A two-dimensional array is used to store a table of values. Two loops (using for loops) are used to access each value in the table, first loop acts as a row selector and second loop acts as a column selector in the table.

4.4 Declare, Initialize Array of Char Type

Declaration of Array of char type: A string variable is any valid C variable name and is always declared as an array. The syntax of declaration of a string variable is:

char string-name[size];

The size determines the number of character in the string name.

Example: An array of char type to store the above string is to be declared as follows:

char str[8];

str[0]	str[1]	str[2]	str[3]	str[4]	str[5]	str[6]	str[7]
P	r	o	g	r	a	m	\0

An array of char is also called as a string variable, since it can store a string and it permits us to change its contents. In contrast, a sequence of characters enclosed within a pair of double quotes is called a string constant.

Example: "Program" is a string constant.

Initialization of Arrays of char Type

The syntax of initializing a string variable has two variations:

Variation 1

char str1 [6] = { 'H', 'e', 'l', 'l', 'o', '\0'};

Here, str1 is declared to be a string variable with size six. It can store maximum six characters. The initializer – list consists of comma separated character constants. Note that the null character '\0' is clearly listed. This is required in this variation.

Variation 2

char str2 [6] = { "Hello" };

Here, str2 is also declared to be a string variable of size six. It can store maximum six characters including null character. The initializer-list consists of a

string constant. In this variation, null character '\0' will be automatically added to the end of string by the compiler.

In either of these variations, the size of the character array can be skipped, in which case, the size and the number of characters in the initializer-list would be automatically supplied by the compiler.

Example

char str1 [ ] = { "Hello" };

The size of str1 would be six, five characters plus one for the null character '\0'.

char str2 [ ] = { 'H', 'e', 'l', 'l', 'o', '\0'};

The size of str2 would be six, five characters plus one for null character '\0'.

Example 1

Program to sort a list of numbers.

```
#include<stdio.h>
#include<conio.h>
void main( )
{
        int x [6], n, i, j, tmp;
        clrscr( );
        printf ("Enter the no. of elements \n");
        scanf ("%d", & n);
        printf ("Enter %d numbers \n", n);
        for (i=0; i<n; i++)
        scanf("%d", & x [i]);
        /* sorting begins */
        for (i=0; i<n; i++)
        for (j=i + 1; j<n; j++)
        if (x[i]>x[j])
```

```
        {
                tmp = x [i];
                x [i] = x [j];
                x [j] = tmp;
        }
        /* sorting ends */
        printf ("sorted list \n");
        for (i=0; i<n; i++)
        printf ("%d", x [i]);
        getch( );
}
```

Input – Output

```
Enter the no. of elements
5
Enter 5 numbers
10      30      20      50      40
Sorted list
10 20      30      40      50
```

Example 2 :

C program for addition of two matrices.

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
void main( )
{
        int x [5] [5], y [5] [5], z [5] [5], m, n, p, q, i, j;
```

```
clrscr ( );
printf ("Enter number of rows and columns of matrix x \n");
scanf ("%d %d", &m, &n);
printf ("Enter number of rows and columns of matrix y \n");
scanf ("%d %d", &p, &q);
if ((m ! =P) | | (n!=q))
{
        printf ("Matrices not compatible for addition \n");
        exit(1);
}
printf ("Enter the elements of x \n");
for (i=0; i<m; i++)
for (j=0; j<n; j++)
scanf ("%d", &x [i][j]);
printf ("Enter the elements of x \n");
for (i=0; i<p; i++)
for (j=0; j<n; j++)
scanf ("%d", &y [i] [j]);
/* Summation begins */
for (i=0; j<m; i++)
for (j=0; j<n; j++)
z[i] [j] = x [i] [j] + y [i] [j];
/* summation ends */
printf ("Sum matrix z \n);
for (i=0; i<m; i++)
{
        for (j=0; j<n; j++)
```

```
            printf("%d", z[i] [j]);
            printf("\n");
        }
        getch( );
}
```

Example3

Write a program for multiplication of two matrices.

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
void main( )
{
        int x [5] [5], y [5] [5], z [5] [5], m, n, p, q, I, j, k;
        clrscr ( );
        printf ("Enter number of rows and columns of matrix x \n");
        scanf ("%d %d", &m, &n);
        printf ("Enter number of rows and columns of matrix y \n");
        scanf (%d %d", &p, &q);
        if (n!=p)
        {
            printf ("Matrices not compatible for multiplication \n");
            exit (1);
        }
        printf ("Enter the elements of x \n");
        for (i=0; i<m; i++)
        for (j=0; j<n; j++)
        scanf ("%d", &x [i] [j]);
```

```
        printf("Enter the elements of y \n");
        for (i=0; i<p; i++)
        for (j=0; j<q; j++)
        scanf("%d", &x [i] [j]);
        printf("Enter the elements of y \n");
        for (i=0; i<p; i++)
        for (j=0; j<q; j++)
        scanf("%d", &y [i] [j]);
        /* Multiplication of matrices of x 7 y ends */
        for (i=0; i<m; i++)
        for (j=0; j<q; j++)
        { z [i] [j] = 0;
        for (k=0; k<n; k++)
        z [i] [j] + = x [i] [k] * y [k] [j];
        }
        /* Multiplication of matrices of x & y ends */
        printf("Product Matrix z \n");
        for (i=0; i<m; i++)
        {
                for (j=0; j<q; j++)
                        printf("%d", z[i] [j]);
                printf("\n");
        }
        getch( );
}
```

Example 4: C program to print transpose of a matrix.

[The transpose of a matrix is obtained by switching the rows and columns of matrix].

```
#include<stdio.h>
#include<conio.h>
void main ( )
{
        int a[3] [3], b[3] [3], i, j;
        clrscr ( );
        printf ("Enter the elements of the matrix : \n");
        for (i=0; i<3; i++)
        for (j=0; j<3; j++)
        scanf ("%d", &a[i] [j]);
        printf ("given matrix is : \n");
        for (i=0; i<3; i++)
        {
                for (j=0; j<3; j++)
                        printf ("%d", a[i] [j]);
                        printf ("\n");
        }
        printf ("transpose of given matrix is : \n");
        for (i=0; i<3; i++)
        {
                for (j=0; j<3; j++)
                {
                        b [i] [j] = a [j] [i];
                        printf ("%d", b[i] [j]);
```

```
                printf("\n");
            }
        }
        getch( );
    }
```

Example 5

Write a 'C' program to find the average marks of 'n' students of five subjects for each subject using arrays.

Ans.

```
#include <stdio.h>
void main( )
{
    int Sno, S1,S2,S3;
    float tot, avg;
    char sname[10];
    printf("Enter student no;");
    scanf("%d", & Sname);
    printf("Enter subject - 1, sub - 2, sub - 3 marks;");
    scanf("%d %d %d", &s1,&s2,&s3);
    tot = S1+S2+S3;
    avg = tot/3;
        printf("total = %f", tot);
        printf("Average = %f", avg);
}
```

Example 6

Write a C program to check the given word is 'Palindrome' or not.

Ans.

```
#include<stdio.h>
```

```
#include<conio.h>
void main ( )
{
        char str[80], rev[80];
        int k, i, j, flag = 0;
        clrscr ( );
        printf ("Enter any string (max. 80 chars) : \n");
        gets (str);
for (i=0; str[i]!= '\0'; i++);
        for (j=i-1; k=0; j>=0; j—, k++)
                rev[k] = str[j];
        rev[k] = '\0';
        for (i=0; str[i]!= '\0'; i++)
        {
                if(str[i]!=rev[i])
                {
                        flag=1;
                        break;
                }
        }
                if (flag = =1);
                        printf ("Given string is not palindrome. \n");
        else
                        printf ("Given string is palindrome. \n");
        getch( );
}
```