2D Array

Two Dimensional Array in C

- 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.
- However, 2D arrays are created to implement a relational database lookalike data structure.
- It provides ease of holding the bulk of data at once which can be passed to any number of functions wherever required.

- **Declaration of two dimensional Array in C**
- The syntax to declare the 2D array is given below.
- 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**
- In the 1D array, we don't need to specify the size of the array if the declaration and initialization are being done simultaneously.
- However, this will not work with 2D arrays. We will have to define at least the second dimension of the array.
- The two-dimensional array can be declared and defined in the following way.
- int arr[4][3]={{1,2,3},{2,3,4},{3,4,5},{4,5,6}};

- **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
```

- C 2D array example: Storing elements in a matrix and printing it.

```
#include <stdio.h>
void main ()
{
    int arr[3][3],i,j;
    for (i=0;i<3;i++)
    {
        for (j=0;j<3;j++)
```

```
{
            printf("Enter a[%d][%d]: ",i,j);
            scanf("%d",&arr[i][j]);
        }
    }
printf("\n printing the elements ....\n");
    for(i=0;i<3;i++)
    {
        printf("\n");
```

```
for (j=0;j<3;j++)
        {
            printf("%d\t",arr[i][j]);
        }
    }
}
```

Output

```
Enter a[0][0]: 56
Enter a[0][1]: 10
Enter a[0][2]: 30
Enter a[1][0]: 34
Enter a[1][1]: 21
Enter a[1][2]: 34
Enter a[2][0]: 45
Enter a[2][1]: 56
Enter a[2][2]: 78
printing the elements ....
56      10      30
34      21      34
45      56      78
```

- **Return an Array in C**
- **What is an Array?**
- An array is a type of data structure that stores a fixed-size of a homogeneous collection of data.
- In short, we can say that array is a collection of variables of the same type.

- For example, if we want to declare 'n' number of variables, n1, n2...n., if we create all these variables individually, then it becomes a very tedious task.
- In such a case, we create an array of variables having the same type. Each element of an array can be accessed using an index of the element.
- Let's first see how to pass a single-dimensional array to a function.

- **Passing array to a function**

```
#include <stdio.h>
void getarray(int arr[])
{
    printf("Elements of array are : ");
    for(int i=0;i<5;i++)
    {
        printf("%d ", arr[i]);
    }
}
int main()
{
   int arr[5]={45,67,34,78,90};
   getarray(arr);
   return 0;
}
```

- In the above program, we have first created the array arr[] and then we pass this array to the function getarray().
- The getarray() function prints all the elements of the array arr[].
- Output
- Elements of array are: 45,67,34,78,90