

2D Arrays in C++

Assignment Solutions



Q1 - Given a 2D matrix with m rows and n columns containing integers, find and print the maximum value present in the array.

(Easy)

m=3
n=3
arr[] = {{1,2,3}, {4,5,6}, {7,8,9}}
Output: 9

Explanation: We will iterate through all the elements of the matrix using 2 for loops and find the maximum amongst all those elements i.e. mat[i][j].

Here INT_MIN is the minimum value possible for any integer variable to have. Its value is
INT_MIN = -2147483648

```
#include <iostream>
#include <climits>
using namespace std;
int main()
{
    int m,n;
    cin>>m>>n;
    int mat[m][n];
    for(int i=0;i<m;i++)
    {
        for(int j=0;j<n;j++)
        {
            cin>>mat[i][j];
        }
    }
    int val=INT_MIN;
    for(int i=0;i<m;i++)
    {
        for(int j=0;j<n;j++)
        {
            if(mat[i][j]>val)
            {
                val=mat[i][j];
            }
        }
    }
    cout<<val;
    return 0;
}
```

```
3 3
1 2 3
4 5 6
7 8 9
9

...Program finished with exit code 0
Press ENTER to exit console.
```

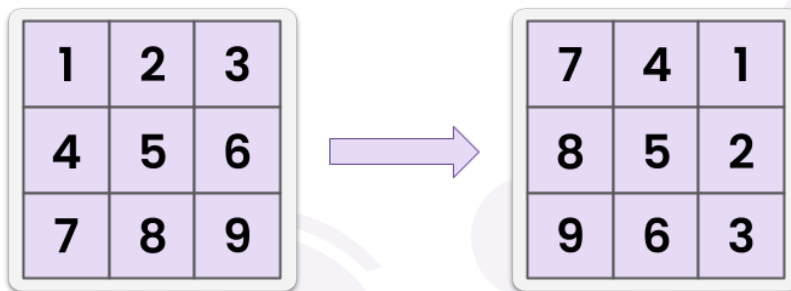
Q2 - You are given a $n \times n$ square matrix, you need to rotate the matrix by 90 degrees in clockwise direction. You need to do it in-place i.e. you are not allowed to make a new matrix and allocate the elements to it. Make the changes in the same matrix and print it.

(Medium)

$n=3$

`arr[] = {{1,2,3}, {4,5,6}, {7,8,9}}`

Output: `{{7,4,1},{8,5,2},{9,6,3}}`



Explanation: In order to rotate a matrix clockwise we first need to take the transpose of the matrix and then swap the diagonal elements i.e. for diagonals we need to swap (i,j) with $(j,size-j-1)$. The transpose and the swapping part has been commented below in the code.

`swap(a,b)` : is an inbuilt function of c++ which is used to swap the values of two variables/elements(a and b).

```
#include <iostream>
#include <vector>
using namespace std;
int main()
{
    int n;
    cin>>n;
    int mat[n][n];
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            cin>>mat[i][j];
        }
    }

    for(int i=0;i<n;i++)
    {
        for(int j=0;j<i;j++)
        {
            swap(mat[i][j],mat[j][i]);
        }
    }
    //swap diagonal elements
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n/2;j++)
        {
            swap(mat[i][j],mat[i][n-j-1]);
        }
    }
    cout<<endl;
    //printing the mat
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            cout<<mat[i][j]<<" ";
        }
        cout<<endl;
    }
}
```

```

3
1 2 3
4 5 6
7 8 9

7 4 1
8 5 2
9 6 3

...Program finished with exit code 0
Press ENTER to exit console.
    
```

Q3 – Given a $m \times n$ integer matrix. If an element of the matrix is 0 then set the complete row and column of that element to 0. Make the changes inplace and print the matrix.

(Hard)

```

m=3
n=3
arr[]={{1,2,3},{1,0,1},{5,6,7}}
Output: {{1,0,3},{0,0,0},{5,0,7}}
arr[]={{0,1,2,0},{3,4,5,2},{1,3,1,5}}
Output:{{0,0,0,0},{0,4,5,0},{0,3,1,0}}
    
```

Explanation: We will first store all the cell positions(i,j) in a vector of pair which has the value 0. Then we will traverse through the vector and make all the rows and columns corresponding to that cell(i,j) as 0.

```
#include <iostream>
#include <climits>
#include <vector>
using namespace std;
int main()
{
    int m,n;
    cin>>m>>n;
    int mat[m][n];
    for(int i=0;i<m;i++)
    {
        for(int j=0;j<n;j++)
        {
            cin>>mat[i][j];
        }
    }
    vector<pair<int,int>> ans; // we are storing all the cells which have value 0
    for(int i=0;i<m;i++)
    {
        for(int j=0;j<n;j++)
        {
            if(mat[i][j]==0)
            {
                ans.push_back({i,j});
            }
        }
    }
    for(int i=0;i<ans.size();i++) //traversing the ans vector and making row and column of
    that
    { //call to 0
        int x=ans[i].first;
        int y=ans[i].second;
        int row=0;
        int col=0;
        while(row<m)
        {
            mat[row][y]=0;
            row++;
        }
        while(col<n)
        {
            mat[x][col]=0;
            col++;
        }
    }
}
```

```
cout<<endl;
//printing the updated matrix
for(int i=0;i<m;i++)
{
    for(int j=0;j<n;j++)
    {
        cout<<mat[i][j]<<" ";
    }
    cout<<endl;
}
}
```

```
3 4
0 1 2 0
3 4 5 2
1 3 1 5

0 0 0 0
0 4 5 0
0 3 1 0

...Program finished with exit code 0
Press ENTER to exit console.
```