was not the last command in the program, then the program will continue by executing the next command in the program. In the program illustrated in Figure B7-7, when the program user presses **ENTER** after viewing the graph, the program calls and executes program **WHATEVER**.



Commands such as **PlotsOff**, **FnOff**, and **ZStandard** can be entered in your program from the Catalog menu.

PROGRAM:GRAPH :PlotsOff :FnOff :FnOn 1,2 :ZStandard :Input :PRgmWHATEVER	
--	--

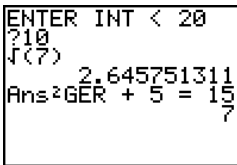
**Figure B7-7:** Using the **Input** command to display a graph.

## Housekeeping Issues

Because programs display their output on the Home screen, it is a good idea to have your program clear the Home screen before the output is displayed. This is done by inserting the **ClrHome** I/O command in the program before the commands used to display the output, as in the program in the first picture in Figure B7-5.

When the **Disp** I/O command is used to display the program output, it isn't necessary to clear the Home screen at the end of the program. After the program is executed, the calculator uses the next available line on the Home screen to evaluate any arithmetic expressions or to execute any commands you enter.

However, when the **Output** I/O command is used to display program output, it is wise to have the program end by clearing the Home screen. Clearing the Home screen is necessary in this situation because the calculator may type over the **Output** item when you use it to evaluate an arithmetic expression or execute a command after exiting the program, as in Figure B7-8. This figure shows what happens when you use the calculator after executing the program in Figure B7-6.



**Figure B7-8:** The consequence of not clearing the Home screen.

---

Because you want to give the program user a chance to view any output before clearing the Home screen from a program, place the **Pause** control command before the **ClrHome** I/O command in the program. (The **Pause** control command is discussed in Chapter B6.)

Better yet, put the **CLRHOME** program in the first picture in Figure B7-9 on your calculator, and have your program call it whenever you want your program to allow the program user to view the program output before the program clears the Home screen. Calling an external program from within a program is discussed in Chapter B6. The second picture in Figure B7-9 illustrates what happens when the **prgm CLRHOME** command is placed at the end of a program like that

shown in Figure B7-6: The program invites the user to press **ENTER**, and when the user does so, the program clears the Home screen.

<pre>PROGRAM:CLRHOME :Output(7,3,"PRE SS ENTER") :Output(8,3,"TO CONTINUE") :Pause :ClrHome</pre>	<pre>ENTER INT &lt; 20 ?10 INTEGER + 5 = 15 PRESS ENTER TO CONTINUE</pre>
---	---

**Figure B7-9:** Using the CLRHOME program to clear the Home screen.

---