Example 7.1

Write a program using a single-subscripted variable to evaluate the following expressions:

$$\text{Total} = \sum_{i=1}^{10} x_i^2$$

The values of x1,x2,....are read from the terminal.

Program in Fig.7.1 uses a one-dimensional array **x** to read the values and compute the sum of their squares.

PROGRAM SHOWING **ONE-DIMENSIONAL ARRAY**

Program :

```
main()
   {
       int    i ;
       float  x[10], value, total ;

 /* . . . . . .READING VALUES INTO ARRAY . . . . . . */

       printf("ENTER 10 REAL NUMBERS\n") ;

       for( i = 0 ; i < 10 ; i++ )
       {
          scanf("%f", &value) ;
          x[i] = value ;
       }
 /* . . . . . .  .COMPUTATION OF TOTAL . . . . . .*/

       total = 0.0 ;
       for( i = 0 ; i < 10 ; i++ )
          total = total + x[i] * x[i] ;

 /*. . .  . PRINTING OF x[i] VALUES AND TOTAL . . . */

       printf("\n");
       for( i = 0 ; i < 10 ; i++ )
          printf("x[%2d] = %5.2f\n", i+1, x[i]) ;

       printf("\ntotal = %.2f\n", total) ;
   }
```

Output

```
ENTER 10 REAL NUMBERS
1.1  2.2  3.3  4.4  5.5  6.6  7.7  8.8  9.9  10.10
```

```
x[ 1] =  1.10
x[ 2] =  2.20
x[ 3] =  3.30
x[ 4] =  4.40
x[ 5] =  5.50
x[ 6] =  6.60
x[ 7] =  7.70
x[ 8] =  8.80
x[ 9] =  9.90
x[10] = 10.10

Total = 446.86
```

Fig.7.1 *Program to illustrate* ***one-dimensional array***

Example 7.2

Given below is the list of marks obtained by a class of 50 students in an annual examination.

43 65 51 27 79 11 56 61 82 09 25 36 07 49 55 63 74 81 49 37
40 49 16 75 87 91 33 24 58 78 65 56 76 67 45 54 36 63 12 21
73 49 51 19 39 49 68 93 85 59

Write a program to count the number of students belonging to each of following groups of marks: 0-9, 10-19, 20-29,.....,100.

The program coded in Fig.7.2 uses the array **group** containing 11 elements, one for each range of marks. Each element counts those values falling within the range of values it represents.

For any value, we can determine the correct group element by dividing the value by 10. For example, consider the value 59. The integer division of 59 by 10 yields 5. This is the element into which 59 is counted.

PROGRAM FOR **FREQUENCY COUNTING**

Program

```
#define   MAXVAL    50
#define   COUNTER   11

main()
{
    float        value[MAXVAL];
    int          i, low, high;
    int   group[COUNTER] = {0,0,0,0,0,0,0,0,0,0,0};

    /* . . . . . .READING AND COUNTING . . . .  . .*/
    for( i = 0 ; i < MAXVAL ; i++ )
    {
     /*. . . . . .READING OF VALUES . . . . . . . . */
       scanf("%f", &value[i]) ;

     /*. . . . . .COUNTING FREQUENCY OF GROUPS. . . . . */
       ++ group[ (int) ( value[i] + 0.5 ) / 10] ;
```

```
    }
     /* . . . .PRINTING OF FREQUENCY TABLE . . . . . .*/
    printf("\n");
    printf(" GROUP     RANGE     FREQUENCY\n\n") ;
    for( i = 0 ; i < COUNTER ; i++ )
    {
       low  = i * 10 ;
       if(i == 10)
         high = 100 ;
       else
         high = low + 9 ;
       printf("  %2d    %3d to %3d       %d\n",
              i+1, low, high, group[i] ) ;
    }
}
```

Output

```
43 65 51 27 79 11 56 61 82 09 25 36 07 49 55 63 74
81 49 37 40 49 16 75 87 91 33 24 58 78 65 56 76 67
45 54 36 63 12 21 73 49 51 19 39 49 68 93 85 59
```

(Input data)

```
GROUP          RANGE            FREQUENCY
  1          0   to   9            2
  2          10  to  19            4
  3          20  to  29            4
  4          30  to  39            5
  5          40  to  49            8
  6          50  to  59            8
  7          60  to  69            7
  8          70  to  79            6
  9          80  to  89            4
 10          90  to  99            2
 11          100 to 100            0
```

Fig.7.2 *Program for* ***frequency counting***

Example 7.3

Write a program using a two-dimensional array to compute and print the following information from the table of data discussed above:

(a) Total value of sales by each girl.
(b) Total value of each item sold.
(c) Grand total of sales of all items by all girls.

The program and its output are shown in Fig.7.4. The program uses the variable **value** in two-dimensions with the index i representing girls and j representing items. The following equations are used in computing the results:

(a) Total sales by m^{th} girl = (girl_total[m]) $\sum_{j=0}^{2}$ value [m][j]

(b) Total value of n^{th} item = (item_total[n]) $\sum_{i=0}^{3}$ value [i][n]

(c) Grand total $= \sum_{i=0}^{3} \sum_{j=0}^{2}$ value[i][j]

$= \sum_{i=0}^{3}$ girl_total[i]

$= \sum_{j=0}^{2}$ item_total[j]

PROGRAM SHOWING TWO-DIMENSIONAL ARRAYS

Program:

```
#define     MAXGIRLS     4
#define     MAXITEMS     3

main()
{
    int   value[MAXGIRLS][MAXITEMS];
    int   girl_total[MAXGIRLS] , item_total[MAXITEMS];
    int   i, j, grand_total;

/*.......READING OF VALUES AND COMPUTING girl_total ...*/

    printf("Input data\n");
    printf("Enter values, one at a time, row-wise\n\n");

    for( i = 0 ; i < MAXGIRLS ; i++ )
    {
       girl_total[i] = 0;
       for( j = 0 ; j < MAXITEMS ; j++ )
       {
          scanf("%d", &value[i][j]);
          girl_total[i] = girl_total[i] + value[i][j];
       }
    }
 /*.......COMPUTING item_total.........................*/

    for( j = 0 ; j < MAXITEMS ; j++ )
```

```
        {
           item_total[j] = 0;
           for( i =0 ; i < MAXGIRLS ; i++ )
              item_total[j] = item_total[j] + value[i][j];
        }


/*.......COMPUTING grand_total.....................*/

        grand_total = 0;
        for( i =0 ; i < MAXGIRLS ; i++ )
           grand_total = grand_total + girl_total[i];

/* .......PRINTING OF RESULTS.....................*/

        printf("\n GIRLS TOTALS\n\n");
        for( i = 0 ; i < MAXGIRLS ; i++ )
           printf("Salesgirl[%d] = %d\n", i+1, girl_total[i] );
        printf("\n ITEM  TOTALS\n\n");
        for( j = 0 ; j < MAXITEMS ; j++ )
           printf("Item[%d] = %d\n", j+1 , item_total[j] );
        printf("\nGrand Total = %d\n", grand_total);
    }
```

Output

```
Input data
Enter values, one at a time, row_wise

310  257  365
210  190  325
405  235  240
260  300  380

GIRLS TOTALS

Salesgirl[1] = 950
Salesgirl[2] = 725
Salesgirl[3] = 880
Salesgirl[4] = 940

 ITEM  TOTALS

Item[1] = 1185
Item[2] = 1000
Item[3] = 1310

Grand Total = 3495
```

Fig.7.4 *Illustration of two-dimensional arrays.*

Example 7.4
Write a program to compute and print a multiplication table for numbers 1 to 5 as shown below:

	1	2	3	4	5
1	1	2	3	4	5
2	2	4	6	8	10
3	3	6	.	.	.
4	4	8	.	.	.
5	5	10	.	.	25

The program shown in Fig.7.5 uses a two-dimensional array to store the table values. Each value is calculated using the control variables of the nested for loops as follows:

product(i,j) = row * column

where i denotes rows and j denotes columns of the product table. Since the indices i and j ranges from 0 to 4, we have introduced the following transformation:

row = i+1
column = j+1

PROGRAM TO PRINT MULTIPLICATION TABLE

Program:

```
#define    ROWS     5
#define    COLUMNS  5

main()
{
    int   row, column, product[ROWS][COLUMNS] ;
    int   i, j ;

    printf("   MULTIPLICATION TABLE\n\n") ;
    printf("    ") ;

    for( j = 1 ; j <= COLUMNS ; j++ )
      printf("%4d" , j ) ;
    printf("\n") ;
    printf("------------------------\n");

    for( i = 0 ; i < ROWS ; i++ )
    {
        row = i + 1 ;
        printf("%2d |", row) ;

        for( j = 1 ; j <= COLUMNS ; j++ )
        {
           column = j ;
           product[i][j] = row * column ;
           printf("%4d", product[i][j] ) ;
        }
        printf("\n") ;
    }
}
```

Output

```
        MULTIPLICATION TABLE
           1    2    3    4    5
     ------------------------------
1    |     1    2    3    4    5
2    |     2    4    6    8   10
3    |     3    6    9   12   15
4    |     4    8   12   16   20
5    |     5   10   15   20   25
```

Fig.7.5 *Program to print multiplication table using two-dimensional array*

Example 7.5

A survey to know the popularity of four cars (Ambassador, Fiat, Dolphin and Maruti) was conducted in four cities (Bombay, Calcutta, Delhi and Madras). Each person surveyed was asked to give his city and the type of car he was using. The results, in coded form, are tabulated as follows:

```
M 1   C 2   B 1   D 3   M 2   B 4
C     1   D 3   M 4   B 2   D 1   C     3
D     4   D 4   M 1   M 1   B 3   B     3
C     1   C 1   C 2   M 4   M 4   C     2
D     1   C 2   B 3   M 1   B 1   C     2
D     3   M 4   C 1   D 2   M 3   B     4
```

Codes represent the following information:

M - Madras 1 - Ambassador
D – Delhi 2 - Fiat
C – Calcutta 3 - Dolphin
B – Bombay 4 - Maruti

Write a program to produce a table showing popularity of various cars in four cities.

A two-dimensional array **frequency** is used as an accumulator to store the number of cars used, under various categories in each city. For example, the element **frequency** [i][j] denotes the number of cars of type j used in city i. The **frequency** is declared as an array of size 5x5 and all the elements are initialized to zero.

The program shown in fig.7.6 reads the city code and the car code, one set after another, from the terminal. Tabulation ends when the letter X is read in place of a city code.

PROGRAM TO TABULATE SURVEY DATA

Program

```
main()
{
    int  i, j, car;
    int  frequency[5][5] = { {0},{0},{0},{0},{0} };
```

```
        char city;

        printf("For each person, enter the city code \n");
        printf("followed by the car code.\n");
        printf("Enter the letter X to indicate end.\n");

    /*. . . . . TABULATION BEGINS  . . . . . */

        for( i = 1 ; i < 100 ; i++ )
        {
           scanf("%c", &city );
           if( city == 'X' )
             break;

           scanf("%d", &car );

           switch(city)
           {

                 case 'B' :  frequency[1][car]++;
                             break;
                 case 'C' :  frequency[2][car]++;
                             break;
                 case 'D' :  frequency[3][car]++;
                             break;
                 case 'M' :  frequency[4][car]++;
                             break;
           }
        }
/*. . . . .TABULATION COMPLETED AND PRINTING BEGINS. . . .*/

        printf("\n\n");
        printf("              POPULARITY  TABLE\n\n");
        printf("-----------------------------------------\n");
        printf("City     Ambassador  Fiat  Dolphin  Maruti \n");
        printf("-----------------------------------------\n");
        for( i = 1 ; i <= 4 ; i++ )
        {
```

```
        switch(i)
        {
              case 1 :  printf("Bombay    ") ;
                        break ;
              case 2 :  printf("Calcutta  ") ;
                        break ;
              case 3 :  printf("Delhi     ") ;
                        break ;
              case 4 :  printf("Madras    ") ;
                        break ;
        }
        for( j = 1 ; j <= 4 ; j++ )
           printf("%7d", frequency[i][j] ) ;
        printf("\n") ;
     }
     printf("-----------------------------------------\n");
  /*. . . . . . . . . PRINTING ENDS. . . . . . . . . . .*/
}
```

Output

```
For each person, enter the city code
followed by the car code.
Enter the letter X to indicate end.

M 1 C 2 B 1 D 3 M 2 B 4
C 1 D 3 M 4 B 2 D 1 C 3
D 4 D 4 M 1 M 1 B 3 B 3
C 1 C 1 C 2 M 4 M 4 C 2
D 1 C 2 B 3 M 1 B 1 C 2
D 3 M 4 C 1 D 2 M 3 B 4  X
```

POPULARITY TABLE

City	Ambassador	Fiat	Dolphin	Maruti
Bombay	2	1	3	2
Calcutta	4	5	1	0
Delhi	2	1	3	2
Madras	4	1	1	4

Fig.7.6 Program to tabulate a survey data