Guide to the HP 9815A

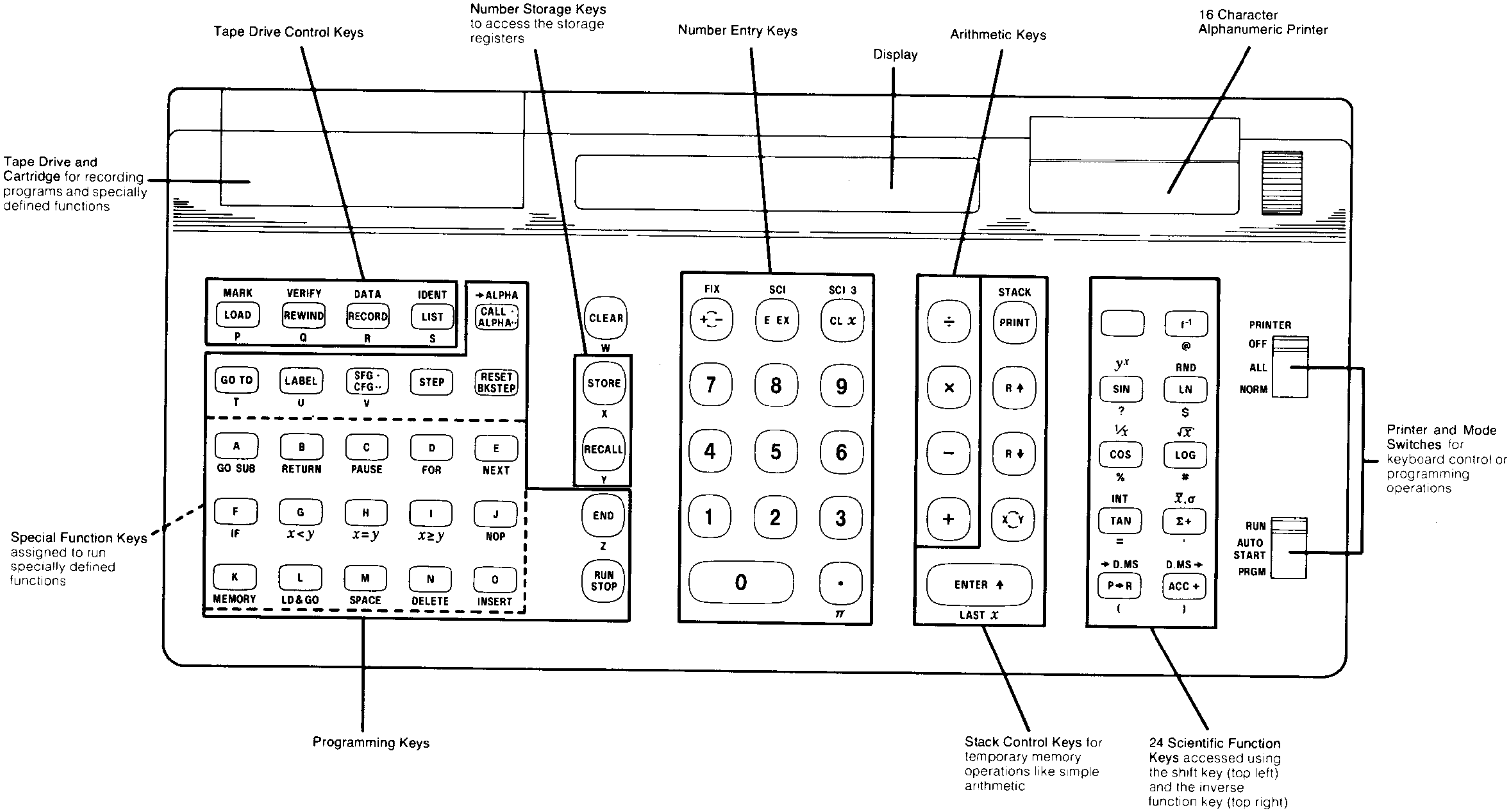
HP 9815A Calculator

HEWLETT–PACKARD CALCULATOR PRODUCTS DIVISION
P.O. Box 301, Loveland, Colorado 80537, Tel. (303) 667-5000

Table of Contents

Introduction . . . 1
Setting up the calculator . . . 3
Keying in numbers . . . 4
Looking at the stack . . . 5
Entering numbers . . . 7
Doing arithmetic . . . 8
Combining operations . . . 10
Changing formats . . . 12
Using storage registers . . . 15
Using math functions . . . 18
Looking inside a programmable calculator . . . 21
Labeling and defining special functions . . . 22
Writing a program . . . 25

THE HP 9815A CALCULATOR



Introduction

The purpose of this guide is to acquaint you with the main features of the HP 9815A without overwhelming you with the details of every operation. This way, you'll be able to do some easy arithmetic and math functions right from the start.

Keyboard operations are covered first, including arithmetic, number storage, a few basic math functions, printer operations and control of the display format.

Next you'll see what makes the HP 9815A a **programmable** calculator by taking a look inside at the memory and storage register areas, where programs and data can be stored.

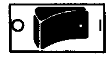
Then a few easy examples are used to illustrate the basic steps in program writing - how to define the problem, how to organize it into steps to find its solution, how to draw a diagram or flowchart that outlines the steps to be performed and how to key these steps into the calculator to create a program to solve the problem. Then you'll be able to write your own programs and specially defined functions.

If, after using this guide, you want to know more about the HP 9815A, refer to the Operating Manual.

Setting up the calculator

Before you begin...

If your calculator has never been plugged in, switched on, loaded with paper or diagnostically tested, then you must refer to Appendix 1 in the Operating Manual. Otherwise the calculator could be severely damaged.



Now let's start by turning the calculator on. Press 1 on the (Off / On) switch and the display lights up with -

If it doesn't, see Appendix 2 in the Operating Manual.

Next look at the other two switches at the right of the calculator. They control the calculator's mode and its built-in printer.

For now, set the PRINTER switch to OFF, so that nothing, except error messages, is printed.* In the ALL position, every number and operation that is keyed in is automatically printed. The NORM (normal) position is used for writing and running programs, but that's covered later.

Then set the mode switch to RUN for the keyboard operations that follow. The AUTO START mode is used for automatic loading and running of HP prerecorded tape cartridges. The PRGM (program) mode is set when you're keying in programs, but that's covered later, too.

*Various error messages are printed whenever key sequences are wrong, undefined (lettered) keys are pressed or other illogical operations are performed. The message is a brief description of the problem. A list of error messages is found at the back of this guide.

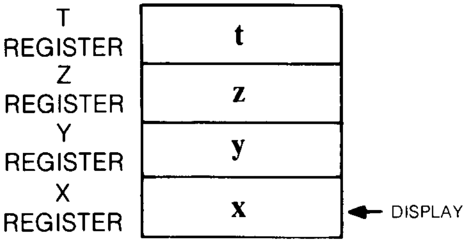
Keying in numbers

Let's try keying in whole and decimal (or fractional) numbers. First press the (CLEAR) key to remove any numbers that may have been keyed in previously.

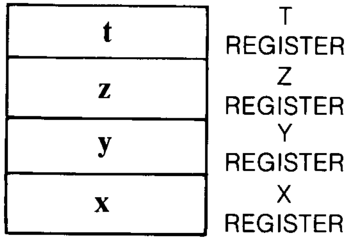
To key in a whole number, like 5, press (5) and the display shows -

Then press (·) (5) (5) to display -

The number is displayed as it is keyed in and it is also placed in a temporary storage register called the X register. There are actually four temporary storage registers as shown below, but for most of the arithmetic operations in this guide, we'll be using only the bottom two registers called X and Y. The X register always contains what's in the display and the Y register saves the other number in any arithmetic operation until the answer is displayed.



Looking at the stack



The Stack

These four registers are called the operational stack. To avoid confusion, register names are designated by capital letters and their contents by small letters. The four working registers of the stack temporarily save numbers to be used in arithmetic calculations. The X register is always displayed and can be printed by pressing (PRINT).

You can get a listing of the stack contents by pressing -

■* (PRINT) [STACK] to print this -

```
0.00
0.00
0.00
5.55
```

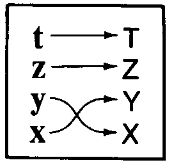
The 5.55 you keyed in is saved in the X register of the stack.

Then to clear all numbers from the stack and get a new stack listing, press (CLEAR) ■ (PRINT) [STACK] -

```
0.00
0.00
0.00
0.00
```

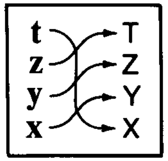
You can move numbers around in the stack by pressing these keys -

(x⇄y) swaps the contents of the X and Y registers, displaying the contents of the Y register

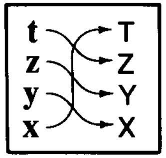


*Whenever the ■ (shift) key is pressed before another key, that key's alternate function is accessed. The shifted function of a key is usually printed above the key in the same color as the ■ key. In this case, the PRINT STACK function is performed.

[R↑] rolls all the numbers up in the stack and the number in the T register is moved to the X register -



[R↓] rolls all the numbers down in the stack and the number in the X register is moved to the T register -

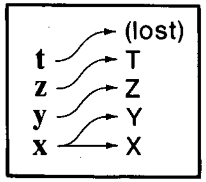


Entering numbers

Now let's **enter** the same number just keyed in by pressing -

[CLEAR]* [5] [·] [5] [5] [ENTER ↑]

The display remains the same because 5.55 is again in the X register, but the number is now **saved** - it is stored in the Y register, too, as shown below -



Press ■ [STACK PRINT] to see how the stack looks now.

```
 0.00
 0.00
 5.55
 5.55
```

To enter a negative number, press [5] [·] [5] [5] [ENTER ↑] [FIX +/-]

and −5.55 is displayed and stored in the X register.

The [ENTER ↑] key is used for saving numbers in the stack registers so that arithmetic operations can be performed. It automatically pushes up all numbers in the stack by one register. We can see that happen by pressing -

[ENTER ↑] [ENTER ↑] ■ [STACK PRINT]

```
 5.55
-5.55
-5.55
-5.55
```

*The [CLEAR] key doesn't have to be pressed each time you do a calculation, but it's used here to be sure the example works right.

Doing arithmetic

The four arithmetic operations - addition, subtraction, multiplication and division - use the numbers saved in the X and Y registers of the stack and an operator key -

(+) for addition

(-) for subtraction

(×) for multiplication

(÷) for division

While these operations are being performed, the calculator lets you know that it's "busy" by displaying -

- - - - - - - - - - - - - - - -

Now let's do a few examples. Key in this sequence for addition and watch what happens to the X and Y registers of the stack. First clear the stack by pressing (CLEAR) .

Key	Register	Contents
	Y	0.00
6.81	X	6.81
	Y	6.81
(ENTER ↑)	X	6.81
	Y	6.81
4	X	4
	Y	0.00
(+)	X	10.81

A close look at the stack shows that (ENTER ↑) causes the number that was keyed in (6.81) to be pushed up or duplicated in the Y register. This allows the 4, when keyed in, to replace it in the X register. Then the addition key ((+)) adds x to y. The result is displayed and saved in the X register.

Subtraction works in the same way. Try this -

9.99 (ENTER ↑) 1.11 (-) → 8.88

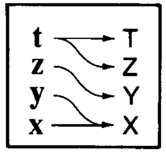
Now let's multiply two numbers. Key in this sequence -

4.11 (ENTER ↑) 5 (×) → 20.55

Here's a division example to try. Press -

80 (ENTER ↑) 20 (÷) → 4.00

In each case, the number in the X register is added to, subtracted from, multiplied by or divided into the number in the Y register. Then the result is displayed and saved in the X register.



The X register always saves the last result so that you can perform a series of operations without having to remember and re-key in the result.

Combining operations

Now that you've tried some basic arithmetic, you're ready to do a few problems that require more than just a single operation.

Let's suppose you have to carpet an entire building - three floors and a basement. The building is 140 feet long and 85 feet wide with a basement exactly half that size. At $5.04 a square yard, what would that cost?

First calculate the area of one floor.

140 [ENTER ↑] 85 [×]   11,900.00

Then multiply that result by the number of floors in the building to find the total floor space, in square feet, to be carpeted.

3.5 [×]   41,650.00

Then divide by nine to change square feet to square yards.

9 [÷]   4,627.78

And finally multiply the total yardage by the cost per square yard to find the total cost.

5.04 [×]   23,324.00

It would cost $23,324 to carpet that building!

Here's another easy problem. If you kept $100 in a savings account that earned 6% interest compounded monthly, how much money would you have in the account after a year? Here's the formula -

$$\text{amount} = \text{principal}\ (1+\text{rate})^{\text{interest periods}}$$

To solve this one, first put the monthly interest rate (.06 ÷ 12 = .005), plus 1, into all four registers of the stack.

1.005 [ENTER ↑] [ENTER ↑] [ENTER ↑]

Then multiply $100 by that, once for each month of interest earned. Key in 100 and press -

[×] twelve times

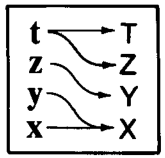
And your $100, after a year, becomes

106.17

In the first example, the result of each operation was saved by the **automatic enter** feature. Each time a new number is keyed in directly after an operator key, the result of the operation is automatically entered or duplicated in the Y register. To see this happen, press [STACK PRINT] between each operation. (Automatic enter occurs after every operator except [ENTER ↑] [CLEAR] and [CL x].)

In the second example, three registers of the stack contain the interest rate so that as many operations as needed can be performed without having to re-key in the result or the interest.

Following each multiplication operation, the result in the X register is displayed and a new copy of the interest rate is moved into the Y register. Since a new copy of the interest rate is duplicated each time the stack moves down, you never have to reenter it.



Changing formats

When the HP 9815A is switched on, the display has a two decimal place format. If you want to display more (or less) than two decimal places, you can change the format by pressing -

FIX and the number of places to be displayed, up to nine.

The number format affects displayed and printed numbers only; it does not affect calculating accuracy.

Let's see how that works. First key in 0.123456789 and press the ENTER ↑ key -

0. 12

Then to display three decimal places, press -

FIX 3	0. 123
or five FIX 5	0. 12346
or none FIX Ø	0

These key sequences fix or set the number of decimal places displayed. Notice that the number in the display is now Ø. That's because the calculator has automatically rounded it. Automatic rounding occurs when the number of decimal places entered exceeds the number of decimal places you've set to be displayed. When this happens the first excess digit is checked, and if it is -

5 or more - then the last digit to be displayed is **increased** by 1.

But if the first excess digit is 4 or less, then the last digit to be displayed is not changed. In either case, all excess digits are dropped.

In the example above, when only five decimal places of 0.123456789 are displayed, the 5 becomes 6 (because the first excess digit that follows is five or more) and all excess digits are dropped.

The key sequences that set the number of decimal places in the display affect the printer, too. The number of places that are displayed are also printed whenever the PRINT key is used. You can print out all ten formats of the number already entered by keying in these sequences. (Use the thumb wheel next to the printer to manually advance the printer paper.)

FIX 9 PRINT	0.123456789
FIX 8 PRINT	0.12345679
FIX 7 PRINT	0.1234568
FIX 6 PRINT	0.123457
FIX 5 PRINT	0.12346
FIX 4 PRINT	0.1235
FIX 3 PRINT	0.123 *
FIX 2 PRINT	0.12
FIX 1 PRINT	0.1
FIX Ø PRINT	0

When the calculator is out of paper, it lets you know by printing the error message OUT OF PAPER.

Of course the number of decimal places displayed has no effect on the internal workings of the calculator. All calculations are done to twelve significant places; accuracy depends on the operation performed.

*Notice that the 3 is not rounded because the 5 in the number above it is actually a 4 to the internal workings of the calculator.

If the numbers you're working with get very large or very small, the calculator will automatically convert its display to a format that handles more than nine decimal places. This format is called scientific notation.

Here's an example. First key in 9,000,000,000,000 and press (ENTER ↑) and the number is automatically displayed in scientific notation -

9 12

All that means is: move the decimal point in the whole number (which is 9) to the right the number of places indicated by the exponent (which is 12). So "9 12" is really just another way of writing 9,000,000,000,000 or 9×10^{12}.

If the exponent in the display is negative, the decimal point is moved to the left to convert back to fixed format. This is for very small numbers, like .000 000 000 000 8. Key in this number and the display is -

8. - 13

So "8 −13" is the same as .000 000 000 000 8 or 8×10^{-13}.

Using storage registers

Some calculations become much easier when there are a few permanent places to store numbers. Costs, percentages, rates or any numbers that are used repeatedly can be saved in any of the 20 storage registers of the calculator. And if 20 isn't enough, you can easily set up more registers when needed. If that's too many, some of the registers can be eliminated.

These storage registers have letter and number names, like register A and register 5. The ten lettered registers are named A through J; the numbered registers are 0 through 9.

Let's try using a few of these registers in the following example. We'll need three to store costs and sizes of precut glass available through the local glass dealer. His three best sellers are -

1' × 2' frosted glass (1/8" thick) at $3.50 per pane
2' × 4' window glass (1/16" thick) at $5.08 per pane
5' × 6'8" thermal glass (1/2" thick) at $178.85 per set of doors

We'll store these prices in registers A, B and C, once the format is set by pressing (■) (FIX) 2.

3.50 (STORE) (A)

5.08 (STORE) (B)

178.85 (STORE) (C)

If you're using more than one or two of these storage registers, the storage map on the following page is handy for keeping track of the registers used and their contents.

HP 9815A STORAGE MAP

REG #	CONTENTS	REG #	CONTENTS
A	1 x 2 Frosted 1/8" 3.50	F	
B	2 x 4 Window 1/16" 5.08	G	
C	5 x 6' 8" Thermal 1/4" 178.85	H	
D		I	
E		J	

REG #	CONTENTS	REG #	CONTENTS	REG #	CONTENTS	REG #	CONTENTS
0		0		0		0	
1		1		1		1	
2		2		2		2	
3		3		3		3	
4		4		4		4	
5		5		5		5	
6		6		6		6	
7		7		7		7	
8		8		8		8	
9		9		9		9	
0		0		0		0	
1		1		1		1	
2		2		2		2	
3		3		3		3	
4		4		4		4	
5		5		5		5	
6		6		6		6	
7		7		7		7	
8		8		8		8	
9		9		9		9	
0		0		0		0	
1		1		1		1	
2		2		2		2	
3		3		3		3	
4		4		4		4	
5		5		5		5	
6		6		6		6	
7		7		7		7	
8		8		8		8	
9		9		9		9	
0		0		0		0	
1		1		1		1	
2		2		2		2	
3		3		3		3	
4		4		4		4	
5		5		5		5	
6		6		6		6	
7		7		7		7	
8		8		8		8	
9		9		9		9	
0		0		0		0	
1		1		1		1	
2		2		2		2	
3		3		3		3	
4		4		4		4	
5		5		5		5	
6		6		6		6	
7		7		7		7	
8		8		8		8	
9		9		9		9	

With this information stored in the calculator, the dealer can quickly and accurately quote prices. Let's suppose a customer needs 2 panes of frosted glass, 8 panes of window glass and 1 set of thermal glass sliding doors. Here's how he'd get that quote -

Keys	Display
RECALL A 2 ×	7.00
RECALL B 8 × +	47.64
RECALL C +	226.49

And the quote is displayed. Of course, this is a very simple illustration of arithmetic operations that use the contents of storage registers. Imagine how useful it would be with a large inventory of items.

Numbers can be stored and recalled whenever necessary and the storage registers can be used or cleared out as they're needed. The contents of all storage registers are removed or deleted when the calculator is turned off. You can also clear registers A through J by pressing STORE CLEAR. To clear an individual register, store zero in it.

Using math functions

Let's take a look at those keys at the far right associated with the math functions. A function is a set of operations that together solve a problem, like squaring a number or calculating a square root or finding the sine of an angle. These and 21 other functions are built right into the HP 9815A and are very easy to use.

Let's try squaring a number. First set the display - ■ [÷ (FIX)] ∅.To square 11,111, press -

11111 [ENTER ↑] **2** ■ [SIN (y^x)] — display: 123,454,321

A number can be raised to any power (or multipled by itself as many times as indicated by the exponent in X) up to the limits* of the calculator, of course, by entering the base number in the Y register and the exponent in the X register and pressing ■ [SIN (y^x)].

Now here's a look at some of the other math functions. You can find the square root of a number by keying in the number followed by ■ [LOG ($\sqrt{x}$)].

If the previous result is still in the display, just press ■ [LOG ($\sqrt{x}$)] to get the square root of 123,454,321.

Display: 11,111

And the reciprocal of a number (or 1 divided by that number) is found using the ■ and [COS ($1/x$)] keys. (Change the format to ■ [÷ (FIX)] **9**.)

Find the reciprocal of 66 by pressing **66** ■ [COS ($1/x$)] -

Display: 0.0151515151 5

Another key sequence displays the value of π. Press ■ [• (π)] -

Display: 3.141592654

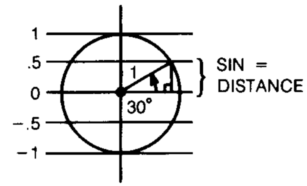
*The calculator can handle numbers between $-9.99999999999 \times 10^{99}$ and $+9.99999999999 \times 10^{99}$. If the number entered or calculated exceeds this, then the error message OVERFLOW is displayed.

Additional math functions built into the calculator are listed below.

- To round a number, enter the number in the Y register and the number of places you're rounding to in the X register. Then press ■ [LN (RND)]. (Use negative numbers to round decimal numbers.)
- To find the integer (or the whole number part) of a value, key in the number followed by the ■ and [TAN (INT)] keys.

The circular or trigonometric functions are also built right into the HP 9815A, and can be calculated with the angles expressed in any one of three units - decimal degrees, metric grads or radians. Besides sine, cosine, and tangent, the reciprocals of these (cosecant, secant and cotangent) and their inverse functions are available using the [COS ($1/x$)] (reciprocal) and the [f^{-1}]* (inverse function) keys.

Detailed descriptions of these functions are found in the Operating Manual, so for now, let's just find the sin of a 30° angle (in degrees). First press ■ [÷ (FIX)] **2** and then **30** [SIN] -



Display: 0.50

The sin of 30° is .5 as illustrated by the diagram and the display.

Log functions are also available. The common log (base 10) or the natural log (base e) of a number is displayed when the number is keyed in and the appropriate log key is pressed -

[LOG] for common logs
or
[LN] for natural logs

To display the constant e, first set the format (■ [÷ (FIX)] **9**) and then press -

[1] [f^{-1}] [LN] — display: 2.718281828

*When the [f^{-1}] key is pressed before a math function key, the inverse function of that key is accessed.

Antilogs, both common and natural, are also available using the [f^{-1}] key before the [LOG] or [LN] key.

Other math functions include -

- Statistical functions - summation, mean and standard deviation
- Complex Arithmetic - accumulate + and accumulate - for two sets of values
- Conversions - decimal degree to degree-minute-second conversions and back; rectangular to polar conversions and back.

All functions are fully described in the Operating Manual.

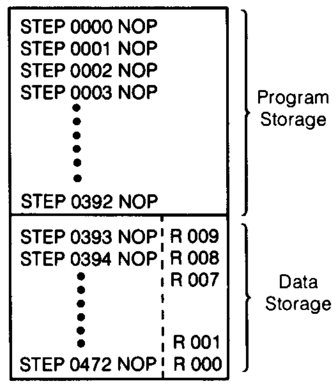
Looking inside a programmable calculator

Up to now we've been concerned with the operations and functions, like arithmetic and math, that are run directly from the keyboard by pressing a key or a sequence of keys. We've also seen the calculator's visible features like the printer, display, tape drive and cartridge. But what's on the inside that makes the HP 9815A a **programmable** calculator?

To understand what makes a calculator programmable, you must first understand what a program is. Very simply, a program is a set of instructions or steps that together perform the operations necessary to solve a math problem. To do this, the calculator needs a **language** so you can communicate with (or instruct) it and a **memory** so it can remember your instructions.

The language of the HP 9815A is found right on the keyboard. Nearly all of the keys are programmable; that is, they can be used to perform an operation within a program. The keys at the far left show the main **words** in the calculator's language. The meanings of some, like GO TO, STEP, PAUSE, IF and END may seem obvious; others like NOP, SFG, CFG and GO SUB, not so obvious, but all of the words you'll need to write an easy program will become very familiar in the next few pages!

Besides a language, a programmable calculator has a memory. The number storage area and a larger area for storing program steps make up the calculator's memory. A single step can contain one or more **words** of programming language. In the basic calculator, there's room for 392 steps which is quite adequate for most programs and 20 storage registers for data.* With Option 001, 2008 steps are available.



The Memory

Notice that the memory contains NOP (No OPeration) instructions. These remain in the memory until they are replaced by other program steps. If there is not enough program memory available for a program step or storage register assignment, the printer will indicate this by printing the error message, MEMORY OVERFLOW, and the program step or register assignment is ignored.

*When the calculator is turned on, 20 storage registers are available. If necessary, the 10 numeric-named registers can be converted to 80 extra programming steps.

Labelling and defining special functions

Once a program is written and keyed into the calculator to be stored as program steps, you've created a time saving, convenient routine that performs any function you want to define, like balancing your checkbook or calculating your overtime pay. Since it's stored in memory (until the calculator is turned off or the memory is erased*), it can be used repeatedly with different "**givens**," like a month's checks or a week's overtime.

To illustrate this, we'll use an example that converts Fahrenheit temperatures to Celsius using a **specially defined function**. The temperature conversion is based on this formula -

$$°\text{Celsius} = °\text{Fahrenheit} - 32 \times 5/9$$

The formula is easily changed to a keyboard sequence -

[ENTER ↑] 32 [−] 5 [×] 9 [÷]

Each time you want to convert a temperature from Fahrenheit to Celsius, key in the Fahrenheit temperature followed by the key sequence above.

Let's test that keyboard sequence with a 212° F temperature, because we know 212° F = 100° C. The result is displayed in degrees Celsius.

(First set the format by pressing [■] [FIX] 2.)

100.00

To write a specially defined function, that key sequence is converted to program instructions which are stored in memory as steps in the conversion program.

*Programs can also be recorded on tape cartridges to save keying them in at a later date.

For convenience, this program is associated with a lettered key* called its label. Once it's in memory you can do any temperature conversion by keying in the temperature to be converted, followed by the appropriate label key. The advantage to this is that **many** specially defined functions can be stored or stacked in the program memory as separate programs each named by its own label. A blank key overlay can be placed over these lettered keys so you can write the name of the program below the appropriate key.

Now let's key in the program. First press [END] and set the mode switch to PRGM to store the key sequences in the memory. Then set the PRINTER switch to NORM so the program steps are printed out as you key in these program instructions -

[LABEL] [C] (Label C for Conversion)

(It's not necessary to include the [ENTER ↑] key because when the program is run, an automatic enter occurs placing the Fahrenheit temperature to be converted in the Y register.)

32 [−]
5 [×]
9 [÷]
(performs the conversion calculation and displays the result)

[END] (ends the program)

The function is stored in memory as steps Ø through 9 and is associated with label C.

*There are 15 lettered keys that can be assigned to specially defined functions - LABELS A through O.

To run it, switch back to the RUN mode and key in the Fahrenheit temperature (use 212° again to test the program). Then press [C] and the equivalent Celsius temperature is displayed. Try a few more conversions:

41° F = 5° C

50° F = 10° C

−40° F = −40° C

This specially defined function converts any Fahrenheit temperature to Celsius without having to re-key the whole conversion sequence. Specially defined functions become even more valuable as the functions you're defining become more complex.

To get a listing of the program, press (END) [LIST] while in the RUN mode.

```
0000 LBL
---- C
0002 3
0003 2
0004 -
0005 5
0006 *
0007 9
0008 ÷
0009 END
```

Now that you've been through the mechanics of storing a program in memory and you've seen how it runs, you're ready for the basics of program writing.

Writing a program

There are four basic steps in writing a program -

1) Define the problem.
2) Find the best solution to the problem by performing the operations that solve the problem directly from the keyboard.
3) Organize these operations into logical steps by drawing a picture outline of the problem, called a flowchart.
4) "Translate" the steps of the flowchart into program instructions to **automatically** perform the operations required to solve the problem.

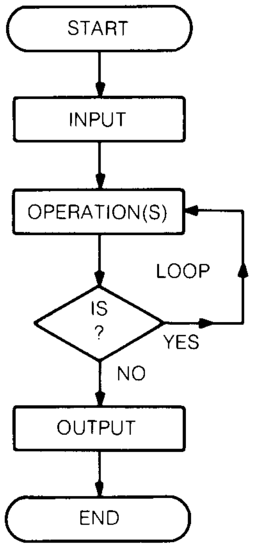
Let's take a closer look at each of these steps.

First you must determine exactly what the problem is and its overall structure. If you perform a series of related operations many times with different inputs, or if you use the result of one operation in the next operation, your program may contain a number of subroutines, or supporting routines, within the main program to calculate the final result. If the solution to your problem is found using a single operation, you can write a simple main program or you may want to write a specially defined function as shown in the previous section. Whatever the problem, you should first specifically define it and then determine its overall structure.

Next you should decide which operations solve the problem and in what order they should occur. To do this, use a typical value (or values) as input and perform the operations to solve the problem **directly from the keyboard.** By using a value whose result you already know (like 212° F in the Celsius conversion program), you can test your key sequence to see if it works. This technique gives you the basic math for the program you're writing.

The third step is to draw a flowchart of the problem so you can visualize the method for finding its solution. A flowchart is a series of boxes representing the operations necessary to solve the problem. The boxes are connected by arrows to show the order in which the operations are performed. Here's a general flowchart that could fit any program -

The beginning and end of the flowchart are indicated by ovals. All operations, including input and output, are enclosed in rectangles. The diamond-shaped block indicates a decision point, where either of two alternatives may be chosen. Usually a question requiring a yes or no answer is found in the decision block.



If the answer to the question is yes, then a loop is made back to a previous step and then to the question again. This loop is repeated until the answer to the question is no and the loop is exited. Then the final result is output.

The last step in program writing is to **translate** the operations in the flowchart to program instructions which can be stored in the calculator's memory. A Program Form like the one shown on the following page, is used to list the program instructions which make up the program.

To illustrate these four steps, let's try writing a program that performs a routine which we'll assume you must repeat many times each day - calculating the average of different groups of numbers.

Starting with the first step, we'll define the problem - we want a routine that averages a series of numbers. The structure is that of a basic program-an operation to add numbers, one to count them and another to divide the sum by the count, or number of inputs. We'll include those three math operations in the program.

HP 9815A PROGRAM FORM

PROGRAM TITLE ______ PAGE ____ OF ____ PAGES

MEMORY SIZE ☐ Standard 472 steps ☐ Expanded 2008 steps DATE ______ PROGRAMMER ______

TOTAL REGISTERS SET ☐

PERIPHERALS ☐ HP 9862A Plotter (Uses 72 steps.) ☐ HP 9871A Printer (Uses 40 steps.) ☐ Other ______ (Uses 8 steps.) ☐ Other ______ (Uses 8 steps.)

STEP #	INSTRUCTION	REMARKS	STEP #	INSTRUCTION	REMARKS
0			50		
1			51		
2			52		
3			53		
4			54		
5			55		
6			56		
7			57		
8			58		
9			59		
10			60		
11			61		
12			62		
13			63		
14			64		
15			65		
16			66		
17			67		
18			68		
19			69		
20			70		
21			71		
22			72		
23			73		
24			74		
25			75		
26			76		
27			77		
28			78		
29			79		
30			80		
31			81		
32			82		
33			83		
34			84		
35			85		
36			86		
37			87		
38			88		
39			89		
40			90		
41			91		
42			92		
43			93		
44			94		
45			95		
46			96		
47			97		
48			98		
49			99		

NOTE: The following instructions require two program steps. All others require only one step.

- All storage instructions involving numbered registers or indirect storage.
- All branch instructions, except computed branches.
- All display formatting instructions.
- Any instruction beginning with **CALL ALPHA**.
- All labels.

HEWLETT·PACKARD HEWLETT·PACKARD HEWLETT·PACKARD HEWLETT·PACKARD

The next step is to do an example from the keyboard. Let's calculate the average age of the ten richest people in the U.S. According to a magazine article* in 1973, their ages are 35, 43, 54, 63, 49, 71, 43, 60, 58 and 56, in order of their wealth. First set the format - ■ FIX ∅

From the keyboard, this sequence calculates the average -

35 ENTER ↑

Then key in the other nine ages each followed by + .

And press **10** ÷

The average age, 53, is displayed.

53

The next step is the flowchart. First we'll jot down the general steps needed to calculate the average of any series of numbers.

1) Input a number.
2) Add it to storage register (A)
3) Count it
4) Any more Inputs ?
   - if yes, Go back To step 1
   - if No, go to the next step
5) Divide the sum in register A by the count
6) Output (print or display) result

Two of these steps aren't complete. To calculate an average we need a **counter** to keep track of the number of inputs. We'll use storage register C as the counter for this program. Each time an input is made, 1 is added to register C. When step 5 is reached, the amount in register A is divided by the count in register C to get the average. So step 3 is changed to -

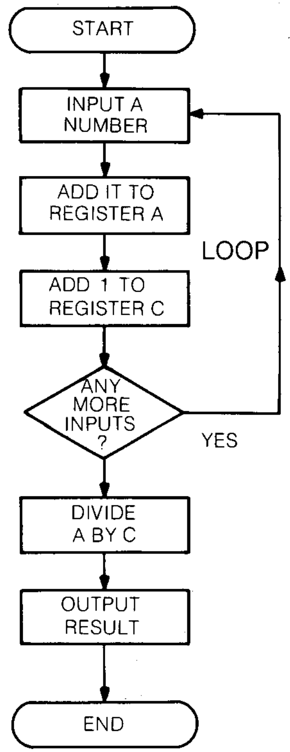
3) Add 1 to register C

and step 5 is changed to -

5) Divide the sum in register A by the count in register C

Now we'll draw the flowchart.

*M. Louis, "The New Rich of the Seventies", Fortune, Sept., 1973, P. 170 - 175+.



The flowchart outlines the operations that must be performed to calculate an average. The "loop" adds the inputs and counts them, storing the sum in register A and the count in C.

The decision point question causes either repetition of the loop if the answer to the question is yes, or continuation to the next step if the answer to the question is no.

When the answer is no, the final math calculation is performed - the sum in A is divided by the count in C and the answer is output.

Now the last step is to convert the flowchart into program instructions to calculate the average of a series of numbers. Here's where all the details, like clearing the stack and the storage registers and recalling the sum and the count to the X and Y registers, are written into the program.

First set the PRINTER switch to NORM and the mode switch to PRGM.

Clear the program memory by pressing K (MEMORY) N (DELETE) .

HP 9815A PROGRAM FORM

PROGRAM TITLE ______ PAGE ___ OF ___ PAGES

MEMORY SIZE ☐ Standard 472 steps ☐ Expanded 2008 steps DATE ______ PROGRAMMER ______

TOTAL REGISTERS SET ☐

PERIPHERALS ☐ HP 9862A Plotter (Uses 72 steps.) ☐ HP 9871A Printer (Uses 40 steps.) ☐ Other ______ (Uses 8 steps.) ☐ Other ______ (Uses 8 steps.)

STEP #	INSTRUCTION	REMARKS	STEP #	INSTRUCTION	REMARKS
0	Clear		50		
1	Store Clear		51		
2	Stop		52		
3	Store + A		53		
4	1		54		
5	Store + C		55		
6	Stop		56		
7	IF +		57		
8	Go to 3.		58		
9	—		59		
10	Recall A		60		
11	Recall C		61		
12	÷		62		
13	Print		63		
14	End		64		
15			65		
16			66		
17			67		
18			68		
19			69		
20			70		

HEWLETT · PACKARD

We'll use the Program Form above to list the program instructions as they're converted from the flowchart. Key in each instruction as shown on the Program Form. A key by key explanation of what occurs when the program is run, follows.

The program begins by clearing the stack with the (CLEAR) key in step Ø. Then the lettered storage registers A through J are cleared using the (STORE) (CLEAR) keys in step 1.

Next the (RUN STOP) key halts the program for input. When the program is running, the number input is automatically added to the amount in register A as indicated by the instructions in step 3, (STORE) (+) (A).

The counter is found in steps 4 and 5 where (1) and (STORE) (+) (C) are keyed in. This increases the amount in register C by 1 each time a number is input.

In these last three steps, register arithmetic is performed. Two operations (arithmetic and number storage) are automatically combined; the addition is performed and the result is stored in the designated storage register.

The (RUN STOP) key in step 6 halts the program for all other inputs.

Next step 7 asks the decision question by checking if the last input is a positive number using the (F/IF) (+) keys.

If the answer is yes, then the next instruction is performed ((GO TO) (3)) and the program branches back to step 3 where the next input number is added to the amount in register A. If the answer to the question is no (the input is a negative number), then the next instruction is skipped and the instruction in step 10 is performed. This indicates that all inputs are made and the final calculation can be performed. (When keying in step 8, key in the step address as "3" followed by a decimal point to terminate it.)

Steps 10 and 11 recall the total from register A to the Y register ((RECALL) (A)) and the number of inputs from register C to the X register ((RECALL) (C)). Step 12 then divides the amount in Y by the amount in X, as instructed by the (÷) key.

The result is output as indicated by the (PRINT) key in step 13 which prints the result of the calculation found in the X register.

The last line of the program is an END instruction and that's the averaging program! Now let's see it work!

First list the program by switching to the RUN mode and pressing

(END) (LIST).

```
0000 CLEAR
0001 CLRA→J
0002 STOP
0003 STO+    A
0004 1
0005 STO+    C
0006 STOP
0007 IF +
0008 GOTO    0003
0010 RCL     A
0011 RCL     C
0012 ÷
0013 PRINT
0014 END
```

Now we'll run the program and key in the inputs from the keyboard. Still in the RUN mode, press [END] [RUN STOP] and key in the ages, pressing [RUN STOP] after each input to restart the program -

35, 43, 54, 63, 49, 71, 43, 60, 58, 56.

Once all inputs are made, key in a negative number to indicate this - (use Ø followed by [←]) and then press [RUN STOP] .

This technique is used to get out of the totaling loop and on to the final division operation. Then the result is displayed and printed - the average age of the ten richest people is 53.

This short program illustrates a number of basic programming techniques like data input, branching, looping, IF instructions for decision points and data output.

Additional program instructions found on the keyboard can be used for program editing and more complex programming techniques like printing alphanumerics, setting and clearing flags, using subroutines with GOSUBs and RETURNs and looping with FOR-NEXT instructions. These topics are covered fully in the Operating Manual.

So if you want to know more about your HP 9815A and its capabilities, see the Operating Manual.

See the Operating Manual for a list of Hewlett-Packard Sales and Service Offices.

ERROR MESSAGES

OVERFLOW	Number or result exceeds calculating range.
NO I/O DEVICE	Peripheral device or interface not connected.
ILLEGAL ADDRESS	Improper step address or storage register specified.
ILLEGAL ARGUMENT	Improper value for operation (e.g., ASIN, ACOS of no. >1 or <1; over 7 digits converted to D.MS format).
MEMORY OVERFLOW	Program instruction, storage register assignment or program loaded from tape exceeds available memory.
GOSUB OVERFLOW	More than seven subroutines (including special functions) nested at a time.
KEY NOT DEFINED	Special function just called is not defined.
IMPROPER SYNTAX	Incorrect use of [FIX ←] [SCI E EX] [SCI 3 CL x] [CALL ALPHA] [SFG CFG] [D FOR] [E NEXT] or [F IF].
CHECKSUM ERROR	Program or data loaded into calculator not identical to that in file; this usually indicates a dirty tape head or a worn tape.
SECURED MEMORY	Attempting to list, edit or record a secured program.
VERIFY FAILED	Program or data in file not identical to that in calculator.
WRONG FILE TYPE	Attempting to load an empty, extra or binary file; recording on an extra file.
END OF TAPE	End of tape reached during MARK operation. Also indicates a broken or defective tape; if the tape does not appear to be broken, (advance it using the drive wheel), replace the cartridge, press [REWIND] and continue.
PROTECTED TAPE	The cartridge RECORD slide is positioned to prevent MARK and RECORD operations.

Explanations of many of the items on these two pages are beyond the scope of this book. See the Operating Manual.