Two Dimensional Arrays

...

By:

S.Sairam Karthik

> Objectives:

- Understand what is an “Array”

- Understand 2-Dimensional Arrays

- Use 2-D Arrays to perform multiplication of two Matrices

> What is an Array?

- An array is a data structure, that stores elements of the same data type, in a **contiguous** block of memory.

- Syntax:

```
int array[4] = {10, 20, 30, 40};
```

An Integer array compared to a Character array in Computer Memory

Integer Array

```
> int arr[4] = {10, 20, 30, 40};
```

Character Array

```
> char arr[4] = {‘A’, ‘B’, ‘C’,’D’};
```

> 2-Dimensional Array

- 2-D Arrays, are Arrays which contain two dimensions, i.e. Rows and Columns.

- We can think of it as an arrangement of the Data in a grid like structure, where we can access the element, by the row and column number.

- In fact, it is pretty similar to a Matrix.

> Syntax:

Compile Time Initialization:

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

OR

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

Runtime Initialization:

```
for(i=0;i<2;i++){
    for(j=0;j<3;j++){
        scanf("%d", &A[i][j]);
    }
}
```

Multiplication Of Matrices

> Step 1: Checking the Order of Matrix

```c
#include <stdio.h>
void main()
{
    int r1, c1;
    int r2, c2;

    printf("Enter the order of the First matrix: ");
    scanf("%d %d", &r1, &c1);
    printf("Enter the order of the Second Matrix: ");
    scanf("%d %d", &r2, &c2);

    if(c1 != r2)
    {
        printf("The two matrices cannot be multiplied!\n");
        return;
    }
```

> Step 2: Initializing The Matrices

```
20      int A[r1][c1];
21      int B[r2][c2];
22      int C[r1][c2];
23      printf("Enter first Matrix: \n");
24      for(int i = 0; i < r1; i++) {
25          for(int j = 0; j < c1; j++) {
26              scanf("%d", &A[i][j]);
27          }
28      }
29      printf("Enter second Matrix: \n");
30      for(int i = 0; i < r2; i++) {
31          for(int j = 0; j < c2; j++) {
32              scanf("%d", &B[i][j]);
33          }
34      }
```

> Step 3: Multiplying the two Matrices

```
37      printf("The Result is: \n");
38      for(int i = 0; i < r1; i++)
39      {
40          for(int j = 0; j < c2; j++ )
41          {
42              int sum = 0;
43              for(int k = 0; k < c1; k++)
44              {
45                  sum = sum + (A[i][k] * B[k][j]);
46              }
47              C[i][j] = sum;
48              printf("%d\t", C[i][j]);
49          }
50          printf("\n");
51      }
52  }
```

> Output

```
Enter the order of the First matrix: 2 3
Enter the order of the Second Matrix: 3 4
Enter first Matrix:
1 2 3 4 5 6
Enter second Matrix:
10 10 10 20 10 10 20 20 10 20 20 20
The Result is:

60      90      110     120
150     210     260     300
```

thank you;

Slides and Source Code:
https://github.com/ss-karthik/matrix-multiplication