

Two dimensional Arrays

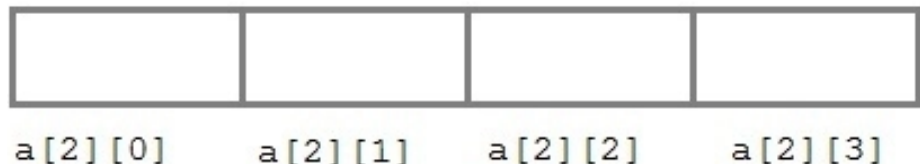
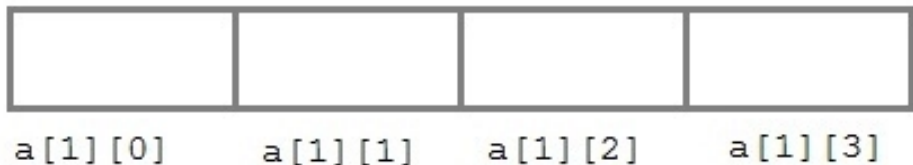
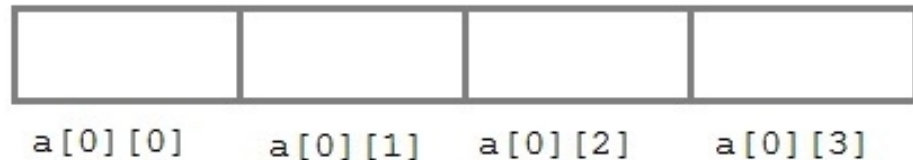
- C language supports **multidimensional arrays** also.
- The simplest form of a multidimensional array is the **two-dimensional array**.
- Both the row's and column's **index begins from 0**.

Two-dimensional arrays are declared as follows,

- **data-type array-name[row-size][column-size];**

/* Example */

- `int a[3][4];`



Two dimensional Arrays

- An array can also be declared and initialized together.

For example,

- `int arr[][3] = { {0,0,0}, {1,1,1} };`

Note:

- ✓ We have **not assigned any row value** to our array in the above example. It means we **can initialize any number of rows**.
- ✓ But, we must **always specify number of columns**, else it will give a compile time error.
- ✓ Here, a **2*3** multi-dimensional matrix is created.

Compile time initialization of a two dimensional Array

```
#include<stdio.h>
```

```
void 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
```

```
}
```

Compile time initialization of a two dimensional Array

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`

Runtime initialization of a two dimensional Array

```
#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]);
```

```
        }
```

```
    }
```

Runtime initialization of a two dimensional Array

```
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]);
```

```
    }
```

```
}
```

```
}
```

Runtime initialization of a two dimensional Array

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

Example: Matrix Addition

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int m, n, i, j;
```

```
    printf("Enter the number of rows and columns of the matrices: ");
```

```
    scanf("%d%d", &m, &n);
```

```
    int a[m][n], b[m][n], c[m][n];
```

```
    printf("Enter the elements of matrix A: \n");
```

```
    for (i = 0; i < m; i++)
```

```
    {
```

```
        for (j = 0; j < n; j++)
```

```
        {
```

```
            scanf("%d", &a[i][j]);
```

```
        }
```

```
    }
```



```
printf("Enter the elements of matrix B: \n");
for (i = 0; i < m; i++)
{
    for (j = 0; j < n; j++)
    {
        scanf("%d", &b[i][j]);
    }
}
// add the matrices
for (i = 0; i < m; i++)
{
    for (j = 0; j < n; j++)
    {
        c[i][j] = a[i][j] + b[i][j];
    }
}
```

```
// print the result
```

```
printf("The sum of the two matrices is: \n");
```

```
for (i = 0; i < m; i++)
```

```
{
```

```
    for (j = 0; j < n; j++)
```

```
    {
```

```
        printf("%d ", c[i][j]);
```

```
    }
```

```
    printf("\n");
```

```
}
```

```
return 0;
```

```
}
```

Output

Enter the number of rows and columns of the matrices: 2 2

Enter the elements of matrix A:

1 2

3 4

Enter the elements of matrix B:

5 6

7 8

The sum of the two matrices is:

6 8

10 12

Example: Matrix Multiplication

In matrix multiplication *first matrix one row element is multiplied by second matrix all column elements.*

2*2 and 3*3 matrices,

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad B = \begin{pmatrix} 5 & 6 & 7 \\ 8 & 9 & 10 \end{pmatrix}$$

Multiplication of two matrixes:

$$A * B = \begin{pmatrix} 1*5 + 2*8 & 1*6 + 2*9 & 1*7 + 2*10 \\ 3*5 + 4*8 & 3*6 + 4*9 & 3*7 + 4*10 \end{pmatrix}$$

$$A * B = \begin{pmatrix} 21 & 24 & 27 \\ 47 & 54 & 61 \end{pmatrix}$$

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
int main()
```

```
{
```

```
    int a[10][10],b[10][10],mul[10][10],r,c,i,j,k;
```

```
    printf("enter the number of row=");
```

```
    scanf("%d",&r);
```

```
    printf("enter the number of column=");
```

```
    scanf("%d",&c);
```

```
    printf("enter the first matrix element=\n");
```

```
    for(i=0;i<r;i++)
```

```
    {
```

```
        for(j=0;j<c;j++)
```

```
        {
```

```
            scanf("%d",&a[i][j]);
```

```
        }
```

```
    }
```

```
printf("enter the second matrix element=\n");  
for(i=0;i<r;i++)  
{  
    for(j=0;j<c;j++)  
    {  
        scanf("%d",&b[i][j]);  
    }  
}
```

```
printf("multiply of the matrix=\n");  
for(i=0;i<r;i++)  
{  
    for(j=0;j<c;j++)  
    {  
        mul[i][j]=0;  
        for(k=0;k<c;k++)  
        {  
            mul[i][j]+=a[i][k]*b[k][j];  
        }  
    }  
}
```

```
//for printing result
```

```
for(i=0;i<r;i++)
```

```
{
```

```
    for(j=0;j<c;j++)
```

```
    {
```

```
        printf("%d\t",mul[i][j]);
```

```
    }
```

```
    printf("\n");
```

```
}
```

```
return 0;
```

```
}
```


Output:

enter the number of row=3

enter the number of column=3

enter the first matrix element=

1 1 1

2 2 2

3 3 3

enter the second matrix element=

1 1 1

2 2 2

3 3 3

multiply of the matrix=

6 6 6

12 12 12

18 18 18