

19 Functions

Functions are similar to procedures, but they have only one purpose - to give a single result. The easiest way to understand a function is to describe some of the computer's own. It has lots - the trigonometric functions such as **SIN**, **TAN** and **COS**. One of the most useful functions is **RND**, which supplies random numbers. It is usually used with a parameter, and gives a random integer between 1 and the value of the parameter. So, **RND(50)** will pick a random number between and possibly including, 1 and 50. When you type **X = RND(4)**, you know that the result of the function **RND** will be placed in **X**. The **RND** function is described in more detail in chapter 25.

A function can be used with any number of parameters, both string and numeric. Here is a function to determine the mass of a sphere:

```
100 DEF FNmass_of_sphere(radius,density)
110 = 4/3*PI*radius^3*density
```

Here's another example of using a function in a program

```
5 CLS
10 REM Discount calculator
20 PRINT "''''''This program calculates the
  following discounts:"
30 PRINT "''20% on £100 or less"
40 PRINT "''30% on £101 to £200"
50 PRINT "''50% on anything over £200"
60 INPUT "''''Enter the sum      £" Y
70 PRINT "''''Final sum with discount is
£";FN_discount(Y)
80 END
100 DEF FN_discount(SUM)
110 IF SUM <= 100 THEN =SUM - (20*SUM/100)
120 IF SUM > 100 AND SUM <= 200 THEN =SUM -
  (30*SUM/100)
130 IF SUM > 200 THEN =SUM - (50*SUM/100)
```

The main program starts at line 5 and ends at line 80.

88 Functions

Line 5 clears the screen, and lines 20 to 50 print instructions on the screen.

Line 60 prints a request for you to enter an amount, waits for you to do so, and puts the value into variable Y.

Line 70 prints a message, and calls a function called **FN_discount(Y)**. The value in Y is passed to the function's parameter, (which is the 'actual' parameter).

Line 100 starts the definition of the function, and passes the parameter value to a 'formal' parameter called **SUM**.

Line 110 contains a conditional statement. If the value of **SUM** is 100 or less, then the function returns the result given by **SUM - (20*SUM/100)**. If the value of **SUM** is more than 100, then the execution of line 110 stops before working out the **SUM - (20*SUM/100)**, and line 120 has a go - and so on.

Notice the underline character in **FN_discount**. This helps to make the function's name more readable.