A Matter Of Style

Now that we have covered the fundamental rules of Basic, we can concentrate on important aspects of programming style and some new commands to perfect programming technique

The computerised address book program we have developed in previous instalments of the course uses many of the more important features of the BASIC language, but certainly not all of them. In the concluding parts of the Basic Programming course we will look at where BASIC can take you next if you wish to become an advanced programmer. Unfortunately, this cannot be exhaustive, and readers are advised to refer to the owner's manual, or one of the many supplementary books that have been published for most of the popular home computers, for more extensive analysis of their machine's version of BASIC.

Machine Language Programs

Most versions of BASIC allow routines written in machine language to be included as part of the program. Broadly, there are two ways of doing this. The simplest is to use PEEK and POKE. PEEK is a statement used to examine specific memory addresses. For example, LET X = PEEK(1000) will get the value stored in address location 1000 and assign it to the variable X. Executing PRINT X will then print the value that was (and still is) in location 1000. Here is a short program that will PEEK at the contents of 16 memory locations and print them out on the screen:

```
10 INPUT "ENTER 'PEEK' START ADDRESS";S
20 PRINT
30 FOR L = 1 TO 16
40 LET A = PEEK(S)
50 PRINT "LOCATION ";S;" CONTAINS: ";A
60 LET S = S + 1
70 NEXT L
80 PRINT "PRESS SPACE BAR TO EXAMINE NEXT 16
   LOCATIONS"
90 PRINT "OR RETURN TO END"
100 FOR I = 1 TO 1
110 LET C$ = INKEY$
120 IF C$ < > CHR$(13) AND C$ < > " " THEN
    I = 0
130 NEXT I
140 IF C$ = CHR$(13) THEN GOTO 160
150 GOTO 30
160 END
```

The loop in lines 100 to 130 checks the input from the keyboard and then either goes to the end of the program, if the character typed was a RETURN (13 in ASCII), or back to the beginning, skipping the INPUT statement.

If desired, the ASCII character of the memory location can also be printed by using PRINT CHR$(A). But be careful, as ASCII values lower than decimal 32 (ASCII for the 'space' character) are not uniformly defined. All ASCII values from 0 to 31 represent non-printable characters or special functions, such as cursor controls. About the only agreement between different computer manufacturers is that ASCII 13 is usually the carriage return and ASCII 7 sounds the internal speaker or produces a 'beep'.

POKE is the converse of PEEK. It allows you to write any value from 0 up to 255 in any RAM memory location. This facility must be used with extreme caution, however, as writing to a part of memory that is already being used by the program can cause unexpected or catastrophic results. Routines written in machine code can be POKEd to the appropriate addresses and invoked when the program is run by the CALL statement. How to write programs in machine code is beyond the scope of a course on BASIC. Suffice it to say that machine code runs very much more quickly than even the best BASIC dialects. In situations where speed of execution is essential, or where great precision is required, machine code is by far the better alternative.

Moving The Cursor

Many home computers now allow locations on the screen to be addressed directly, but even if your machine does not support this, it is possible to move the cursor to the left, right, up and down the screen relatively easily. First you need to know what ASCII codes are used to represent the cursor control keys. The following short program will ask you to type a key and will then report the ASCII value corresponding to that key:

```
1 REM FINDING THE ASCII CODES FOR THE CURSOR
  KEYS
10 PRINT "PRESS A KEY";
20 FOR I = 1 TO 1
30 LET K$ = INKEY$
40 IF K$ = "" THEN I = 0
50 NEXT I
60 PRINT ASC(K$)
70 GOTO 10
80 END
```

This routine will also allow you to find the code for the RETURN key (usually 13), ESCape (usually 27) and the space key (usually 32), in addition to the codes for the cursor control keys. The Sord M23 computer, on which the programs in the Basic Programming course were developed, uses the