



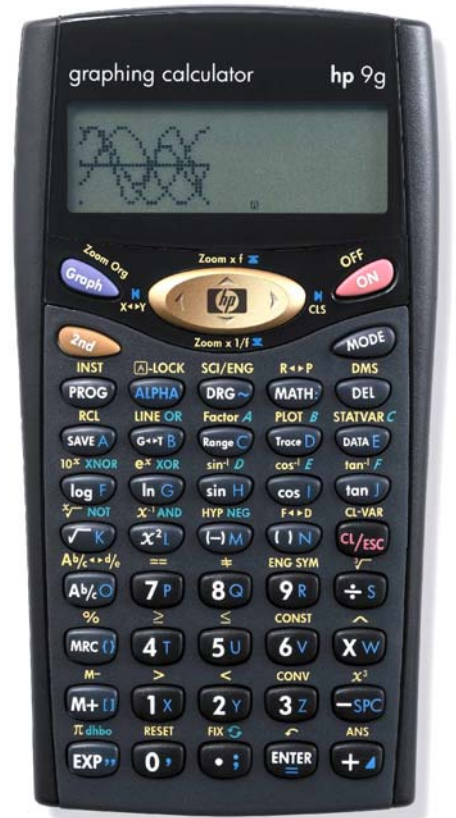
hp calculators

HP 9g Using Memories to Solve Problems

The Running Memory and the Standard Memory Variables

Array Variables and the Defm Command

Practice Using Memories



The running memory and the standard memory variables

In addition to the ANS memory (a special register in which results are stored every time the ENTER key is pressed to do a calculation– refer to the HP 9g learning module *Basic Arithmetic*), the HP 9g have two more types of memories: the running memory and the memory variables.

The running memory is similar to the one found on many calculators. It's controlled by the three keys: $\text{M}+\text{D}$, MRC and M^- . This memory is most useful to add a series of results (e.g. when balancing your checkbook). $\text{M}+\text{D}$ and M^- act as the ENTER key – the calculation in the entry line is performed and the result is displayed in the result line and stored in the ANS memory. But it is also added ($\text{M}+\text{D}$) or subtracted (M^-) to the running memory. Note that adding a negative amount using $\text{M}+\text{D}$ is the same as subtracting that amount with a positive sign using M^- . Pressing MRC ENTER recalls the contents of the running memory. Pressing it twice in a row (MRC MRC) clears the running memory (i.e. stores 0 in it). When the running memory contains a non-zero value, a small M appears in the display. The running memory can be used in calculations – just press MRC and its name (M) will be inserted where the cursor is.

The HP 9g also has 26 memory variables that can store numbers. They are named A, B, C, etc. through Z. Notice that variable M is actually the *same* memory as the running memory. Pressing RCL displays a menu of all the variables. Use the arrow keys to scroll through the three screens. Non zero-values are displayed in the result line for the currently selected variable (underlined). To recall the value (not the name) of a variable into the entry line, select a the desired variable and press ENTER . If you want to insert the variable *name* instead, just type its name in ALPHA mode.

To assign a value to a memory variable use the SAVE A key. The format is *value* SAVE A *variable_name*. Note that SAVE A automatically sets ALPHA mode, so you can type the variable name without pressing ALPHA first. The way to assign values to memory variables *in a program* is different; the format is now *variable_name* ALPHA ENTER *value*.

Array variables and the Defm command

Memory variables can be handled in other way. In addition to being named (A though Z) they can be referred to this way: A[0] through A[25]. It's important to note that this is another way of referring to the very same data. A[n] are called array variables and are very useful in programming, e.g. to apply a FOR loop to a set of variables. They are used in the same way as memory variables.

But 26 variables may not be enough in some occasions. In such cases, the HP 9g provides the option to convert program steps (there are 400 initially) into variables. The exchange rate is 12 program steps for one variable. Up to 33 variables can be created this way ($12 \times 33 = 396$) which are added to the 26 named variables that are always present. Since there are no more letters for the newly created variables, they have to be referred to as array variables. The command that performs this conversion is Defm: MATH \wedge 2Y . Pressing MATH \wedge 2Y ENTER without arguments displays the current setting, e.g. M-26 S-120 which means that there are 26 variables and 120 remaining program steps. Press MATH \wedge 2Y and the desired number of new variables and ENTER to create new variables. To restore the original configuration (the 26 variables) specify Defm 0.

The HP 9g learning module *Clearing, Editing and Correcting* explains how to clear variable memories. In brief, CL-VAR clears all the variables, and one variable can be cleared storing zero in it. Notice that $0 \rightarrow \text{SAVE A} \leftarrow \text{M} \text{ENTER}$ is an alternate way of clearing the running memory.

Practice using memories

Example 1: Calculate $1.46 + 12 - 7.5 + 67 + 987 + 6 - 89$ using the running memory.

Solution: Chain calculations involving additions and subtractions of numbers can be easily done with the running memory. First make sure the running memory contains zero by pressing MRC MRC ENTER . Now let's type in each number but instead of entering them with ENTER use the $\text{M}+\text{D}$ key:

1X . 4T 6V $\text{M}+\text{D}$ 1X 2Y $\text{M}+\text{D}$ M 7P . 5U $\text{M}+\text{D}$ 6V 7P $\text{M}+\text{D}$
 9R 8Q 7P $\text{M}+\text{D}$ 6V $\text{M}+\text{D}$ M 8Q 9R $\text{M}+\text{D}$

The total is displayed by MRC ENTER

Compare the above way with the usual way of entering the entire calculation and pressing ENTER at the end:

1X . 4T 6V + 1X 2Y -SPC 7P . 5U + 6V 7P + 9R 8Q 7P + 6V -SPC 8Q 9R ENTER

This second method is indeed shorter, but using the running memory has the advantage that you decide what results are added to the total and also lets you do intermediate calculations.

Answer: 976.96

Example 2: An engineer is working with a type of concrete that has a density of 149.8 lb/ft^3 . Store the density of this concrete in a variable and use it to calculate the mass of a concrete beam 2 ft by 1.5 ft by 20ft.

Solution: In many cases it is helpful to use variable names as mnemonics, for example D for density or P for pressure, but when registers are used to store a table, for instance, then names are meaningless and the numeral for each register is what counts. In this example, storing the density in the variable D makes good sense, to do it press:

1X 4T 9R . 8Q SAVE A Trace D ENTER

When SAVE A is pressed, the symbol "A" appears at the bottom of the screen. This tells the user that the next key pressed should be one of the keys with letters A to Z at their right, and that the corresponding letter will be used. For the letter "D", press the Trace D key. To calculate the mass, multiply the length by the width by the height. Then recall the number from D and multiply by that. Press the keys:

2Y XW 1X . 5U XW 2Y 0 ALPHA Trace D ENTER

The multiplication symbol is not needed if one of the parts (or both) is a variable name. In this example, notice the implicit multiplication between 20 and D.

Answer: The beam has a mass of 8988 pounds. This value of the variable D can be used for further calculations. If a different type of concrete is selected, the density of this new type can be stored in D and the calculations can be repeated. The value in D is not lost when the calculator is turned off.

HP 9g Using Memories to Solve Problems

Example 3: Store the value 25 into variables A through M.

Solution: Instead of assigning the value to each variable manually, there's a much faster way. If it's the same value that is assigned to a several consecutive variables, you can use the ~ key (ALPHA DRG~) to represent a range of variables. In this example, press

2Y 5U SAVE A SAVE A ALPHA DRG~ ALPHA (←)M ENTER

Answer: 25 is returned to the result line. Variables A through M now contain the value M. Try viewing them with 2nd RCL

Example 4: Create ten new variables and store the values 0 through 35 into variables A[0] through A[35] respectively.

Solution: Since the value to be assigned is not the same for all the variables, we can't use the previous method now. The fastest way is to write a small program. To create the new variables execute Defm 10 by pressing

MATH: ^ 2Y 1X 0, ENTER

If the message error "MEMORY Er" appears then you don't have enough free program steps to be converted into ten variables – you'll have to delete all the programs you don't need any more. The program is simply a FOR loop:

```
FOR ( A = 1;A ≤ 35 ; Z++ )
  { A [A] = A }
A=0;
```

Here are the keystrokes needed to enter the program once you are in Program Edit mode. (Refer to the HP 9g learning module *Writing a Small Program* for more information on creating and editing programs).

Line No.	Command	Explanation	Keys
Line 1	FOR (A = 1;A ≤ 35 ; Z++) {	The following instruction is done 35 times	2nd INST 3Z ALPHA SAVE A ALPHA ENTER 1X > ALPHA SAVE A 2nd ≤ 3Z 5U > ALPHA SAVE A 2nd INST >>> 2Y >> ENTER
Line 2	A [A] = A }	The value of the loop counter A is stored in A[A], i.e. 1→A[1]=B, 2→A[2]=C, ..., 35→A[35]	2nd ALPHA SAVE A M+[] SAVE A ALPHA > ALPHA ENTER ALPHA SAVE A > ENTER
Line 3	A=0;	Finally, 0 is stored in A, which has been used as a loop counter.	ALPHA SAVE A ALPHA ENTER 0,

Answer: Once this program is executed, variables A through Z will contain the values 0 through 25 respectively and variables A[26] though A[35] will contain the values 26 through 35 respectively.

Note: A[n] is not the only way of referring to array variables. Actually, you can use any letter in lieu of A, but with another meaning. For example B[-1] is the same as A[0], B[0] is the same as A[1] and so on; C[-2] is the same as A[0] and B[-1], C[-1] contains the same value as A[1] and B[0], etc.

HP 9g Using Memories to Solve Problems

A=0.773576436 in the entry line and the cursor is placed just after the equal sign. This is the value that was stored in variable A in the previous example. To enter the new value, just press 2Y 4T , CL/ESC to discard the previous numbers and finally ENTER . The same procedure applies to B and C:

2Y 8O CL/ESC ENTER
 3Z 4T CL/ESC ENTER

The result, 15.68, is displayed in the result line. Let's now solve the second triangles by running PROG 0 again: PROG 0 ENTER ENTER A is now 18 so press 1X 8O CL/ESC ENTER . Since B is still 28, just press ENTER . C has changed, though, press 4T 5U CL/ESC ENTER and the side c of this triangle appears in the result line. Finally, these are the keys need to solve the third exercise:

PROG 0 ENTER ENTER
 2Y 3Z CL/ESC ENTER
 1X 7P CL/ESC ENTER
 ENTER

Answer: Rounding to two decimal places, the side c of the three triangles are 15.68, 19.88 and 16.28 respectively.