Paper Name: Programming Concepts

Topic: Introduction to 2D Arrays in C

Paper Code: BCA GE202

Semester II

By Ms. Manisha Prasad

Head, Assistant Professor,

Department of Computer Science

Patna Women's College

E – mail : manisha_prasad@yahoo.com

INTRODUCTION TO 2D ARRAY IN C

An array of arrays is known as 2D array. The two dimensional (2D) array in C programming is also known as matrix. A matrix can be represented as a table of rows and columns. We can visualize a 2D array as follows:

	col[0]	col [1]	col[2]
row[0]			
row[1]			
row[2]			

int arr[3][3]

Logically we can think row[0] as an array with 3 elements, and the same for row[1] and row[2]. Since these three arrays are brought together in a tabular form, so now the resultant array becomes an example of an array of arrays or simply we can say a two dimensional array. Now here each element of the two dimensional array will be represented by two indices, first the row number and next the column number. For example the first element's address is arr[0][0], second element of row[0] as arr[0][1], third element of row[0] as arr[0][2]. Similarly the first element's address of row[1] is arr[1][0], second element of row[1] as arr[1][1], third element of row[1] as arr[1][2].]. Similarly the first element's address of row[2] is arr[2][0], second element of row[2] as arr[2][1], third element of row[2] as arr[2][2].

Initialization of 2D Array

There are two ways to initialize a two Dimensional arrays during declaration.

```
int arr[3][3] = {

   {1, 2, 3},

   {4,5, 6, },

   {7,8, 9}

};
```

or

```
int arr[3][3] = { 1, 2, 3, 4, 5, 6, 7, 8, 9};
```

Although both the above declarations are valid, it is recommended to use the first method as it is more readable, because you can visualize the rows and columns of 2D array in this method.

Important points to remember while initializing a 2D array

We already know, when we initialize a one dimensional array during declaration, we may or may not specify the size of it. However that’s not the case with 2D array, we must always specify the second dimension even if we are specifying elements during the declaration. Let’s understand this with the help of few examples –

/* Valid declaration*/

int arr[2][2] = {1, 2, 3 ,4 }

/* Valid declaration*/

int arr[][2] = {1, 2, 3 ,4 }

/* Invalid declaration – you must specify second dimension*/

int arr[][] = {1, 2, 3 ,4 }

/* Invalid because of the same reason mentioned above*/

int arr[2][] = {1, 2, 3 ,4 }

How to store user input data into 2D array

We can calculate how many elements a two dimensional array can have by using this formula:

The array arr[n1][n2] can have n1*n2 elements.

The array that we have in the example below is having the dimensions 5 and 4. These dimensions are known as subscripts. So this array has first subscript value as 5 and second subscript value as 4.

So the array arr[5][4] can have 5*4 = 20 elements.

To store the elements entered by user we are using two for loops, one of them is a nested loop. The outer loop runs from 0 to the (first subscript) and the inner for loops runs from 0 to the (second subscript). This way the order in which user enters the elements would be arr[0][0], arr[0][1], arr[0][2]…so on.

A program to accept elements in a two dimensional array and display them.

```
#include <stdio.h>
#include <conio.h>
int main()
{
 /* 2D array declaration*/
 int arr[3][3];
/*Counter variables for the loop*/
 int i, j;
/*Loop to accept values in 2D array*/
  for(i=0; i<3; i++)
   {
     for(j=0;j<3;j++)
      {
        printf("Enter value for arr[%d][%d]:", i, j);
        scanf("%d", &arr[i][j]);
      }
  }

/*Loop to display the values in 2D array*/
for(i=0; i<3; i++)
{
      printf("\n”);
        for(j=0;j<3;j++)
        {
              printf("\t %d",  arr[i][j]);
           }
   }
  getch();
}
```