2D Array Problems Set 5

Assignment Solutions

SKILLS

1) Given a 2D matrix with m rows and n columns containing integers. Print and return the row number with maximum sum in the array.

Note: In case multiple rows have the same sum then return any row number with that sum.
m=3
n=3
arr[] = {{1,9,6}, {4,5,2}, {7,8,3}}

Output: 3

Explanation: The 3rd row has the maximum sum which is 18

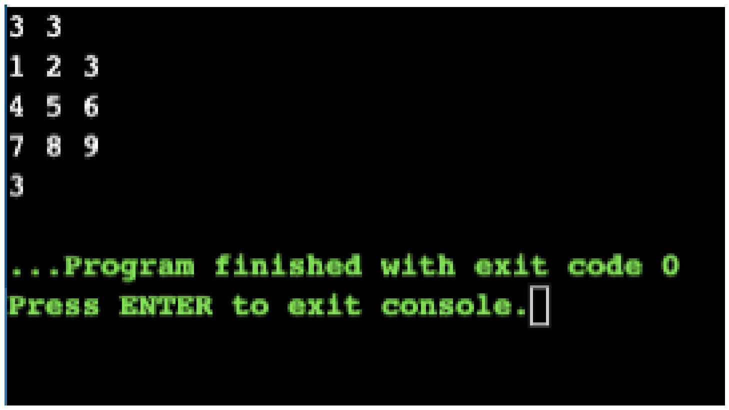
m=3
n=3
arr[] = {{1,2,3}, {1,3,2}, {2,1,3}}

Output: 1

Explanation: All the rows have the same sum i.e. 6 so return any row number.

Explanation: We will find the prefix sum matrix for the given matrix and once the prefix sum of i'th row is calculated, we will compare it with max sum of all the i-1 rows which will be stored in temp variable and then store the row number of that row in a separate variable named 'row' whose sum is initially stored in 'temp'. In the end row + 1 is returned because in arrays by default the indexing is 0 based but we have to return in 1-based format.

```
#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 sum=INT_MIN;
  int row=-1;
  int temp=-1;
  for(int i=0;i<m;i++) //loop to convert the matrix to prefix sum matrix
  {
      for(int j=1;j<n;j++)
      {
          mat[i][j]=mat[i][j]+mat[i][j-1];
      }
      if(mat[i][n-1]>temp)
      {
          temp=mat[i][n-1];
          row=i;
      }
  }
  cout<<row+1;

}
```



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

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

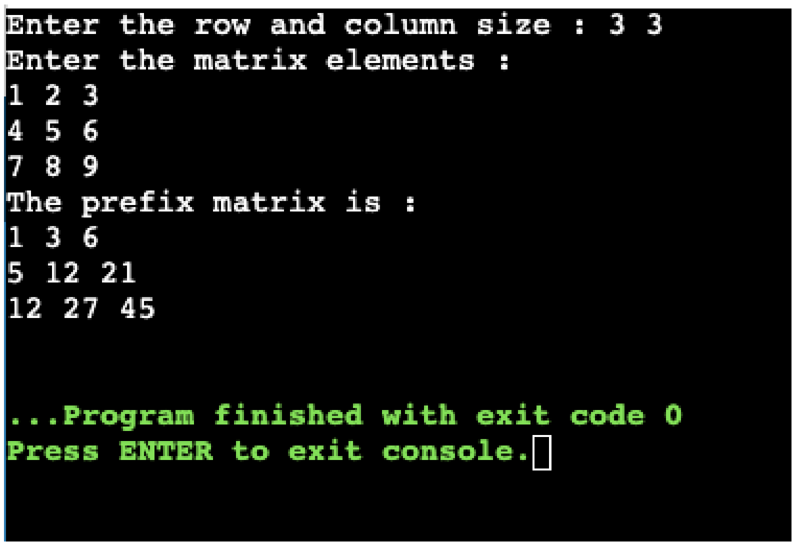
2) Given a matrix arr[][] of integers, print the prefix sum matrix for it.

Sample Input:[[1,2,3],[4,5,6],[7,8,9]]
Sample Output:[[1,3,6],[5,12,21],[12,27,45]]

Sample Input:[[1,0,1],[0,1,0]]
Sample Output:[[1,1,2],[1,2,3]]

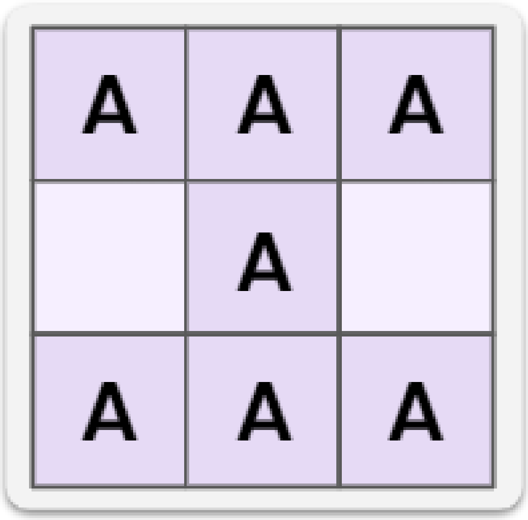
Explanation: We will first convert the first row and column of the matrix to prefix row and column. We will then convert the remaining matrix to prefix matrix as it requires the elements of first row and column to be the prefix sum.

```
#include <iostream>
using namespace std;

int main()
{
   int r;
   int c;
   cout<<"Enter the row and column size : ";
   cin>>r>>c;
   int arr[r][c];
   cout<<"Enter the matrix elements :  "<<endl;
   for(int i=0;i<r;i++)
   {
       for(int j=0;j<c;j++)
       {
           cin>>arr[i][j];
       }
   }
    int temp[r][c];
    temp[0][0]=arr[0][0];
    for(int i=1;i<r;i++)
    {
        temp[i][0]=temp[i-1][0]+arr[i][0];
    }
    for(int j=1;j<c;j++)
    {
        temp[0][j]=temp[0][j-1]+arr[0][j];
    }
    for(int i=1;i<r;i++)
    {
        for(int j=1;j<c;j++)
        {
            temp[i][j]=temp[i-1][j]+temp[i][j-1]-temp[i-1][j-1]+arr[i][j];
        }
    }
    //printing the prefix sum matrix
    cout<<"The prefix matrix is : "<<endl;
    for(int i=0;i<r;i++)
    {
        for(int j=0;j<c;j++)
        {
            cout<<temp[i][j]<<" ";
        }
        cout<<endl;
    }
}
```



```
Enter the row and column size : 3 3
Enter the matrix elements :
1 2 3
4 5 6
7 8 9
The prefix matrix is :
1 3 6
5 12 21
12 27 45


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

3) You are given an m x n integer matrix grid. Here m>=3 and n>=3
We define an hourglass as a part of the matrix with the following shape:



Return the *maximum sum of the elements of an hourglass.*

Note that an hourglass cannot be rotated and must be entirely contained within the matrix.

Sample Input: [[**1,2,3**],
[4,**5**,6],
[**7,8,9**]]

Sample Output: 35

Explanation: This has only one hourglass shape i.e. 1+2+3+5+7+8+9=35

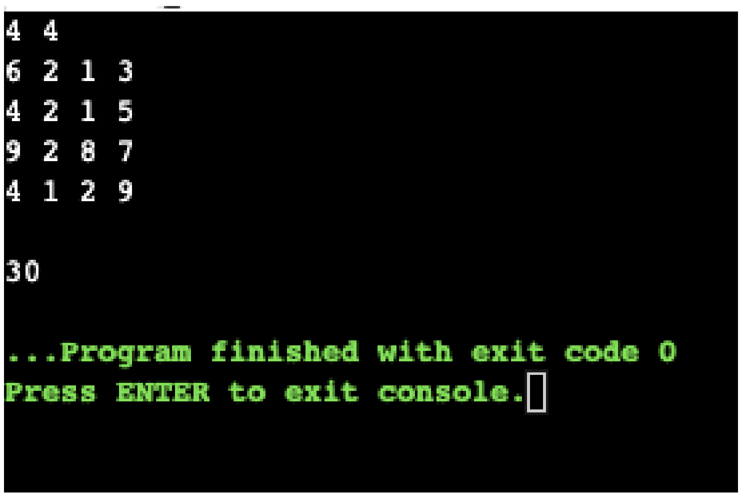
Sample Input: [[**6,2,1**,3],
[4,2,1,5],
[**9,2,8**,7],
[4,1,2,9]]

Sample Output:30

Explanation: Amongst all possible hourglass in this matrix the maximum sum will be of the hourglass 6+2+1+2+9+2+8

Explanation: An hourglass shape is possible only in matrix with dimensions greater than 3*3. Each 3*3 matrix can have only one hourglass within itself so we need to find the sum of hourglass elements in all the 3*3 matrix and amongst those we need to find the max sum.

```
#include <iostream>
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];
   }
  }
  cout<<endl;
  int ans = 0;
       int sumsofar=0;

       for(int i=0; i<=m-3; i++)
       {
           for(int j=0; j<=n-3; j++)
           {
               sumsofar=0;
               for(int k=j; k<j+3; k++)
               {
                   sumsofar = sumsofar + mat[i][k]; //this adds the first row of hourglass shape
                   sumsofar = sumsofar + mat[i+2][k]; //this adds the third row of hourglass shape
               }
               sumsofar += mat[i+1][j+1]; //this adds the mid element of the hourglass
              if(ans<sumsofar)
              {
               ans=sumsofar; //this stores the maximum sum hourglass
              }
           }
       }
       cout<<ans;
}
```



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

30

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