13 Arrays

Arrays are groups of variables. An array variable has a name, just as any other, and it also has one or more subscripts. A subscript is a number, and an array variable is numbered according to its position in the array. For example, **A(0)** is the first variable in the away named **A**, and **A(1)** is the second variable. The computer must be told how many variables you wish to use in the array, and this is done by using a DIM instruction:

```
DIM A(9)
```

allocates space in the computer's memory for ten variables, each called A, but each having a different subscript.

These variables are **A(0)**, **A(1)**, **A(2)** . . . **A(8)**, **A(9)**. Each one of these variables may be individually assigned, just like any ordinary variable.

String arrays may also be used.

```
DIM A$(9)
```

allocates space for ten string variables – each of up to 255 characters.

The examples shown above are one dimensional arrays – you can think of them as a line containing a number of variables, subscripted from 0 to 9 in sequence. Two dimensional arrays can be thought of as the printing on the TV screen. Each character printed on the screen is at a particular position from the left, and a particular position from the top.

```
DIM A(2,2)
```

allocates space for nine variables, each called A, and each having two subscripts:

```
A(0,0)  A(1,0)  A(2,0)
A(0,1)  A(1,1)  A(2,1)
A(0,2)  A(1,2)  A(2,2)
```

Arrays may have as many dimensions as you like, may contain as many variables as you like, and may be either real, integer, or string.

One of the invaluable features of an array variable is that its subscript need not be specified as a number; you can use a variable:

```
10 DIM A(9)
20 X = 6
30 A(X) = 27
```

This program will place 27 in A(6). You can even use an array variable as the subscript:

```
10 DIM A(29)
20 X = 6
30 A(X) = 27
40 A(A(X)) = 564.3
```

Any arithmetic expression may be used as a subscript, and if the subscript works out to a number with a decimal point, then the number is truncated to its integer value, ie just the part before the decimal point.

Remember, when using arrays, that if you DIM using three subscripts, each variable must be called with three subscripts.

```
10 DIM NAME$(2,2,2)
20 NAME$(0) = "FRED"
```

will not work. You would have to use, say,

```
20 NAMES(0,0,0) = "FRED"
```

Be careful when using arrays they consume vast amounts of memory, and if you try to use too many variables the computer will say,

```
DIM Space
```

meaning that there isn’t enough room in its memory.