B - Print 2D Array in Spiral Order

Given a 2D Array of Size MxN. Print the elements in spiral order.

**Input Format**

M N - Two integers representing Rows and Columns of Array.  
Next M lines contains N integers each.

**Constraints**

xx

**Output Format**

MxN space separated numbers on single line of output.

**Sample Input**

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

**Sample Output**

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

**Explanation**

Spiral Order is First Row, Last Column, Last Row (Reverse), First Column (Reverse), Keep following the same pattern.

#include <iostream>

#include <vector>

using namespace *std*;

void fill2DMatrix(*vector*<*vector*<int>>& matrix)

{

int ele;

int rows = matrix.*size*(), cols = matrix[0].*size*();

for (auto i = 0; i < rows; i++)

{

for (auto j = 0; j < cols; j++)

{

*cin* >> ele;

matrix[i][j] = ele;

}

}

}

void printInSpiralOrder(*vector*<*vector*<int>>& matrix)

{

int minr = 0; int minc = 0;

int maxr = matrix.*size*() - 1; int maxc = matrix[0].*size*() - 1;

int totalElements = matrix.*size*() \* matrix[0].*size*();

int count = 0;

while (count < totalElements)

{

// TOP wall

for (auto i = minr, j = minc; j <= maxc && count < totalElements; j++)

{

*cout* << matrix[i][j] << " ";

count++;

}

minr++;

// Right wall

for (auto i = minr, j = maxc; i <= maxr && count < totalElements; i++)

{

*cout* << matrix[i][j] << " ";

count++;

}

maxc--;

// Bottom wall

for (auto i = maxr, j = maxc; j >= minc && count < totalElements; j--)

{

*cout* << matrix[i][j] << " ";

count++;

}

maxr--;

// Left wall

for (auto i = maxr, j = minc; i >= minr && count < totalElements; i--)

{

*cout* << matrix[i][j] << " ";

count++;

}

minc++;

}

}

int main()

{

int n, m;

*cin* >> n >> m;

*vector*<*vector*<int>> matrix(n, *vector*<int>(m));

fill2DMatrix(matrix);

printInSpiralOrder(matrix);

return 0;

}