```
MAIN PROGRAM
(DATA ENTRY)

MENU
(SELECT SUBROUTINES)

END
```

A little further refinement may show that we will need subroutines to calculate totals for months or for categories (MONTHTOTAL and CATTOTAL), average monthly expenditure (MONTHAV) and average yearly expenditure by category (CATAV). The reason for using one-word names for these subroutines is to help us to plan the program without having to worry about details such as line numbers at this stage. On reflection we may decide that even the main menu selection part of the program should be dealt with as a subroutine in order to keep the main part of the program as a separate module. The next stage of refinement of the program will look like this:

```
MAIN PROGRAM (DATA ENTRY)
   MENU (CALL SUBROUTINE)
END

**SUBROUTINES**

1 MENU
2 TOTALS
3 AVERAGES

(2) TOTALS
4 MONTHTOTAL
5 CATTOTAL

(3) AVERAGES
6 MONTHAV
7 CATAV
```

This sketch of the program shows that the MENU subroutine will give us a choice of either TOTALS or AVERAGES. Both of these will themselves be subroutines. The TOTALS subroutine will give a further choice of MONTHTOTAL or CATTOTAL. These will be the subroutines that perform the actual calculations.

The AVERAGES subroutine will give a choice of MONTHAV or CATAV, and again these will be subroutines to perform the appropriate calculations. At this stage it should be possible to see whether our 'program' will do what we want, without doing any actual coding (detailed program writing in BASIC). If we can be satisfied that 'so far so good', we are ready to tackle the writing of the modules (subroutines) themselves. The only change needed to the main program will be a subroutine call before the END statement, so we could add:

```
145 GOSUB **MENU**
```

Note that we are still using 'names' for subroutines rather than line numbers. Many languages, PASCAL, for example, allow sub-programs to be called by name, but most versions of BASIC do not and actual line numbers are needed instead. However, these 'details' can be incorporated later.

Let's see how the MENU subroutine could be written (line numbers have been omitted and you can add appropriate ones if you wish to implement this program).

```
REM THE **MENU** SUBROUTINE
PRINT "WOULD YOU LIKE T(OTALS) OR
   A(VERAGES)?"
PRINT "TYPE EITHER A OR T"
INPUT L$
IF L$ = "T" THEN GOSUB *TOTALS*
IF L$ = "A" THEN GOSUB *AVERAGES*
RETURN
```

Note: we are marking the subroutines called by enclosing them within *——* marks. You will have to use line numbers instead. These can be inserted when you are in a position to know what they are.

Suppose you type T for TOTALS. The program will then call the TOTALS subroutine. This will then present another menu and could look like this:

```
REM THE **TOTALS** SUBROUTINE
PRINT "WOULD YOU LIKE TOTALS FOR"
PRINT "M(ONTH) OR C(ATEGORY)?"
PRINT "TYPE EITHER M OR C"
INPUT L$
IF L$ = "M" THEN GOSUB *MONTHTOTAL*
IF L$ = "C" THEN GOSUB *CATTOTAL*
RETURN
```

Suppose you selected M for MONTHTOTAL. Let's see how we could write a module to calculate the total expenditure for any month in the year.

```
REM THE **MONTHTOTAL** SUBROUTINE
REM THIS CALCULATES TOTAL EXPENDITURE FOR
REM ANY MONTH
PRINT "SELECT MONTH"
PRINT "1-JAN 2-FEB 3-MAR 4-APR 5-MAY"
PRINT "6-JUN 7-JUL 8-AUG 9-SEP"
PRINT "10-OCT 11-NOV 12-DEC"
PRINT "TYPE A NUMBER FOR THE MONTH"
LET T = 0
INPUT M
FOR C = 1 TO 5
LET T = T + A(M,C)
NEXT C
PRINT "THE TOTAL EXPENDITURE FOR MONTH"
PRINT "NUMBER ";M;" IS ";T
RETURN
```

The number representing the month is typed in and the INPUT statement assigns the number to the variable M (MONTH). M is used to specify the 'row' subscript of the two-dimensional array A. The FOR-NEXT loop increments the value of C (column) from one to five so the first time through the loop, if we had selected three for March, the LET statement would be equivalent to LET T = T + A(3,1). The next time round it would be equivalent to LET T = T + A(3,2) and so on.

This week we'll leave you to write the other subroutines, or try out the other exercises. Two-dimensional arrays are ideal for any program that involves tables of data, be they statistical, financial or any other quantity.