To test that the program has been recorded correctly, erase the computer's memory by typing NEW‹CR›. Rewind the cassette, put it into the play mode and then load the program back into the computer using the LOAD command. LOAD must be followed by the file name of the file wanted. Type LOAD "TILES" and then hit RETURN.

After the program has been loaded into the computer, a message on the screen such as READY or OK indicates that the load has been completed. LIST the program and check that it is the same as the one you typed in.

GOSUB

GOSUB is a statement that diverts the flow of a program to a subroutine. A subroutine is like a separate mini-program or program-within-a-program. In the program used to illustrate it here, the subroutine is very simple. It is included to show the principle, although other ways could easily have been devised to produce the same results without using a subroutine.

Our program calculates the number of tiles needed to tile a room by finding out the area of the tiles used. It then asks for the length and height of each wall to be entered. It works out the area of the wall after the length and height has been converted from metres to millimetres. The number of tiles needed is found by dividing each wall's area by the area of a tile and adding the results. The conversion of wall length and height into millimetres is done in the subroutine, which simply multiplies the length or height (in metres) by 1000 to find the equivalent in millimetres.

Subroutines have three advantages. Frequently used parts of programs can be separated off and only need to be written once — no matter how often the operation is required. They allow long and complex programs to be broken down into more manageable and easily understood units or sections. Finally, subroutines can be re-used in any program where its function is appropriate.

In our program, the subroutine starts at line 380 and consists of only one statement: LET D2 = D * 1000. This takes D, the wall dimension (length or height) and multiplies it by 1000 to convert from metres to millimetres. The result is assigned to variable D2.

The instruction that forces the program to go to the subroutine is GOSUB. It occurs first in line 160. Variable D was assigned the value of the length of the wall in line 130. Line 160 forces the program to go to the subroutine, where variable D2 is given the value of D multiplied by 1000. The RETURN instruction in line 390 is needed to make the program return from the subroutine to the main program. Subroutines always return to the line after the GOSUB statement, in this case, to line 170.

The next occurrence of GOSUB is in line 240, which 'calls' the same subroutine. This time, the subroutine RETURNs to line 250. Although this program uses only one subroutine, it is possible to use as many as are needed. In every case, the GOSUB statement will have to include the line number of the appropriate subroutine. Notice that the END statement occurs in line 370, before the subroutine. END indicates the end of the main program and also serves to stop the program from running on through the subroutines after it has been completed.

Although this program is a little longer than previous programs in this course, it is really no more complex. Try and follow it through, line by line, and see what is happening at each stage. Apart from GOSUB and subroutines this program introduces only one new concept — longer variable names.

It may be helpful to draw boxes with the variable names written on them and to write in the values at each stage.

Line 300: LET S = A3/A2 will sometimes give a number with a decimal fraction. Try running the program and entering the tile size as 110mm and the wall length and height as 2.3 and 1.8 metres respectively, using just one wall. You should get an answer of 342.149 tiles. Since tiles are never sold in units of less than one, this answer is not completely appropriate. Next time we will look at one of the ways of getting an appropriate answer in whole numbers.

Exercises

- See what happens if you enter the size of the tile as 0mm. You should get an error message at the end of the run. Why is this? Why don't you get a similar error message when you enter the length of one of the walls as 0 metres? Hint: multiplying by zero and dividing by zero are not the same thing — try it on your calculator!
- The program only works for square tiles. See if you can change lines 30 to 60 to find the area of rectangular tiles (just as we found the area of rectangular walls later in the program).
- Add a statement at line 355 to increase the total number of tiles by five per cent to allow for wastage. Multiplying a number by 105/100 will increase it by five per cent.

Basic Flavours

END
This is not available on the Spectrum, ZX81, Oric-1 or Dragon 32, so replace it by STOP

LET
Line 45 is required on Sinclair machines and on the BBC Micro. On most other computers it can be omitted

GOSUB
Appears as two words on the Spectrum and ZX81, although only one keystroke is required