



hp calculators

HP 33S Writing a Simple Program

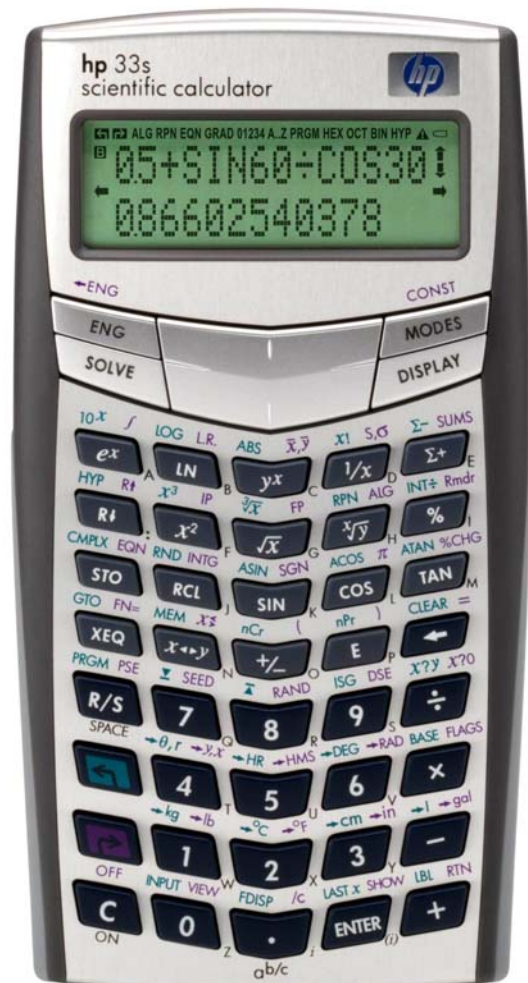
Programming the HP 33S

Practice example: the area of a circle

Tools for programming

Practice example: the sinc function

Using a program in another program



Programming the HP 33S

Doing a simple calculation once on the HP 33S is easy. Doing the same calculation many times, or doing a complicated calculation, takes longer. It can be better to store all the steps needed for the calculation in a **program**. A program is a set of instructions, stored all together. Once it is written, it can be tested to see if it works correctly. Then it can be used many times, without the need to press every key of the calculation each time.

A simple program is just a set of keystrokes stored so that they can be carried out with one key. The HP 33S provides many commands to let programs do more, for example stop and ask for input, or show an intermediate result. This training aid concentrates on simple programming; it also shows a few of the more advanced programming commands.

Practice example: the area of a circle

Given “r” the radius of a circle, the circle’s area “A” is calculated from the formula $A = \pi r^2$. For example to calculate the area of a circle with radius 3cm, the following keys are pressed.

In RPN mode: 

And in algebraic mode: 

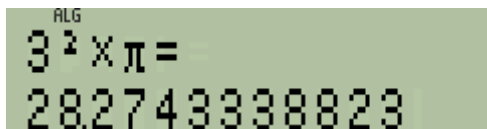






Figure 1

Repeating the same 4 keys in RPN mode or 5 keys in algebraic mode for many circles is a lot of work. Here is how a program would make it easier.



Example 1: Write a program in RPN mode to calculate the area of a circle, given its radius r.

Solution: An RPN program and an algebraic program are shown below. The RPN program is shown first, called “A” for “Area”.

In RPN mode, type the same keys for calculating the area as before, but mark the beginning of the program with a **label** and the end of the program with a **return**.

First set program mode by typing  (PRGM) and then go to the top of program memory (called PRGM TOP, see Figure 2), ready to begin a new program, by typing  (GTO)  . Now the program can be typed:

 (LBL)  (A)     (RTN)

The letter A is on the keyboard, below and to the right of the  key. Whenever  is typed, the HP 33S uses the keys with the letters A – Z marked next to them to enter those letters.

The program will look as shown in Figures 2 through 4 below.



Figure 2



Figure 3



Figure 4

Each line of the program begins with the letter of the label, and then has a four-digit line number. The rest of the program line is a function or a program instruction. The program instructions are LBL A and RTN, the functions are x^2 , π and \times .

To use the program, it is necessary to move out of program mode, and to return to the beginning of the program. Then a radius is typed and the **[R/S]** key is pressed. The HP 33S **executes** (or **runs**) the program.

To calculate the area of a circle with radius 5, make sure RPN mode is set, then press:

[←] [PRGM] [→] [RTN] [5] [R/S]

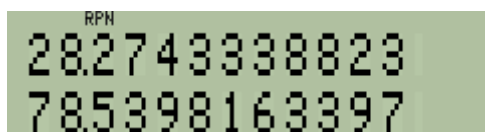


Figure 5

Answer: The area of a circle with radius 5 cm is just over 78.53 square cm.

To calculate the area of another circle with a different radius, it is enough to type the new radius and press **[R/S]** again. The program works like a new function, executed by means of the **[R/S]** key instead of a key with the name of one of the functions built into the HP 33S.

Example 2: Write a program in RPN mode to calculate the area of a circle, given its radius r .

Solution: In algebraic mode, the program looks very similar, but with the calculation keys typed in algebraic order. The HP 33S does not allow more than one program to have the same label, so the algebraic program below is given the label B.

Go to the beginning of program memory by typing **[←] [GTO] [0] [0]** and set program mode by typing **[←] [PRGM]**. Then type the algebraic mode program:

LBL ALG ENTER RTN

To get press the key; it has a small B to its lower right. The program lines are as listed below.

```

B0001      LBL B
B0002      ALG
B0003      x2
B0004      ×
B0005      π
B0006      ENTER
B0007      RTN

```

If a program in algebraic mode performs a complete calculation then it is best to finish it with so that calculations made after the program will not be treated as part of the program's calculation.

The second line is an instruction to set algebraic mode, to make sure program B is not used accidentally in RPN mode. If the user always works in algebraic mode, or always remembers to switch to the right mode before using a program then this program line is not needed, but it is always safer to use it. A similar line, setting RPN mode, could have been used in program A for the same reason.

This program works just like the RPN version. To use it, first move out of program mode, and to go to the beginning of the program. Then a radius is typed and the program is executed.

Therefore, to calculate the area of a circle with radius 5, press:

PRGM RTN



Figure 6

Answer: As in RPN mode, the area of a circle with radius 5 cm is calculated at just over 78.53 square cm.

To calculate the area of another circle with a different radius, it is enough to type the new radius and press again. Like the RPN program, this program works similarly to a new function, executed by means of the key instead of a key with the name of one of the functions built into the HP 33S.

Tools for programming

The HP 33S allows the user to create any or all of the labels from LBL A through LBL Z. The lines after a label all have that label at the beginning of their line numbers. When a new label is typed, the HP 33S starts over again at line 1, with the new label's name at the beginning of each line. Figure 7 shows an example of this; if both program A and program B were typed, then after the RTN in program B, the first line of program A is displayed on the calculator screen.




Figure 7

HP 33S Writing a Simple Program

Each label can be used only one time, and the HP 33S treats the lines from one label until the next label as a separate program. It is also possible to have a program with no label, from the “PRGM TOP” until the first label.




To list the programs in the HP 33S, press the keys   to get the “Memory” menu, and then  to get a display of programs. If the algebraic version of the circle program has been entered then the calculator screen will look like Figure 8. If only program A was typed, the screen will be as in Figure 9.



Figure 8


This means that the program with label B is the first program in memory, and that it is 21 **bytes** long. A byte is a piece of calculator memory made up of 8 **bits**, and a bit is the smallest piece of memory, a single “0” or “1”. The HP 33S has more than 31,000 bytes of memory available to hold programs and equations entered by the user. Given this much memory, a long program could have up to 9,999 program lines, and that is why program line numbers are 4 digits long. The up and down arrows at the right of the screen mean that there are more programs in the calculator – information about each program in turn is displayed if the up or down arrow keys are pressed. If both the programs A and B have been entered, press  to see information about program A.



Figure 9





Another tool to help in writing programs is the **program checksum**. To see the checksum for program A, press  . While the  key is held down, the calculator will show:



Figure 10

CK=DEFD means that the checksum for program A is the hexadecimal number DEFD. (The checksum for program B is 8A0B.) If the program was not typed correctly, then the checksum will be different. Press the key labeled  at the bottom left of the keyboard to cancel the display of information about programs.

The checksum has several purposes:

- If a program is correctly copied to the HP 33S from a web page or a book – the HP 33S manual has many useful example programs – then the checksum on the HP 33S will be the same as the checksum given with the program. If the checksum is not the same, a mistake was made and the program should be checked and corrected before it is used.
- If a user creates a new program on an HP 33S and then writes it down for future use, it is worth keeping a note of the checksum with the written program. That way, the program can be checked if it is typed into the HP 33S again later.

HP 33S Writing a Simple Program

- If a user wants to offer a program to other users, the checksum should be given with the program, so the other users can check if they have entered the program correctly into their HP 33S.

The checksum can be very helpful, but it does not have to be used – for short programs such as the above it can be enough to step through the program with the up and down arrow keys and correct any mistakes.

Mistakes can be either wrongly typed lines, or missing lines. To add a missing line, use the \uparrow and \downarrow keys until the line before the missing line is at the bottom of the screen. Then type the missing line or lines. If a line is wrong, move it to the lower line of the screen and use the \leftarrow key to delete one or more lines, then type the correct line or lines.

Even after all typing errors have been corrected, a program can still be wrong if it was designed wrongly. The HP 33S manual provides advice on program testing and describes more programming tools.

To go to a line in a program, to see it or to change it, use the GTO command, followed by a dot and then the program name and the line number. For example to go to line 5 of program A type \leftarrow GTO \cdot A 0 0 0 5.

The LBL and the RTN help to identify each program. If a program is executed by the user pressing R/S then the RTN goes back to the top of program memory, where the program stops. If both program B and program A are in the HP 33S, and program A is used, then the RTN at the end will go to label B at the top of program memory. Next time R/S is pressed, program B will begin, not program A. This means that only the program at the top of program memory will work like a function key when R/S is pressed. To execute program A if it is not at the top of memory, press XEQ A.

Practice example: the sinc function

Example 3: As a third example, write a program to calculate the Sinc function $\text{Sinc}(x) = \sin(x) \div x$.

At first sight this looks little different from the previous program, but there is one difficulty – the program needs to deal with the case of $x = 0$ when division by zero gives an error. The program needs to test for $x=0$ and to handle that case specially. This example will show how tests are used in programs.

Solution: Again write the program so it will use a number x that has been typed before the program begins. A program like this will probably be used like an additional function on the calculator, so it should work both in RPN mode and in algebraic mode, in the same way as functions built into the HP 33S. To do this, the program will use a **data register**. Data registers are described in detail in a separate practice aid. The program will also use a flag, flags are also described in a separate practice aid.

As before, first set program mode by typing \leftarrow PRGM and then go to the top of program memory by typing \leftarrow GTO \cdot \cdot . Try storing the new program at the end of program memory. To do that, press the up arrow key to go from the top of program memory to the last program instruction. If program A was typed as above, the last instruction will be the RTN at its end. Give the new program the label “C”, using the last letter of “Sinc” to name it. Type \leftarrow LBL C. To type C, press y^x after pressing LBL.



Figure 11

HP 33S Writing a Simple Program

Store the number in a data register, for this example use data register “A”, but any register could be used. To type \boxed{A} , press $\boxed{e^x}$ after pressing \boxed{STO} . Then clear flag 0, which will be used later. To do this press \boxed{F} \boxed{FLAGS} to show the flags menu, press $\boxed{2}$ to clear a flag, and press $\boxed{0}$ to specify that flag 0 is to be cleared.



Figure 12

Now, if the number x is zero, set flag 0. This will let the program check later if x is zero, and will also display a little zero at the top of the screen, to tell the user that sine is being calculated for x equal to zero.

To make a decision like this in a program, it is necessary to use a test function, in this case the test is $x=0$? Press \boxed{F} $\boxed{x=0}$ to get a menu of test functions. Then press $\boxed{6}$ to get the $x=0$? test. If the test is true, flag 0 must be set. To do this press \boxed{F} \boxed{FLAGS} to show the flags menu, press $\boxed{1}$ to set a flag, and press $\boxed{0}$ to specify that flag 0 is to be set. This means that the program will test x to see if it is zero. If the test is true and x is 0 then the next line will be carried out and flag 0 will be set. If x is not 0 then the test is not true and the next step will be skipped, so flag 0 will be clear, because it was cleared at step C00003.

This is the way all tests work in HP 33S programs – do the next step if the test is true, or just **do if true**.

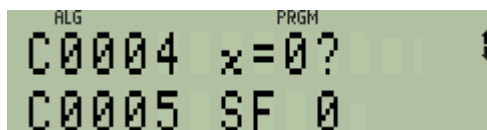


Figure 13

Now calculate the sine of x. If x is zero, then the sine will be zero; nothing will be changed. There is no need for a test, just press $\boxed{\sin}$. After the sine has been calculated, the result can be divided by the original number, at present stored in data register A, but first the number must be brought back from A, while the sine is stored in A. To do this, exchange the sine with A. Press \boxed{F} $\boxed{x\leftrightarrow y}$ \boxed{A} . Once again, press the $\boxed{e^x}$ key to enter the letter A. Figure 14 shows the program steps.

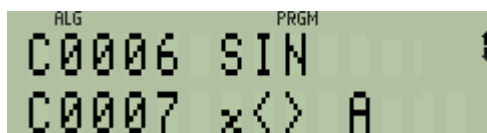


Figure 14

Now the sine can be divided by the original number, but this should only be done if the number is not zero. The program must test x to see if it is not zero. Press \boxed{F} $\boxed{x\neq 0}$ to get a menu of test functions. Then press $\boxed{1}$ to get the $x\neq 0$? test. Then press \boxed{STO} $\boxed{\div}$ \boxed{A} to divide the original number into the sine in register A. If the test is true and x is not 0 then the following line will be carried out and the division will be performed. If x is 0 then the test is not true and the next step will be skipped. Again the program will follow the rule **do if true**. If the original number was zero, then the division is not performed and there is no error caused by a division by zero.

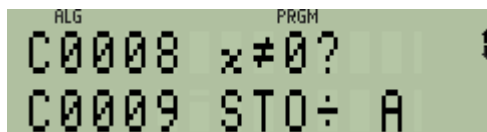


Figure 15

The result is now ready but it is in A and has to be brought back, so another exchange with A is needed. Press $\boxed{\rightarrow}$ $\boxed{x\leftrightarrow y}$ \boxed{A} again, and the answer is ready, if the original number was not zero.



Figure 16

If the original number was zero then it has not been changed, and the result of all the above work is a zero. However, the result might also be a zero if the calculations produced a number smaller than 10^{-499} . The HP 33S replaces numbers this small with zero. So the result should not be tested, instead flag 0 should be tested, and if it is set then the original number was zero. In that case, it must be turned into a 1. First press $\boxed{\rightarrow}$ $\boxed{\text{FLAGS}}$ $\boxed{3}$ $\boxed{0}$. Then the 0 can be turned into a 1 using one of several functions that do this, for example $\boxed{e^x}$.

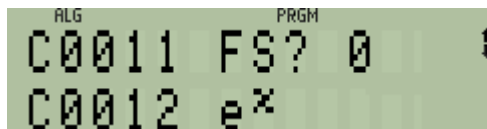


Figure 17

To finish the program, press $\boxed{\rightarrow}$ $\boxed{\text{RTN}}$, then leave program mode by typing $\boxed{\leftarrow}$ $\boxed{\text{PRGM}}$.

The program can be tested with a few values of x. The definition of $\text{Sinc}(x)$ as $\text{Sin}(x) \div x$ assumes that x is in radians. (The program does not set radians mode, as a user might intentionally want to calculate $\text{Sin}(x) \div x$ in degrees or gradians.) Set radians mode if it is not set already, and calculate the sinc of 0.2.

$\boxed{\text{MODES}}$ $\boxed{2}$ $\boxed{0}$ $\boxed{\cdot}$ $\boxed{2}$ $\boxed{\text{XEQ}}$ $\boxed{\text{C}}$

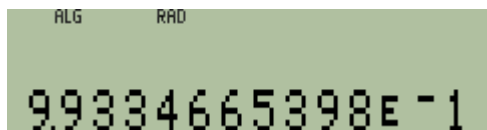


Figure 18

Confirm that the program works for 0. The program works both in RPN and algebraic modes, so try this in RPN mode:

$\boxed{\leftarrow}$ $\boxed{\text{RPN}}$ $\boxed{0}$ $\boxed{\text{XEQ}}$ $\boxed{\text{C}}$

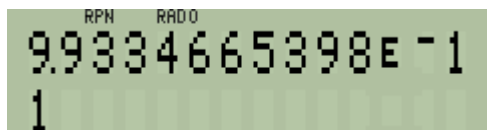


Figure 19

HP 33S Writing a Simple Program

Answer: The program works correctly in RPN and in algebraic modes, even for the special case where x is 0. Furthermore, it works so that the original value of x can be recovered with the **LASTx** key.

The program is 42 bytes long and its checksum is D11C. At 13 steps, it is longer than the “area of a circle” program, but not very long. Extra steps were needed to make it work for the special case, and to use a data register for the calculation, so the program works both in RPN mode and in algebraic mode. This means that the program can be used by other programs that need to calculate sinc.

Using a program in another program

Example 4: Use program C, written above, to calculate a sinc function value inside another program.

Solution: This is very similar to writing a program which calculates a sine as part of a longer calculation. A program which needs to calculate the sine of 1.3 radians might have the following lines in it:

```
D0001    LBL D
D0002    RAD
D0003    1.3
D0004    sin
D0005    ... more program lines ...
```

A program to use the sinc function, provided by program C, would look like this instead:

```
D0001    LBL D
D0002    RAD
D0003    1.3
D0004    XEQ C
D0005    ... more program lines ...
```

As this shows, if another program needs to calculate a sinc value, it can use program C like an HP 33S function. First the number x is typed. Then, Instead of typing a function name, the user must **call** the sinc program by typing XEQ C. When the calling program is run the sinc program will do its calculation. When that reaches the RTN at its end, it will return to the program that called it, which will continue from the next line. When a program is called from another program like this, it is called a **subprogram** or a **subroutine**.

If a program is used as a subroutine, called from another program, then the RTN at its end finishes the subroutine and goes back to the program that called the subroutine. As was explained in the section on tools for programming, if a program is not called up from another program, but is started by the user pressing **R/S**, or **XEQ**, then the **RTN** goes back to the top of program memory, where the program stops.

Programs can do much more than has been described in this training aid. The reader should consult the HP 33S manual and try some of the many example programs in it.