



hp calculators

HP 35s Using Calculator Memories to Help Solve Problems

Variables and Memory Registers

Practice Examples:

Storing and Using a Constant

Storing a Temporary Result

Exchanging and Viewing Registers

Other Operations with Memory Registers



Variables and Memory Registers

When an equation is typed on the HP 35s, it can use **variables** with names from A through Z. For example an equation

$$3X^2 - 5X = A$$

has the variables X and A in it.

Variables can also be used in programs and in calculations from the keyboard.

Each variable consists of a number and of a place in the calculator memory where the number is stored.

The number is called the **value** of the variable. If no value has been give to a variable then its value is 0.

The place in memory where this number is stored is called a **data register**, or a **memory register**, or just a **memory**.

Each memory register can be referred to by a numeral as well as by its name. Register A is -1 and register Z is -26. Six more registers can be referred to by numerals and hold values from statistics calculations. Two of the lettered variables are special "index" registers, which are explained in another training aid. The names "data register" or "memory register" or "memory" refer to all of these, not just the variables, so these names are often used in this training aid, rather than the name "variables."



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 then names are meaningless and the numeral for each register is what counts.

This training aid shows ways in which memory registers can be used. A separate training aid covers the special topic of how arithmetic can be carried out directly in memory registers and using the memory registers.




Practice Example: Storing and Using a Constant

The HP 35s provides a set of physical constants, such as the speed of light. The conversion functions also provide constants to convert between metric and imperial measurements. Users who need to store other constants can put them into the memory registers so they can be easily used in calculations.

Example 1: An engineer is working with a type of concrete that has a density of 149.8 lb/ft³ (Different types of concrete have different densities, so the density of concrete is not a physical constant provided by the HP 35s!) Store the density of this concrete in a data register and use it to calculate the mass of a concrete beam 2 ft by 1.5 ft by 20ft.

Solution: Type the density, then press   and a variable name to store the density in that variable's memory register. To store a density of 149.8 in variable D, press these keys.

When   is pressed, the symbol "A..Z" appears at the top 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 lower right, and that the corresponding letter will be used. For the letter "D", press the  key.

HP 35s Using Calculator Memories to Help Solve Problems

To calculate the mass, multiply the length by the width by the height. Then recall the number from D and multiply by that. In RPN mode, press the keys:

2 **ENTER** **1** **.** **5** **X** **2** **0** **X** **RCL** **D** **X**

In algebraic mode, press:

2 **X** **1** **.** **5** **X** **2** **0** **X** **RCL** **D** **ENTER**

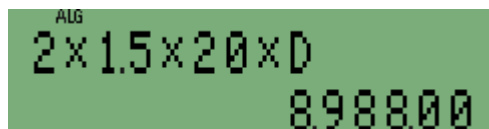


Figure 1

Answer: The beam has a mass of 8,988 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.

Note: The recall and the multiplication can be combined into one command **RCL** \times **D**. A separate training aid describes using arithmetic with the memory registers.

Practice Example: Storing a Temporary Result

The memory registers are available separately from the memory in which ordinary calculations are carried out. In algebraic mode, up to thirteen levels of brackets can be used, together with numbers saved with the brackets. RPN mode has four stack registers, X, Y, Z and T, and also the LastX register, often called L. The memory registers are separate from these. Note that the variables named L, T, X, Y and Z are not the same as the stack registers with these names, and that the Exponent key **E** and the Cancel key **C** do not access the variables with these names.

If a calculation is more complicated than the algebraic or RPN rules allow, temporary results can be stored in memory registers, and can then be used later.

Temporary results can also be stored in memory registers just to make a calculation easier, as in this example.

Example 2: The formula below uses the expression $(0.2 + \sin(35^\circ))$ three times. In algebraic mode, it would be difficult to re-use this expression without typing it in each time. In RPN mode the expression could be calculated once, then kept on the stack and re-used as needed, but this would require keeping track of which number is where on the stack.

$$\frac{5 \cdot \text{ASIN}(.2 + \text{SIN}(35))}{\text{ACOS}(.2 + \text{SIN}(35))} \cdot \sqrt{3 \cdot (.2 + \text{SIN}(35))}$$

Figure 2

Solution: The expression uses degrees, so set Degrees mode if it is not already set. (To do this press **MODE** **1**.) First calculate the expression that is used several times and store its value in a memory register. In this example, use register V for the value of the expression. Then calculate the whole formula, recalling the value each time it is needed.

In algebraic mode, type the expression like this:

$\boxed{0.2} \boxed{+} \boxed{\text{SIN}} \boxed{3} \boxed{5} \boxed{\text{ENTER}}$



0.2+SIN(35)
0.7736

Figure 3

Then store it in register V by typing:

$\boxed{\text{RCL}} \boxed{\text{STO}} \boxed{\text{V}} \boxed{\text{ENTER}}$



LAST x V
0.7736

Figure 3

Now the main formula can be calculated. First calculate the top line at the left-hand of the formula. The value of the expression is still available, so there is no need to recall it, but to see how it all works, recall V, calculate its arc sine and multiply by 5:

$\boxed{\text{RCL}} \boxed{\text{ASIN}} \boxed{\text{RCL}} \boxed{\text{V}} \boxed{>} \boxed{\times} \boxed{5}$

Next, divide by the arc cosine of the expression, recalling it again from V:

$\boxed{\div} \boxed{\text{RCL}} \boxed{\text{ACOS}} \boxed{\text{RCL}} \boxed{\text{V}}$



SIN(V) x 5 ÷ ACOS(V)

Figure 4

Now multiply by the square root of three times the expression.

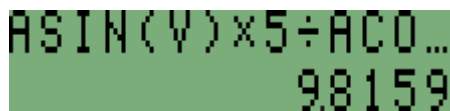
$\boxed{>} \boxed{\times} \boxed{\sqrt{x}} \boxed{3} \boxed{\times} \boxed{\text{RCL}} \boxed{\text{V}}$



5(SIN(V) x SQRT(3 x V)

Figure 5

Finish the calculation by pressing $\boxed{\text{ENTER}}$:



ASIN(V) x 5 ÷ ACOS(V)
9.8159

Figure 6

Answer:

To four decimal places the complete formula evaluates to 9.8159. Calculation was considerably easier, and the expression displayed on the upper line is simpler, because the temporary value was stored in a memory register.

In RPN mode the calculation would be carried out using the following keys:

Calculate the value and store it in V:

3 **5** **SIN** **·** **2** **+** **→** **STO** **V**

As the value is already on the stack, just calculate its arc sine, then multiply by 5:

→ **ASIN** **5** **×**

Next, divide by the arc cosine of the expression, which is recalled from V:

RCL **V** **→** **ACOS** **÷**

Now multiply by the square root of three times the expression. In RPN there is no need for brackets:

3 **RCL** **V** **×** **√x** **×**



0.2000
9.8159

Figure 7

Answer: In RPN mode the answer is the same, and carrying out the calculation in a suitable order reduces the number of keystrokes needed.

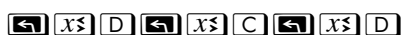
Practice Example: Exchanging and Viewing Registers

Two more commands that work with memory registers are “exchange” and “view”. Pressing **↔** **↔** displays the A..Z symbol, and any letter from A through Z can be entered. The value in the chosen register is exchanged with the value shown on the lower line of the calculator screen.

One use for this is to see what value is stored in a register. For example, the engineer from Example 1 might forget whether the density of concrete is in register C for Concrete or in register D for Density. Pressing **↔** **↔** **C** will bring the value from register C to the lower line of the screen so it can be seen, and pressing **↔** **↔** **C** again will put the values back as they were. The exchange command is rarely used for this, because the VIEW command, described below, does the job better. Another use for the exchange command is shown in the example below.

Example 3: In the middle of a calculation, the engineer from Example 1 decides that register D should be used to store the depth of the foundations of a building. Currently the Area, the height, and the depth of the building are in registers A, B and C. That means the density of concrete should be taken from D and placed in C, while the number in C must be placed in D, all without losing the value currently in use shown on the lower line of the display. How can this be done?

Solution: The engineer first exchanges D with the current value. The current value gets stored in D and the density is in the current value. Then the density, now in the current value, is exchanged with the depth, in register C. This puts the density of concrete into register C and the depth into the current value. Finally, exchanging the depth in the current value with the original value now in D completes the exchange. The keys pressed are:



Answer: The three exchanges swap values in two data registers without affecting an ongoing calculation. Note that this works in algebraic and in RPN mode.

Pressing displays the A..Z symbol, and any letter from A through Z can be pressed. The value in the chosen register is displayed in a message without changing the value on the lower line of the calculator screen. For example, pressing after Examples 1 and 3 should now display the density of concrete stored in register C.



Figure 8

Pressing the cancel key on the lower left of the keyboard, or the back-arrow at the middle right, clears the message and brings back the display in the calculator screen.

Other Operations with Memory Registers

There is no special command to clear a register, it is enough to store a zero in it, but see also the description below of CLVARS and of the register catalog.

Variables are used in equations and in programs. When the Solver is used, it asks which variable is the unknown one and prompts for values for the other variables. The Integrate command must be told which is the variable to integrate for.

The INPUT command allows a program to request a value for a variable.

Variables can also be used with the program loop control commands ISG (Increment and Skip if Greater) and DSE (Decrement and Skip if Equal).

The CLVARS command is activated by pressing and clears all the variables A through Z to zero.

Registers 28 through 33 are the statistics registers, normally used for statistical calculations. Numbers can not be stored directly into them through statistical operations, but can be placed in them through the index register. This will lead to errors if incorrect values are stored into these registers. The statistics registers are cleared when is pressed.

Pressing selects the Memory menu and then pressing displays a catalog of the registers. The and cursor keys step through the registers. To see all the digits of a register, press when the register is displayed. To copy a register to a calculation, press . To clear the variable to zero, press . To cancel the catalog, press the cancel key , or to go back to the Memory menu, press .

Arithmetic calculations using the registers directly are described in a separate training aid, as has been mentioned already.

As the above shows, the memory registers provide many ways to help in calculations on the HP 35s.