### Spirally traversing a matrix

Given a matrix of size r\*c. Traverse the matrix in spiral form.

**Example 1:**

**Input**:

r = 4, c = 4

matrix[][] = {{1, 2, 3, 4},

  {5, 6, 7, 8},

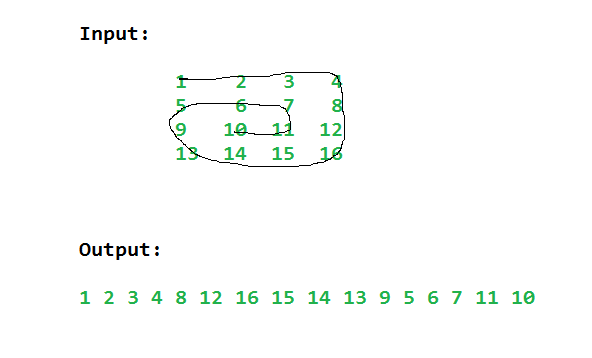
  {9, 10, 11, 12},

  {13, 14, 15,16}}

**Output**:

1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10

**Explanation**:



**Example 2:**

**Input**:

r = 3, c = 4

matrix[][] = {{1, 2, 3, 4},

  {5, 6, 7, 8},

  {9, 10, 11, 12}}

**Output**:

1 2 3 4 8 12 11 10 9 5 6 7

**Explanation**:

Applying same technique as shown above,

output for the 2nd testcase will be

1 2 3 4 8 12 11 10 9 5 6 7.

JAVA CODE

//{ Driver Code Starts

import java.io.\*;

import java.util.\*;

class CodingMaxima

{

public static void main(String args[])throws IOException

{

Scanner sc = new Scanner(System.in);

int t = sc.nextInt();

while(t-- > 0)

{

int r = sc.nextInt();

int c = sc.nextInt();

int matrix[][] = new int[r][c];

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

{

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

matrix[i][j] = sc.nextInt();

}

Solution ob = new Solution();

ArrayList<Integer> ans = ob.spirallyTraverse(matrix, r, c);

for (Integer val: ans)

System.out.print(val+" ");

System.out.println();

}

}

}

// } Driver Code Ends

class Solution

{

//Function to return a list of integers denoting spiral traversal of matrix.

static ArrayList<Integer> spirallyTraverse(int matrix[][], int r, int c)

{

ArrayList<Integer> ar=new ArrayList<Integer>();

int top=0, bottom=r-1, left=0, right=c-1;

while(top<=bottom && left <= right){

for(int i=left;i<=right;i++){

ar.add(matrix[top][i]);

}

top++;

for(int i=top;i<=bottom;i++){

ar.add(matrix[i][right]);

}

right--;

if(top<= bottom){

for(int i=right;i>=left;i--){

ar.add(matrix[bottom][i]);

}

}

bottom--;

if(left<=right){

for(int i=bottom ;i>=top;i--){

ar.add(matrix[i][left]);

}

}

left++;

}

return ar;

}

}