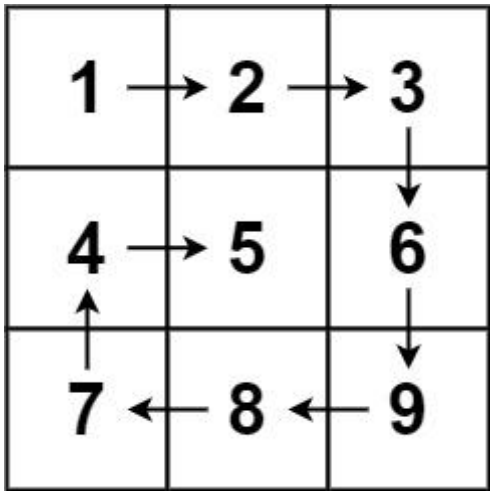
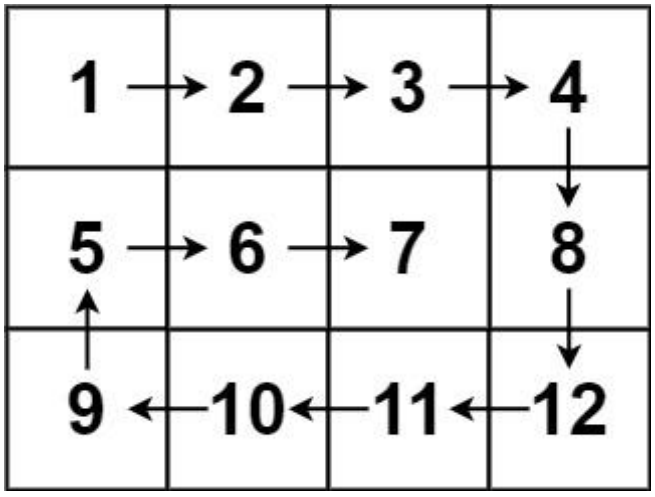
Given an m x n matrix, return *all elements of the* matrix *in spiral order*.

Example 1:



Input: matrix = [[1,2,3],[4,5,6],[7,8,9]]
Output: [1,2,3,6,9,8,7,4,5]

Example 2:



Input: 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]

Constraints:

- m == matrix.length
- n == matrix[i].length

- 1 <= m, n <= 10
- -100 <= matrix[i][j] <= 100

Solution:

```
class Solution {
    public List<Integer> spiralOrder(int[][] matrix) {
        List<Integer> ans=new ArrayList<>();
        int startrow=0,startcol=0,endrow=matrix.length-1,endcol=matrix[0].length-1;
        while(startcol<=endcol&&startrow<=endrow){
            for(int i=startcol;i<=endcol;i++){
                ans.add(matrix[startrow][i]);
            }
            for(int j=startrow+1;j<=endrow;j++){
                ans.add(matrix[j][endcol]);
            }
            for(int k=endcol-1;k>=startcol;k--){
                if(startrow==endrow){
                    break;
                }
                    ans.add(matrix[endrow][k]);
            }
            for(int l=endrow-1;l>=startrow+1;l--){
                 if(startcol==endcol){
                    break;
                }
                ans.add(matrix[l][startcol]);
            }
            startrow++;
            endrow--;
            startcol++;
            endcol--;
        }
        return ans;
    }
}
```