

## Two Dimensional Arrays

The two-dimensional array can be defined as an array of arrays. At times we need to store the data in form of tables or matrices. For this, we can use the two dimensional arrays. This array is specified by using two subscripts where one subscript is denoted as the row and the other as the column. It is also viewed as an array of arrays.

### Declaration of two dimensional Array

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

**data\_type array\_name[row size][column size];**

#### Example:-

```
float x[3][4]
```

Here, x is a two-dimensional (2d) array. The array can hold 12 elements. You can think the array as a table with 3 rows and each row has 4 columns.

	Column 1	Column 2	Column 3	Column 4
Row 1	x[0][0]	x[0][1]	x[0][2]	x[0][3]
Row 2	x[1][0]	x[1][1]	x[1][2]	x[1][3]
Row 3	x[2][0]	x[2][1]	x[2][2]	x[2][3]

### Initialization of 2D Array in C

1. Using Initializer List
2. Using Loops

#### 1. Using Initializer List

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}};
```

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

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

## 2. Initialization of 2D array using Loops

We can use any C loop to initialize each member of a 2D array one by one as shown in the below example.

### Example:

```
int x[3][4];
for(int i = 0; i < 3; i++)
{
    for(int j = 0; j < 4; j++)
    {
        x[i][j] = i + j;
    }
}
```

## Accessing Elements of Two-Dimensional Arrays in C

Elements in 2D arrays are accessed using row indexes and column indexes. Each element in a 2D array can be referred to by:

### Syntax:

array\_name[i][j]

## 1.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++)
    {
        for (j=0;j<3;j++)
        {
            printf("%d\t",arr[i][j]);
        }
        printf("\n");
    }
}
```

**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

**2. Program to Add Two Matrices**

```
#include <stdio.h>
int main()
{
    int r, c, a[100][100], b[100][100], sum[100][100], i, j;
    printf("Enter the number of rows (between 1 and 100): ");
    scanf("%d", &r);
    printf("Enter the number of columns (between 1 and 100): ");
    scanf("%d", &c);
    printf("\nEnter elements of 1st matrix:\n");
    for (i = 0; i < r; i++)
    {
        for (j = 0; j < c; j++)
        {
            printf("Enter element a%d%d: ", i + 1, j + 1);
            scanf("%d", &a[i][j]);
        }
    }
    printf("Enter elements of 2nd matrix:\n");
    for (i = 0; i < r; i++)
    {
        for (j = 0; j < c; j++)
        {
            printf("Enter element b%d%d: ", i + 1, j + 1);
            scanf("%d", &b[i][j]);
        }
    }
}
```

```

// adding two matrices
for (i = 0; i < r; i++)
{
    for (j = 0; j < c; j++)
    {
        sum[i][j] = a[i][j] + b[i][j];
    }
}
// printing the result
printf("\nSum of two matrices: \n");
for (i = 0; i < r; i++)
{
    for (j = 0; j < c; j++)
    {
        printf("%d ", sum[i][j]);
    }
    Printf("\n");
}
return 0;
}

```

### **Output:-**

Enter the number of rows (between 1 and 100): 2  
 Enter the number of columns (between 1 and 100): 2  
 Enter elements of 1st matrix:  
 Enter element a11: 12  
 Enter element a12: 22  
 Enter element a21: 33  
 Enter element a22: 44

Enter elements of 2nd matrix:  
 Enter element b11: 32  
 Enter element b12: 3  
 Enter element b21: 4  
 Enter element b22: 55

Sum of two matrices:  
 44 25  
 37 99

### 3. Matrix Multiplication

```
#include<stdio.h>
int main()
{
    int r1,r2,c1,c2;
    int m1[r1][c1],m2[r2][c2],mul[r1][c2];
    printf("Enter number of rows for First Matrix:\n");
    scanf("%d",&r1);
    printf("Enter number of columns for First Matrix:\n");
    scanf("%d",&c1);
    printf("Enter number of rows for Second Matrix:\n");
    scanf("%d",&r2);
    printf("Enter number of columns for Second Matrix:\n");
    scanf("%d",&c2);
    if(c1!=r2)
    {
        printf("Matrices Can't be multiplied together");
    }
    else
    {
        printf("Enter first matrix elements \n");
        for(int i=0;i<r1;i++)
        {
            for(int j=0;j<c1;j++)
            {
                scanf("%d",&m1[i][j]);
            }
        }
        printf("Enter Second matrix elements\n");
        for(int i=0;i<r2;i++)
        {
            for(int j=0;j<c2;j++)
            {
                scanf("%d",&m2[i][j]);
            }
        }
        //Multiplying two matrices
        for(int i=0;i<r1;i++)
        {
            for(int j=0;j<c2;j++)
            {
                mul[i][j]=0;
                // Multiplying i`th row with j`th column
                for(int k=0;k<c1;k++)
```

```

        {
            mul[i][j]+=m1[i][k]*m2[k][j];
        }
    }
}
printf("Multiplied matrix\n");
for(int i=0;i<r1;i++)
{
    for(int j=0;j<c2;j++)
    {
        printf("%d\t",mul[i][j]);
    }
    printf("\n");
}
}
return 0;
}

```

### Output:-

Enter number of rows for First Matrix:2

Enter number of columns for First Matrix:2

Enter number of rows for Second Matrix:2

Enter number of columns for Second Matrix:2

Enter first matrix elements

1

23

3

4

Enter Second matrix elements

4

3

6

7

Multiplied matrix

142 164

36 37

#### 4. Program to Find the Transpose of a Matrix

```
#include <stdio.h>
int main()
{
    int a[10][10], transpose[10][10], r, c;
    printf("Enter rows and columns: ");
    scanf("%d %d", &r, &c);
    // assigning elements to the matrix
    printf("\nEnter matrix elements:\n");
    for (int i = 0; i < r; i++)
    {
        for (int j = 0; j < c; j++)
        {
            printf("Enter element a%d%d: ", i + 1, j + 1);
            scanf("%d", &a[i][j]);
        }
    }
    // printing the matrix a[][]
    printf("\nEnter matrix: \n");
    for (int i = 0; i < r; i++)
    {
        for (int j = 0; j < c; j++)
        {
            printf("%d ", a[i][j]);
        }
        printf("\n");
    }
    // computing the transpose
    for (int i = 0; i < r; i++)
    {
        for (int j = 0; j < c; j++)
        {
            transpose[j][i] = a[i][j];
        }
    }
    // printing the transpose
    printf("\nTranspose of the matrix:\n");
    for (int i = 0; i < c; i++)
    {
        for (int j = 0; j < r; j++)
        {
            printf("%d ", transpose[i][j]);
        }
    }
    printf("\n");
}
```

```
}  
return 0;  
}
```

**Output:-**

Enter rows and columns: 2 3

Enter matrix elements:

Enter element a11: 1

Enter element a12: 2

Enter element a13: 3

Enter element a21: 4

Enter element a22: 5

Enter element a23: 6

Entered matrix:

1 2 3

4 5 6

Transpose of the matrix:

1 4

2 5

3 6