Programming in C

(2–D Array)

Prepared By

Alok Haldar

Assistant professor

Department of Computer Science & BCA

Kharagpur College

Two Dimensional Array in C

An array of arrays is known as 2D array.

Declaration of two dimensional Array in C

The syntax to declare the 2D array is given below.

1. data_type array_name[rows][columns];

Consider the following example.
Int twodimen[4][3];

Here, 4 is the number of rows, and 3 is the number of columns.

Initialization of 2D Array in C

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

1. int arr[4][3]={{1,2,3},{2,3,4},{3,4,5},{4,5,6}};

   or

   ```
   int arr[2][4] = {
                        {10, 11, 12, 13},
                        {14, 15, 16, 17}
                   };
   ```

2. int arr[2][4] = { 10, 11, 12, 13, 14, 15, 16, 17};

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 ofj
}//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
```

How to store user input data and displaying it into 2D array

```
#include<stdio.h>
int main(){
   /* 2D array declaration*/
   int arr[4][5];
   /*Counter variables for the loop*/
   int i, j;
   for(i=0; i<4; i++) {
      for(j=0;j<5;j++) {
         printf("Enter value for arr[%d][%d]:", i, j);
         scanf("%d", &arr[i][j]);
      }
   }
 printf("\n printing the elements ....\n");
           for(i=0;i<4;i++)
           {
              printf("\n");
              for (j=0;j<5;j++)
              {
                   printf("%d\t",arr[i][j]);
              }
           }

   return 0;
}
```

Output:

```
Enter the value for a[0][0]: 10

Enter the value for a[0][1]: 20

Enter the value for a[0][2]: 30
Enter the value for a[0][3]: 40
Enter the value for a[0][4]: 50

Enter the value for a[1][0]: 60
Enter the value for a[1][1]: 70
Enter the value for a[1][2]: 80
Enter the value for a[1][3]: 90
Enter the value for a[1][4]: 100
```

Enter the value for a[2][0]: 110
Enter the value for a[2][1]: 120
Enter the value for a[2][2]: 130
Enter the value for a[2][3]: 140
Enter the value for a[1][4]: 150

Enter the value for a[2][0]: 160
Enter the value for a[2][1]: 170
Enter the value for a[2][2]: 180
Enter the value for a[2][3]: 190
Enter the value for a[1][4]: 200

printing the elements ....

```
10   20   30   40   50
60   70   80   90   100
110  120  130  140  150
160  170  180  190  200
```

Conceptually you can visualize the above array like this:

However the actual representation of this array in memory would be something like this: