

## Using Casio Graphics Calculators

(Some of this document is based on papers prepared by Donald Stover in January 2004.)

This document summarizes calculation and programming operations with many contemporary Casio calculators (among them the fx-9750G Plus, CFX-9850G Plus, CFX-9850GB Plus, CFX-9950GB Plus, FX-9860G and FX-7400G Plus). In what follows, key labels are in bold-face type, with parentheses indicating labels above keys. For example, to access the left square bracket, the keystrokes would be indicated as **SHIFT** **[** (**[**). Although every effort is made to make this section answer all problems related to *Inside Your Calculator*, it may be necessary to refer to your calculator *User's Guide*.

### General Operation

Most multiplications are indicated by simple juxtaposition; the **×** key is needed only between numbers. To indicate negative numbers, use the **(-)** key, not the **=** key, which represents subtraction as in **A - B**. The letters **A** through **Z** appear in red above the keys; to type one, first type the **ALPHA** key. (**SHIFT** **ALPHA** locks in this mode; another **ALPHA** cancels it.) Note that you can enter the letter **x** by pressing the **x, θ, T** key. Notice too that your cursor changes from **\_** to **A**, indicating alpha mode. The functions directly above or above left are obtained with the **SHIFT** key. Thus **ALPHA** **×** represents the letter **s**, while **SHIFT** **×** represents the left curly bracket, **{**. In this document these would be shown as **ALPHA** **×** (**s**) and **SHIFT** **×** (**{**) respectively.

Powers are done with the **^** key. Internal calculations are carried to 15 digits.

The calculator comes on displaying a main menu of icons for 15 different modes. You can later access this screen by pressing the **MENU** key. *Inside Your Calculator* focuses attention on only three of those modes: **PRGM**, **MAT** and **GRAPH**. The **MAT** mode is for matrices, not math. Normal arithmetic calculation is done in the **RUN** mode, not discussed here.

Movement around any screen is accomplished by the cursor movement keys. These are the corresponding (up, down, left, right) corners of the black area located in the mid-right area of the calculator face. The area is labeled **REPLAY** in tiny black-on-black print. To choose a mode on the menu screen, use the cursor movement keys to find that mode and then press **EXE**. Alternatively, simply press the number or letter in the lower right corner of the mode desired. For example, press the **5** key to enter graph mode, the **log** (**B**) key (you don't have to press **ALPHA**) to enter program mode.

Within a mode, various labels usually appear at the bottom of the screen, above some or all of the gray **F1**, **F2**, **F3**, **F4**, **F5** and **F6** keys. (Such displays also appear when you use any of the keys **SHIFT**, **ALPHA**, **OPTN**, **VARS**, or **SHIFT** **VARS** (**PRGM**).) A labeled **F** key may perform a single function, but in many situations it results in a new set of **F** key labels — in effect, a submenu. Using **EXE** to choose a highlighted option on the screen is another way you may reach a submenu.

To back up through layers of submenus, use the **EXIT** key. **SHIFT** **EXIT** (**QUIT**) skips back to the initial screen of the current mode. In highlighting screen options using the cursor

## Inside Your Calculator

movement keys, an arrow  $\downarrow$  at the bottom right screen corner means that you can scroll past the bottom to see more lines. Similarly, in some submenus, the symbol  $\triangleright$  appears above the **F6** key; this means you can use that key to obtain further **F** key labels. To leave the current mode, use **MENU** to return to the main menu.

In most modes, **SHIFT** **MENU** (**SET UP**) lets you choose degrees or radians, fixed or scientific notation, and other options such as whether graph axes show labels. (For the programs of *Inside Your Calculator*, you should choose Deg for Angle and either Norm1 or Norm2 for Display. It is not necessary to key **EXE** after using an **F** key to make a change in a highlighted item of the **SET UP** screen. Either **EXIT** or **EXE** leaves the setup screen, returning you to the current submenu.

### Useful Keys

$\Rightarrow$  This important key allows you to store numbers. For example, **5**  $\Rightarrow$  **X** will store the value **5** in location **X**. Similarly, **Mat B[1,1]**  $\Rightarrow$  **Mat A[3,2]** will store the current value in row **1**, column **1** of **Matrix B** in row **3**, column **2** of **Matrix A**.

**DEL** erases what appears at the current cursor location. Typing replaces what is at the cursor location.

**SHIFT** **DEL** (**INS**) allows you to insert typing at the current cursor location.

**SHIFT**  $\sqrt[n]{\phantom{x}}$  ( $\sqrt[n]{\phantom{x}}$ ) calculates roots. For example, **5** **SHIFT**  $\sqrt[n]{\phantom{x}}$  **32**, calculates the 5<sup>th</sup> root of 32 = 2. Square roots and cube roots are done more easily, however, with **SHIFT**  $x^2$  ( $\sqrt{\phantom{x}}$ ) and **SHIFT**  $\sqrt[3]{\phantom{x}}$  ( $\sqrt[3]{\phantom{x}}$ ) respectively.

**OPTN** **F6** **F4** (**NUM**) **F5** (**Intg**) rounds down. Notice that **Intg** **-3.7** = **-4**. This is the *Greatest Integer Function* in mathematics, also called the *Floor* in computer science.

Now we can turn to programming features.

### To Begin Entering a Program

1. From the Main Menu select (using the **EXE** key) the Program menu, **PRGM**. This will give you a screen titled **Program List** that lists programs entered earlier.
2. Key **F3** to select **NEW**, which brings up a screen titled Program Name. Note that the cursor is now labeled **A**, indicating that it is in alpha mode for typing letters.
3. Type in a program title using one to eight letters. If you want to include a number in your title, first key **ALPHA** to leave that mode.
4. Type **EXE** to begin programming. This produces a screen labeled with the title you entered and you can begin entering program lines.
5. After you enter a program line, key **EXE** to record it and move to the next line. Each time you do this the symbol  $\downarrow$  appears at the end of the line.

### To Leave and Return to a Program You Have Been Editing

You can exit your program at any time using the **EXIT** key. You can then return to it by scrolling to your program name in the **Program List** screen and keying **F2** (**EDIT**).

## To Run a Program Entered in Your Calculator

1. Access the **Program List** screen by choosing **PRGM** in the Main Menu.
2. Scroll to the title of your program.
3. Key **F1** (**EXE**).
4. Depending on your programming, the calculator may stop to ask for input. Key an appropriate number and press the **EXE** key. If your program stops to display partial information, to continue key **EXE**. When the program is complete, the screen will show **- Disp -**. Note that keying **EXE** again will repeat your last line but in **RUN** mode. You can then work with your result, for example, using it as a factor in multiplication. If, however, instead of doing an arithmetic calculation, you press **EXE** again, you will rerun your program. This can be useful if you wish to run the same program with new data.

## Command Location When In Programming Mode

(Because this information is so useful for reference while programming,  
it is printed as a single page in the separate file **CASIOCommands.doc**.  
You can print it from that file.)

If you use your calculator often, you will become accustomed to the location of various commands to be used when programming. The following table should prove useful for beginners and for those who program only occasionally. While these commands are also listed in the *Casio User's Guide* on pages 453-455, here only those related to **Inside Your Calculator** are included. To find a given command, follow the hierarchy from left to right. For example, to key the instruction **For** as you develop a program, you would key **SHIFT** **VAR** **F1** **F6** **F1**. At first this will seem very complicated but it soon becomes much more straightforward.

|                                       | <u>Instruction</u> | <u>Notes</u>   |
|---------------------------------------|--------------------|--|
| <b>SHIFT</b> <b>VAR</b> (PRGM)        |                    |  |
| <b>F1</b> (COM)                       |                    |  |
| <b>F1</b> (If)                        | <b>If</b>          |  |
| <b>F2</b> (Then)                      | <b>Then</b>        |  |
| <b>F3</b> (Else)                      | <b>Else</b>        |  |
| <b>F4</b> (IEnd)                      | <b>IfEnd</b>       |  |
| <b>F6</b> <b>F1</b> (For)             | <b>For</b>         |  |
| <b>F6</b> <b>F2</b> (To)              | <b>To</b>          |  |
| <b>F6</b> <b>F3</b> (Step)            | <b>Step</b>        |  |
| <b>F6</b> <b>F4</b> (Next)            | <b>Next</b>        |  |
| <b>F6</b> <b>F6</b> <b>F1</b> (While) | <b>While</b>       |  |
| <b>F6</b> <b>F6</b> <b>F2</b> (Wend)  | <b>WhileEnd</b>    |  |
| <b>F2</b> (CTRL)                      |                    |  |
| <b>F1</b> (Prog)                      | <b>Prog</b>        | These four instructions<br>are only used in a few<br>appendices. |
| <b>F2</b> (Rtrn)                      | <b>Return</b>      |  |
| <b>F3</b> (Brk)                       | <b>Break</b>       |  |
| <b>F4</b> (Stop)                      | <b>Stop</b>        |  |
| <b>F3</b> (Jump)                      |                    |  |
| <b>F1</b> (Lbl)                       | <b>Label</b>       |  |

## *Inside Your Calculator*

|                                  |                                  |             |
|----------------------------------|----------------------------------|-------------|
| <b>F2</b> (Goto)                 | <b>Goto</b>                      |             |
| <b>F4</b> (?)                    | <b>?</b>                         | Input data  |
| <b>F5</b> ( $\Delta$ )           | <b><math>\Delta</math></b>       | Output data |
| <b>F6 F3</b> (Rel)               |                                  |             |
| <b>F1</b> (=)                    | <b>=</b>                         |             |
| <b>F2</b> ( $\neq$ )             | <b><math>\neq</math></b>         |             |
| <b>F3</b> (>)                    | <b>&gt;</b>                      |             |
| <b>F4</b> (<)                    | <b>&lt;</b>                      |             |
| <b>F5</b> ( $\geq$ )             | <b><math>\geq</math></b>         |             |
| <b>F6</b> ( $\leq$ )             | <b><math>\leq</math></b>         |             |
| <b>F6</b> (SYBL) or <b>ALPHA</b> |                                  |             |
| <b>F1</b> (')                    | <b>'</b>                         |             |
| <b>F2</b> (")                    | <b>"</b>                         |             |
| <b>OPTN</b>                      |                                  |             |
| <b>F2</b> (Mat)                  |                                  | Matrices    |
| <b>F1</b> (Mat)                  | <b>Matrix</b>                    |             |
| <b>F6 F2</b> (Dim)               | <b>Dimension</b>                 |             |
| <b>F6 F4</b> (NUM)               |                                  |             |
| <b>F1</b> (Abs)                  | <b>Absolute Value</b>            |             |
| <b>F5</b> (Intg)                 | <b>Greatest Integer Function</b> |             |
| <b>F6 F6 F4</b> (LOGIC)          |                                  |             |
| <b>F1</b> (And)                  | <b>And</b>                       |             |
| <b>F2</b> (Or)                   | <b>Or</b>                        |             |
| <b>F3</b> (Not)                  | <b>Not</b>                       |             |

### Program Control Structures

Unless otherwise instructed a program proceeds line by line in the order they are entered. Control structures (listed under **SHIFT VARS (PRGM) F1 (COM)** while in editing mode) change this order of program operation in specific ways or stop program operation. In this book we will use many of these keys.

**IEnd** and **WEnd** As you will see in the following examples, these instructions send control back to the preceding control structure. **Ends** are like parentheses. When one loop is embedded inside another, that loop's **End** occurs first.

**If <test>**

**Then <instruction>** A test, for example **If X=5**, determines whether the **Then** instruction is performed. When the test is true (in this case if **X** does equal five) then the following single program line beginning with **Then** is performed. When the test is false, that line is skipped. (No **IfEnd** is necessary in this case although it will not make a difference if it is included.)

**If** <test>  
**Then** <instruction 1>  
    <instruction 2>. . .

**IEnd** In this case you want the **If** test to govern more than one line of code. You then write those program lines between **Then** and **IEnd**.

**For** is a counting instruction. It applies a series of steps one at a time<sup>1</sup> within indicated limits. Here is an example of a **For** loop:

**0**  $\Rightarrow$  **A**  
**For N**  $\Rightarrow$  **1 To 5**  
**A + N**  $\Rightarrow$  **A**

**Next** In this example, before we enter the **For** loop **A** is set equal to **0**. There is only one instruction between the **For** and **Next** lines but there could have been many. The **For** command line establishes a variable, **N**, that will take on the values successively between the numbers that follow. In this case the successive values are **N = 1, 2, 3, 4, and 5**. Each time one of these values of **N** is set the instruction or instructions that follow until **Next** are processed. Thus the **For** loop here is a short way of processing the following five steps:

**0 + 1**  $\Rightarrow$  **A**    'A is now 1  
**1 + 2**  $\Rightarrow$  **A**    'A is now 3  
**3 + 3**  $\Rightarrow$  **A**    'A is now 6  
**6 + 4**  $\Rightarrow$  **A**    'A is now 10  
**10 + 5**  $\Rightarrow$  **A**    'A is now 15

At this point the program leaves the **For** loop and any following statements are processed with **A = 15**. (Note that **N = 5** as well.)

**While** is a loop that is processed until the test included in the **While** statement fails. In other words, the **While** instruction is saying "While this is true, do the following until you reach **Wend**, then repeat the test. When the test fails, jump to the line after the **Wend**." Here is a simple example of a **While** loop:

**1**  $\Rightarrow$  **N**  
**While N < 12**  
    **N × 2**  $\Rightarrow$  **N**  
**Wend**

The instruction in this loop will be processed until **N** is no longer less than **12**. Here are the lines that will be processed:

**1 × 2**  $\Rightarrow$  **N**    'N is now 2  
**2 × 2**  $\Rightarrow$  **N**    'N is now 4  
**4 × 2**  $\Rightarrow$  **N**    'N is now 8  
**8 × 2**  $\Rightarrow$  **N**    'N is now 16

After this final step it is no longer true that **N < 12**. At this point the loop is exited and any instructions following it are processed with **N = 16**.

---

<sup>1</sup> It is possible to modify this by including a step size. Two examples: in **For I**  $\Rightarrow$  **1 To 9 Step 2** **N** would jump 2 each time, performing the loop for **N = 1, 3, 5, 7 and 9**; in **For I**  $\Rightarrow$  **5 To 3 Step -1**, **N** would reduce one each time, processing **N = 5, 4 and 3**.

**Lbl** allows you to set a target line for a **Goto** instruction. Include a number in this line as in **Lbl 1**.

**Goto** sends program control to the designated **Lbl** line. For example, the instruction **Goto 1** would send program control to **Lbl 1**, even if this means jumping out of a loop. Although this and other instructions can be useful, they can create endless loops. In that case you must manually stop program operation by pressing **AC/ON**.<sup>2</sup>

### Input and Output in Programs

Here are the instructions that allow you to enter and retrieve data.

- ? If you want to have the program user enter a number **N** at some point in the program (usually at or near the beginning) enter the line:

**"N" ? ➡ N**

When the program is run, it will stop at that line and display **N?**. You would then type a value and press **EXE**. The program would then continue with **N** set equal to the value you typed..

- Δ This symbol gives a way to present information. If you want to have the program show the value of **N**, simply enter the line:<sup>3</sup>

**NΔ**

This output line is useful for other purposes as well. For example, you can use lines like this entered at various points in your program to print data values as an aid to debugging — programmer's lingo for finding and correcting errors.

### Matrices

First, let's understand what a matrix is. It is an array of numbers arranged in rows and columns.<sup>4</sup> For example, you might have the array:

|          |          |          |          |
|----------|----------|----------|----------|
| <b>2</b> | <b>0</b> | <b>1</b> | <b>3</b> |
| <b>1</b> | <b>0</b> | <b>6</b> | <b>6</b> |
| <b>1</b> | <b>4</b> | <b>9</b> | <b>2</b> |

This is a 3 by 4 matrix. (Note that you always use the order row, then column.) Arbitrarily, we refer to matrix dimensions and matrix entries in the order horizontal row then vertical column. If

---

<sup>2</sup> Early programmers used **Goto** instructions extensively until a computer scientist named Dykstra severely criticized such use because they created programs that looked like spider webs and were very difficult to interpret and debug. Since then the control structure has been rarely used. There are, however, situations when it best serves the programmer's purpose. **Goto** is used only a few times in appropriate places in this text.

<sup>3</sup> Once a program has been run, you can also access any value that occurred in that program by typing its name followed by **EXE**. You could, for example, have omitted an **NΔ** line from a program and after the program has been run type **N EXE**.

<sup>4</sup> Matrices (that's the plural of matrix) are very useful mathematical structures that can be manipulated in many ways by well established rules: by addition and multiplication, for example. In this book we will be working only with single matrices and only to store and change value entries. Those matrix applications appear in Appendix 5-1 and Chapter 9.

## Inside Your Calculator

we name this matrix **Mat A**, we can refer to individual entries. For example **Mat A**[3,4] is 2, the number in the 3rd row and 4th column.

We can also enter or change entries in a matrix. For example, the program line:

**5**  $\Rightarrow$  **Mat A**[1,4]

would change the 3 in that matrix example to 5.

Keying a matrix entry may be confusing at first, because you must use the **MAT** then the **Mat** keys. Here are the key strokes that you would use to enter **Mat A**[1,4] broken down into separate parts:

First to get **Mat**: **OPTN** **F2** (**MAT**) **F1** (**Mat**)

Next to get the **A**: **ALPHA** **X,  $\theta$ , T** (**A**)

And finally to get the [1,4]: **SHIFT** **+** (**[**) **1** **,** **4** **SHIFT** **=** (**]**)

### Preparing a Matrix for Program Use

Apparently there is no way to establish a matrix size within a CASIO program.<sup>5</sup> For the few programs involving matrices used in *Inside Your Calculator* and in these web-based supplements, you can simply set up a matrix large enough to include any calculation you will be doing. The Casio calculators allow very large matrices with up to 255 rows and 255 columns.

The program **BIGMULT** of Appendix 6-1 calls for a matrix, **A**, with three rows and a number of columns depending on the size of the multiplicands. The program **BIGDIV** of this website calls for a matrix, **A**, with five rows and a number of columns depending on the divisor, dividend and quotient. To serve those programs we need only set up a matrix **A** with plenty of columns. Most will choose to do so with, say, 50 columns (enough to process 200 digit numbers since four digits are entered in each matrix element. It is possible, however, to use all 255 columns in order to process numbers with over 1000 digits, 4 digits at a time, or over 25000 digits, 10 at a time.

Here is how you can establish a matrix, **A**, with 6 rows and 50 columns. From the menu, choose **MAT**. That will take you to the Matrix screen where **Mat A** will be highlighted. If there is already a listing of rows and columns in the form:

**Mat A** : 6  $\times$  7

you will need to delete this. To do so use:

**F1** (**DEL**) or **F2** (**DEL A**) and then **F1** (**YES**).

Whether or not you had to do this, you are now ready to indicate the 6 rows and 50 columns. To do so, in the Matrix screen simply key:

**6 EXE 50 EXE**

This is all it takes to set up your 5 by 50 matrix **A**, which is filled with zeros.

---

<sup>5</sup> If any reader knows how to set up a matrix within a program, please inform the author and this section will immediately reflect that information.

## *Inside Your Calculator*

You are now ready to use matrix **A** in your program. But you should be careful when you do so to see that when you enter your program data that you fill the rest of the lines you will be using with zeros so that old data does not mess up your calculation.