This is much more readable, and is much less prone to error than either a GOTO loop or a dummy FOR..NEXT loop.

BBC BASIC has the useful debugging aids TRACE, ON ERROR..., and ERL. TRACE causes program line numbers to be displayed on the screen as they are executed; ON ERROR GOTO (or GOSUB), means that any normally fatal error (including pressing the ESCAPE key) during program execution will cause control to pass to a user-defined error-handling routine (such as a dump of all variable values). ERL is a system variable containing the number of the line on which an error occurred.

The BBC has a wealth of unique extensions to BASIC, such as the Operating System calls, the VDU command, the various system variables and the Assembler, which are themselves sufficiently complex to be the subjects of individual articles later in the course.

BBC Graphics

The BBC BASIC commands that relate directly to graphics are:

MODE

This selects the computer display mode with MODE N where N = 0 to 7:

Mode	Graphics	Colours	Text
0	640×256	2	80×32
1	320×256	4	40×32
2	160×256	16	20×32
3	text only	2	80×25
4	320×256	2	40×32
5	160×256	4	20×32
6	text only	2	40×25
7	Teletext		40×25

BBC Model A computers can access only modes 4, 5, 6 and 7; Model B has access to all modes. In modes 0 to 6, the character set can be changed by the user with the VDU command. Mode 7 Teletext characters are fixed and don't correspond to standard ASCII code.

COLOUR

This sets one of 16 colours for text and background according to the mode selected with:

COLOUR N

where N is 0 to 15 for text colours and 128 to 143 for background colours. The colours set by each value of N do not remain constant from mode to mode. Lists of N values related to colours for each mode are given in the user guide.

VDU

This is a highly useful command. VDU A is equivalent to PRINT CHR$(A). Similarly, VDU A,B,C has the same effect as PRINT CHR$(A); CHR$(B); CHR$(C);. This means that the many complex text and graphics routines under the control of the 32 CHR$ codes, duplicating the effects of most of the graphics-related BASIC commands, can be constructed with a small number of VDU commands.

CLG

This clears the graphics area of the current screen and moves the cursor to its 'home' position at the bottom left of the screen.

CLS

This clears the text area of the screen and moves the text cursor to its 'home' position at top left. Any graphics on the screen will also be cleared.

DRAW

This draws lines on the screen in modes 0, 1, 2, 4 and 5. Constructed as:

```
DRAW X,Y
```

The point defined by the X and Y co-ordinates is the end of the line. The starting point can be either the end point of the last line drawn, or a point defined by a MOVE command.

GCOL

This sets the current graphics foreground and background colours with:

```
GCOL N,M
```

where N sets how colour is to be used (0 to 4) and M defines the logical colour using the same principles as COLOUR. N has the following effects:

0 – Plot the colour specified by M
1 – OR M colour with present colour
2 – AND M colour with present colour
3 – Exclusive–OR M colour with present colour
4 – Invert present colour

MOVE

This positions the graphics cursor to a specified point with:

```
MOVE X,Y
```

It has the same effect as DRAW but without drawing a line.

PLOT

This can be used for many graphic functions including point, line and triangle drawing. Constructed as:

```
PLOT K,X,Y
```

where K defines the type of graphic PLOTted. K takes values in the range 0 to 225 to specify the type of lines to be drawn and the colours they take according to lists given in the user guide.

POINT

This gives the number relating to the logical screen colour of the specified screen co-ordinate with:

```
NUMVAR = POINT(X,Y)
```

where NUMVAR is a numeric variable.