```
1020 LET C = 5
1030 LET D = B + FNV(C)
1040 PRINT D
1050 LET G = FNV(16)
1060 PRINT G
```

Some BASICS allow multiple variables to be used in the defined function. Thus, a function to find the average of two numbers could be written:

```
110 DEF FNA(B,C) = (B + C )/ 2
110 INPUT "ENTER TWO NUMBERS";B,C
120 LET A = FNA(B,C): REM THE 'AVERAGE'
  FUNCTION
130 PRINT "THE AVERAGE OF ";B;" AND ";C;" IS ";A
```

Notice that line 110 above combines the equivalent of two separate statements in one. Most BASICS will automatically print words appearing in double quotation marks following the INPUT statement, so this line is equivalent to:

```
110 PRINT "ENTER TWO NUMBERS"
115 INPUT A,B
```

Line 120 also manages to get the equivalent of two statements in one line by using the colon (:) separator. Statements that would normally belong on separate lines may be written on one line provided each 'stand-alone' statement is separated from the preceding one by a colon. This can help to save space in long programs, but its use is not to be encouraged as it makes programs less readable and mistakes more likely.

We have now covered all the main points of the BASIC language. In forthcoming parts of the Basic Programming course we shall look at program development and program design, rather than at the details of BASIC.

Answers To Exercises On Page 197

Bugs

An 'OUT OF DATA ERROR' will result, because there should be a total of 12 values in the DATA statement in line 130. Secondly, an error will arise in line 100 when it tries to address an element A(4,1). Line 100 should read:

```
100 PRINT A(X,Y)
```

Assigning Values

Below is one version of a program that will perform the required functions. Your own program may look different

```
10 DIM A(8,13)
20 FOR R=1 TO 7
30 FOR C=1 TO 12
40 READ A(R,C)
50 NEXT C
60 NEXT R
70 REM ADD TOTALS
80 GOSUB 300
90 REM PRINT REQUESTED DATA
100 GOSUB 200
110 PRINT "MORE DATA?"
120 PRINT "Y OR N"
130 INPUT A$
140 IF A$="N" THEN GOTO 160
150 GOTO 100
160 END
200 PRINT "WHICH MONTH?'
210 PRINT "1-FOR JANUARY,"
220 PRINT "13 FOR TOTAL, ETC"
230 INPUT M
240 PRINT "WHICH EXPENSE?"
250 PRINT "1-FOR PETROL"
260 PRINT "8-FOR TOTAL, ETC"
270 INPUT X
280 PRINT "VALUE IS ";A(X,M)
290 RETURN
300 FOR R=1 TO 7
310 LET T=0
320 FOR C=1 TO 12
330 LET T=T+A(R,C)
340 NEXT C
350 LET A(R,13)=T
360 NEXT R
370 FOR C=1 TO 13
380 LET T=0
390 FOR R=1 TO 7
400 LET T=T+A(R,C)
410 NEXT R
420 LET A(8,C)=T
430 NEXT C
440 RETURN
500 REM YOUR DATA FOLLOWS HERE
510 REM EIGHTY-FOUR VALUES
520 REM 'DATA 11.35, 9.87' ETC
```

Basic Flavours

ASCII

The Dragon-32 has a non-standard version of ASCII, which does not permit lower case characters, so the lower to upper case converter program will not convert lower case to upper case; try it anyway, and read your manual for more information about the Dragon-32 character set.

DEF FN(A,B)

On the Oric-1 the Vic-20, the Dragon-32 and the Commodore 64 you cannot write more than one variable inside the brackets.

ASC( )

On the Spectrum and the ZX81, replace:

```
ASC(A$) by CODE A$
```

and replace:

```
CHR$(65) by CHR$ 65
```

CHR$( )

If the argument is an expression, it must be put in brackets. Simple arguments — like A$ and 65 — do not need brackets, however.

ON... GOSUB

ON... GOTO

These statements are not available on the ZX81, Spectrum or Lynx.

INPUT" "

On the BBC Micro replace

```
INPUT "ANY MESSAGE";MS$
```

by

```
INPUT "ANY MESSAGE",MS$
```

DEF FN

On the BBC Micro you must define functions at the end of the program after the word END or STOP, not at the beginning as in the example given. In this case, as in many others BBC BASIC is more powerful than standard BASIC, so consult your manual for more information.

STEP 0

On the Oric-1 in the two program fragments that demonstrate a good loop structure, replace:

```
IF INKEY$ = "    " THEN LET X = 2
```

by

```
IF KEY$ = "    " THEN LET X = 1
```

On the Dragon-32 replace it by:

```
IF INKEY$ = "    " THEN LET X = 1
```

On the Vic-20 and the Commodore 64 replace:

```
IF INKEY$ = "    " THEN LET X = 2
```

by

```
GET A$:IF A$ = '    " THEN LET X = 1
```