Multi-dimensional Arrays in C

C programming language allows multidimensional arrays. Here is the general form of a multidimensional array declaration –

type name[size1][size2]...[sizeN];

For example, the following declaration creates a three dimensional integer array –

int threedim[5][10][4];

Two-dimensional Arrays

The simplest form of multidimensional array is the two-dimensional array. The two-dimensional array can be defined as an array of arrays. The 2D array is organized as matrices which can be represented as the collection of rows and columns.

Declaration of two dimensional Array in C

data_type array_name[rows][columns];

Consider the following example.

int a[3][4];

A two-dimensional array **a**, which contains three rows and four columns can be shown as follows –

	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

Thus, every element in the array **a** is identified by an element name of the form **a[i][j]**, where 'a' is the name of the array, and 'i' and 'j' are the subscripts that uniquely identify each element in 'a'.

Initializing Two-Dimensional Arrays

Multidimensional arrays may be initialized by specifying bracketed values for each row. Following is an array with 3 rows and each row has 4 columns.

```
int a[3][4] = {
   {0, 1, 2, 3} ,   /*  initializers for row indexed by 0 */
   {4, 5, 6, 7} ,   /*  initializers for row indexed by 1 */
```

```
   {8, 9, 10, 11}   /*  initializers for row indexed by 2 */
};
```

We can also write like this

```
int a[3][4] = {{0, 1, 2, 3},{4, 5, 6, 7},{8, 9, 10, 11}};
int a[3][4] = {0,1,2,3,4,5,6,7,8,9,10,11};
```

Two-dimensional array example in C

```
#include<stdio.h>
int main(){
int i=0,j=0;
int arr[4][3]={{1,2,3},{2,3,4},{3,4,5},{4,5,6}};
//traversing 2D array
for(i=0;i<4;i++){
  for(j=0;j<3;j++){
    printf("arr[%d] [%d] = %d \n",i,j,arr[i][j]);
  }//end of j
 }//end of i
 return 0;
 }
```

Output

```
arr[0][0] = 1
arr[0][1] = 2
arr[0][2] = 3
arr[1][0] = 2
arr[1][1] = 3
arr[1][2] = 4
arr[2][0] = 3
arr[2][1] = 4
arr[2][2] = 5
arr[3][0] = 4
arr[3][1] = 5
arr[3][2] = 6
```