hp calculators

HP 17bII+ Registers / Memory Banks

The Stack Registers

The Storage Registers

HP 17bII+ Registers / Memory Banks

A register is a place where you store information. It is a memory bank where you put information in order to calculate a number or where you want to keep a number for example, that you will need in the future. In this section we are going to look at two sets of registers or memory banks: the stack registers and the storage registers.

THE STACK REGISTERS:

The stack registers are used in Reverse Polish Notation. If you plan to perform all your calculations in the Algebraic mode, there is no need to read on here about the stack registers. However, we suggest you consider RPN, for we think you will find it an efficient system that will save you some keystrokes down the road.

They derive their name from their electronic position that has them "stacked" one above the other. What you see in the display is the 'X' register.   It is the only register you can see. But there are three positions above this 'X' register that also can hold numbers and these are the 'Y,' 'Z,' and 'T' registers, shown as follows:

<div align="center">

T

Z

Y

The display: X

</div>

What you can do in order to see the numbers these three other registers contain is to "pull down the stack." You do this with the R arrow down key that is two keys to the right of the 'INPUT' key. Notice that it is in color, which usually means you have to first hit the shift key to activate this function. But for this key and the color function to its immediate right, the x/y switch key, you don't need to hit the shift key at all if you are in the RPN mode. And, the arrow up and arrow down keys to the left on the keyboard will also move up and down the stack.

<div align="center">

Filling Up the Stack

</div>

To see how these registers work, put your calculator in the RPN mode ('SK,' 'MODES,' and 'RPN'), then press '1' and 'ENTER.' Entering the '1' pushes it up to the 'Y' register, the one right above the 'X' register, and puts another '1' in the 'X' register. For these two steps your stack registers look like this:

<div align="center">

T   0

Z   0

Y   0

X   1

Keyed in: 1

T   0

Z   0

Y   1

X   1

Keyed in: E*

</div>

*E is the Enter button

Every time you key in 'ENTER' it moves the numbers in the stack up one notch. When you key in a number after entering, you replace the number that was in the display ('X' register) with the number you key in.

Put in '2.' It replaces the '1' in the 'X' register:

T   0

Z   0

Y   1

X   2

Keyed in: 2

Hit 'ENTER.' It pushes the '2' up into the 'Y' register, which in turn pushes the '1' up to the 'Z' register, and puts another '2' in the display:

T   0

Z   1

Y   2

X   2

Keyed in: E*

Press '3' and the stack registers will have:

T   0

Z   1

Y   2

X   3

Keyed in: 3

Press 'ENTER:'

T   1

Z   2

Y   3

Z   3

Keyed in: E

Press '4' but don't enter this '4.' Entering '4' would push everything up the stack one notch and you would lose the '1' in the 'T' register as it is knocked out of the stack. And, now you will have these numbers in your stack registers:

           T   1

           Z   2

           Y   3

           X   4

      Keyed in: 4

Your numbers "ascended" the stack registers as shown below in the progression going from left to right:

T	0	0	0	0	0	1	1
Z	0	0	0	1	1	2	2
Y	0	1	1	2	2	3	3
X	1	1	2	2	3	3	4
Keyed in:	1	E*	2	E	3	E	4

      *E is the 'ENTER' button.

Rolling Down the Stack

You have filled up your four stack registers, as shown in the righthand column of the schematic above. You can view each of the numbers by rolling down the stack. Press the R arrow down key and you should see '3.' That was in the 'Y' register directly above and was pulled down to the one register you can see, the 'X' register. The '4' that was in the 'X' register was pulled up to the 'T' register:

           T   4

           Z   1

           Y   2

           X   3

      Keyed in: R*

         *R is the R arrow down key

The numbers actually loop around the stack registers, descending until they reach the 'X' register. The next stop after the 'X' register is the 'T' register, if the R arrow down key is pressed again.

So, press the R arrow down key again and you will see '2.' Hit it again and you'll see '1.' Hit it one more time and you are back where you started. '4' appears in the 'X' register and you have completed the loop of the numbers circling within the four stack registers.

Your numbers "looped" the stack registers as shown in the progression below from left to right:

T	4	3	2	1
Z	1	4	3	2
Y	2	1	4	3
X	3	2	1	4
Keyed in:	R	R	R	R

<u>Dropping Down the Stack With Arithmetic</u>

An arithmetic operation acts on the numbers in the 'X' and 'Y' registers. Let's start adding the numbers, beginning with the '4' in the 'X' register (remember that is the one you are looking at in the display) and the '3' in the 'Y' register above. Press '+' and "7" will appear in the display. You have added the numbers in the 'X' and 'Y' registers.

You began with:
T	1
Z	2
Y	3
X	4

Then, you hit '+' and the stack registers now have:
T	1
Z	1
Y	2
X	7

The addition operation adds the numbers in the 'X' and 'Y' registers, putting the result, "7," in the 'X' register. Gone are the '4' and '3' that have been added to give '7.'

Note that the '2' that was in the 'Z' register has automatically dropped down into the 'Y' register. It does this because an arithmetic operation consumes the numbers in the 'X' and 'Y' registers and puts the answer in the 'X' register that you can see. This leaves a void in the 'Y' register, so the number in the 'Z' register, the one right above the 'Y' register, drops down into the 'Y' register to fill the void.

Same for the number in the 'T' register. When the '2' dropped down from 'Z' register to the 'Y,' register, a void was created in the 'Z' and the '1' in the 'T' register fell to the 'Z' register to fill it.

Hit the '+' again and you'll see '9' in the 'X' register. What has happened? Same thing. The numbers in the 'X' and 'Y' registers have been added, this time the '7' that was in the 'X' register and the '2' that was in the 'Y' register. Remember, that '2' got in the 'Y' register during the first addition when '3' and '4' were added, leaving a vacuum in the 'Y' register to which the '2' "fell" from the 'Z' register. Look at the schematic below, once more reading the progression from left to right:

T	1	1	1	1	1
Z	1	1	1	1	1
Y	2	1	1	1	1
X	7	9	10	11	12
Keyed in:	+	+	+	+	+

The first column shows what you had in the stack registers after hitting the first '+' with '4' in the 'X' register and '3' in the 'Y' register. The answer, "7," appears in the 'X' register.

Sticking in the 'T'

The '1' in the 'T' register dropped down to the 'Z' register just below as it became vacant when the '2' there dropped down to the 'Y' register. The column second from the left in the schematic just above shows what results from hitting another '+.'

But look what happened in the schematic above when a '+' is hit again. And then again. And then again one more time. The three columns to the right show that a '1' is being added each time. Look at the stack registers above the 'X' register. They are filled with '1's.' Why? The answer is that when a number reaches the top rung, the 'T' register, it sticks! Yes, the number in the 'T' register does drop down one to the next level when an arithmetic operation is performed. But it also leaves a duplicate of itself in the 'T' register. That is why you see all those ones in the above schematic.

You can knock a number in the 'T' register completely out of the stack by hitting 'ENTER.' But, when you perform an arithmetic operation and the 'T' number drops down to 'Z,' the 'T' number leaves a trail of itself as it falls down the stack.

Reversing the 'X' and 'Y'

One more point on these stack registers. The number in the 'X' register acts on the number in the 'Y' register. This doesn't make any difference when you are adding or multiplying. '1' plus '2' and '2' plus '1' each come out to '3'. But not so in subtraction and division. '1' minus '2' is not the same as '2' minus '1.' The number in the 'X' register will be subtracted from the number in the 'Y' register. In division, the 'X' register number is divided into the 'Y' register number.

So, when we earlier had '4' in the 'X' register and '3' in the 'Y' register, hitting the '-' key will produce the answer of "-1" in the 'X' register.

Or, instead hit the division sign with '4' in the 'X' register and '3' in the 'Y' register and you'll get the answer of "0.75" in the 'X' register.

Key in '3,', 'ENTER,' '4' and your stack registers will look like this:

T	0
Z	0
Y	3

X   4

If you now key in '÷,' you will have the result of "0.75" ('3' ÷ '4') in the display ('X' register):

T   0

Z   0

Y   0

X   0.75

But, if you key in '4,' 'ENTER,' and '3,' your stack registers will look like this:

T   0

Z   0

Y   4

X   3

And, now when you key in '÷,' your result in the 'X' register will be "1.33" ('4' ÷ '3'):

T   0

Z   0

Y   0

X   1.33

If you want to reverse the sequence of how you have entered numbers into the 'X' and 'Y' registers, press the x/y switch key, which is three keys to the right of the 'INPUT' key. Try it. Clear all your stack registers with 'SK' (shift key) and 'CLR DATA.' Remember that using the other clearing functions - 'CLR' or the sideways arrow key - will only clear the display, that is, the 'X' register.

Now press '3,' 'ENTER,' and '4.' If you want to, hit the R arrow down key to confirm that '3' is in the 'Y' register and '4' in the 'X' register. Then, roll them down so that you bring '4' back to the display. That means that '4' is in the 'X' register and '3' in the 'Y' register.

If you hit '÷' now, you would get the answer you got before, namely '0.75.' Instead, first hit the x/y switch key and then press '÷.' You got a "1.33," didn't you. This is because the x/y switch key put the '4' in the 'Y' register and the '3' in the 'X' register. When you then hit the '÷' key, '3' was divided into the '4,' giving you the "1.33" answer.

The keystrokes for this switch operation:

3
ENTER
4
x/y switch
÷   Answer: 1.33

THE STORAGE REGISTERS:

Now let's take a look at the storage registers. These do what the name says, they store information for you to refer to at a future date. There are ten of them and they are easy to use.

<u>Storing in Ten</u>

Do you see the 'STO' key, in the second row to the far left? Press that key any time
you want to get in to any of the ten electronic storage slots. These are numbered '0' through '9' and you go in by hitting 'STO' and the number of the storage register that you want.

Let's say, for example, you want to store '100' into storage register number 1. First, press in the number that is to be stored, namely '100,' then hit 'STO' and '1.' You have now stored '100' into storage register 1 with these keystrokes:

100
STO
1

<u>Peeking Inside</u>

You want to make sure that '100' really is in storage register 1? Clear the display, so that you won't confuse that '100' in the display with the '100' in storage register 1. Of course, you know if you have covered the information on the stack registers, that the display is a totally different register from a storage register. The display is the 'X' register of the stack registers.

Once you have cleared your display by using any of the clearing functions, you are ready to "look into" the storage register. To do so, press the recall button, which is the 'RCL' key just to the right of the storage key. After hitting 'RCL' press the number of the storage register you want to check, that is, number '1.' The keystrokes:

RCL
1

Do you see "100?"

By the way, you really didn't have to clear the display before recalling a number from the storage register. We did it just so that you could separate the '100's that were in both the display and storage register 1.

<u>Cleaning House</u>

Did it occur to you that the number in the storage register was not cleared when you
performed your clearing function? Even using the most sweeping of the clearing operations, 'SK' and 'CLR DATA,' doesn't clear the storage registers. That is, of course, intentional, in order to guard against mistakenly clearing any of ten storage registers while trying to clear a menu of numbers.

<u>Just a Copy</u>

One more point. When you recall a number, you bring it up to the 'X' stack register, that is, the display. But you don't remove it from the storage register. It is still there. You just made a copy of it so that you could see it in your display. How do you remove it from the storage register, if you want to? Simply press '0,' 'STO,' and the number of the storage register. That "zeroes out" the storage register by kicking out the number that was there and replacing it with zero.

<u>Storage Register Arithmetic</u>

And now one final point about storage registers. You can perform arithmetic operations in them without ever having to bring the stored number up to your display in what is called "storage register arithmetic." Shall we add '50' to the '100' you have in storage

register 1? Why not. Plug in '50' and hit 'STO.' Then, tell your calculator the arithmetic operation you want, addition, by hitting the '+,' and, finally, tell your calculator to what storage register you want to add this '50,' namely '1.' Your storage register arithmetic keystrokes with '100' already in storage register 1:

<div align="center">

50
STO
+
1

</div>

Check it out. Hit 'RCL' and '1'. You should be looking at "150," the result of your adding '50' to the '100' in storage register 1 with storage register arithmetic.