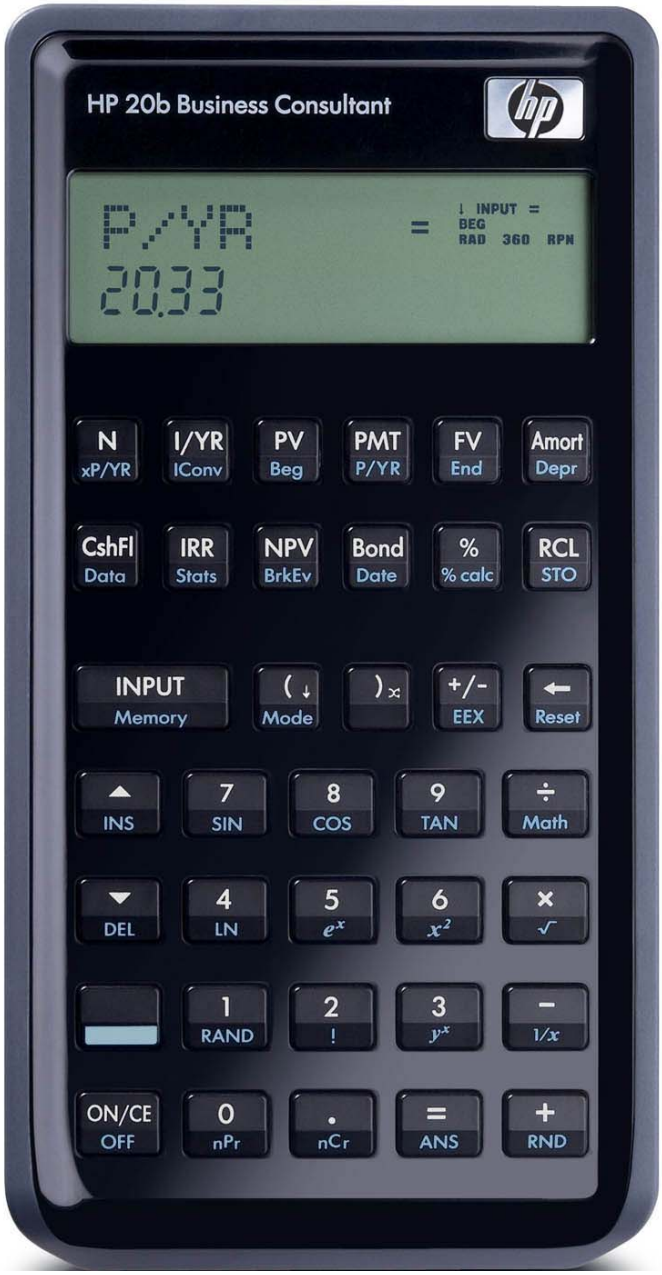
hp calculators

HP 20b
Using Memories to Solve Problems



Constant memory and planning

Memory registers

Other memory locations

Understanding register arithmetic

Viewing register contents

Clearing memories

Practice using registers to solve problems

Constant memory and planning

Because of the HP 20b Constant Memory feature, any current computations may be stopped at any moment for later resumption without any data loss. In the HP 20b, memory is organized as a set of **registers**, where each register can hold a number representation of any positive or negative value with a 12-digit mantissa and a three-digit integer, positive or negative exponent of ten.

Memory registers

There are 10 data storage registers in the HP 20b calculator. These registers can be directly addressed anytime it is necessary, provided that the register is available. Special care must be taken in regard to make sure that the chosen register isn't being used for other purposes. As a general rule, two basic operations can always be performed over existing register contents: storing and retrieving (or recalling) numbers.

Assuming that the display contains a value to be stored, to store that number in a register, press [↓] [STO] and then a number key from 0 to 9. To retrieve the information stored in one of the registers back to the display and use it in further calculations, press [RCL] and then a number key from 0 to 9.

It is also possible to store and recall values directly into the five Time Value of Money (TVM) registers, [N], [I/YR], [PV], [PMT], and [FV] by pressing [↓] [STO] and then one of these keys. Values in these registers are recalled for viewing or further use by pressing and then [RCL] one of these keys. If you are not planning to solve a TVM problem, then these five locations can be used as extra storage registers.

Other memory locations

The HP 20b also has memory space allocated to store statistical data and cash flow data. There is room in the HP 20b to store a total of 50 statistical data pairs or 50 cash flow values and frequencies, or any mixture of the two, provided there are 50 or fewer total pairs. If you were to store one statistics data pair, you would then have room for 49 distinct cash flows, for example. These memory locations are not covered in this learning module, as they are discussed elsewhere.

Example 1: The index for conversion between American Dollars and Euros is stipulated as 1.00 Euro = US$1.59. Which keystroke sequence stores this index in register R_1?

Solution: The complete sequence for typing the index in and storing its value in register R_1 is:

[1] [.] [5] [9] [↓] [STO] [1]

Understanding register arithmetic

Along with storing and retrieval, you can use storage register arithmetic and recall register arithmetic on the ten memory registers. The next example shows how this would work for storage register arithmetic.

Example 2: The index for conversion between American Dollars and Euros has its valued raised by 1%. What is the keystroke sequence that updates the index in register R_1?

Solution: When the keystrokes below are used, the index is updated in R_1.

[1] [.] [0] [1] [▬] [STO] [×] [1]

These keystrokes will multiply the value in the display, 1.01, by the value stored in memory register 1 and store the result in register 1. If it is necessary to check the resulting value stored in R_1 or use it in further calculations, the following keystroke sequence must be used to bring its contents to the display:

[RCL] [1]

Answer: Now the relationship is set to 1.00 Euro = US$1.61. The value displayed is the updated index value. Note that the actual value displayed will vary depending on your choice of the number of decimal places displayed. The full value is 1.6059.

The next example shows how recall register arithmetic works. In chain or algebraic mode, you must press [=] to finish the sequence. In RPN mode, this keystroke is not necessary.

Example 3: Store 123 in R_2 and the multiply the following numbers by the value stored in it: 3, 10, and 14. Use recall register arithmetic.

Solution: First, store 123 in memory 2.

[1] [.] [2] [3] [▬] [STO] [2]

Then, to multiply the first number 3 by the contents of memory 2, press

[3] [RCL] [×] [2] [=] (In RPN mode, there is no need to press [=])

Then, to multiply the second number 10 by the contents of memory 2, press

[1] [0] [RCL] [×] [2] [=] (In RPN mode, there is no need to press [=])

Then, to multiply the third number 14 by the contents of memory 2, press

[1] [4] [RCL] [×] [2] [=] (In RPN mode, there is no need to press [=])

Answer: The values displayed would be 369, 1230 and 1772.

For store or recall register arithmetic, any of the four arithmetic operators can be used: [+], [−], [×], or [÷].

Viewing register contents

To view the contents of a memory register, you can, of course, recall it to the display if you wish. However, this may disrupt the stack or an operation in progress. Another way to view the data registers is to press [▬] [Memory]. This key sequence will display the contents of memory 1 as shown below.

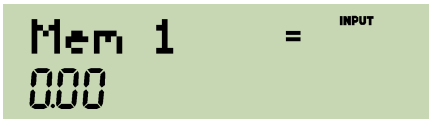


Figure 1

This screen indicates that memory 1 has the value of 0.00 presently stored in it. You can press ▼ or ▲ to scroll through the ten memory registers. Note as well that INPUT is active in the top right of the display. This indicates that if you key in a value and press INPUT, that value will be stored into the memory register displayed, replacing any value previously stored there.

If you press ▼ or ▲ enough times, you will eventually see, in addition to the contents of memories 0 through 9, menu options reporting how many of the available 50 slots for cash flow and statistical data values remain, as well as the number of statistical data values presently stored and the number of cash flow values stored.

Clearing memories

To clear all data memories at once, press ▬ Reset. and then ▼ or ▲ until you see the display shown below in figure 2. At this display, pressing INPUT will reset all ten data memories to a value of 0. The message "Memory Reset" will be scrolled across the screen.

Figure 2

Practice using registers to solve problems

Example 4: The sales price of the last 10 homes sold in the Parkdale community were: $198,000; $185,000; $205,200; $225,300; $206,700; $201,850; $200,000; $189,000; $192,100; $200,400. What is the average of these sales prices?

Solution: Register R_0 will be used to perform all operations. Then the resulting value will be brought back to the display. The average is computed using the following expression,

$$\bar{x} = \frac{\sum d_n}{n}$$

Figure 3

In this example, n is 10 (ten) and d_n is each sales price. So, all values are accumulated in R_0 and then the average is obtained by dividing the total accumulated by the number of entries. Be sure to initialize the contents of R_0 before starting the problem:

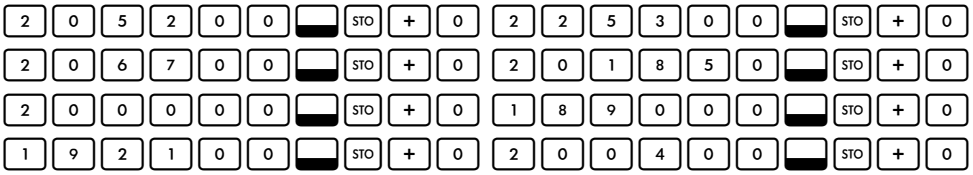
0 ▬ STO 0

Now, each sales price is typed in and added to the existing amount in R_0:

1 9 8 0 0 0 ▬ STO + 0

The second sales price is entered by pressing:

Be sure to key in all sales prices and press STO + 0. When the last sales price is keyed in and accumulated, its value remains in the display:



Now, divide R_0 (the accumulated sales) by ten.

At this point, R_0 contains the resulting value: the average sales price for homes sold in Parkdale community. To display this average, press:

Answer: The average sales price for homes sold in Parkdale community is $ 200,355.00. Of course, this problem is probably more efficiently solved using the statistics data registers and functions, which are described in another learning module, but it does illustrate how memories may be used.