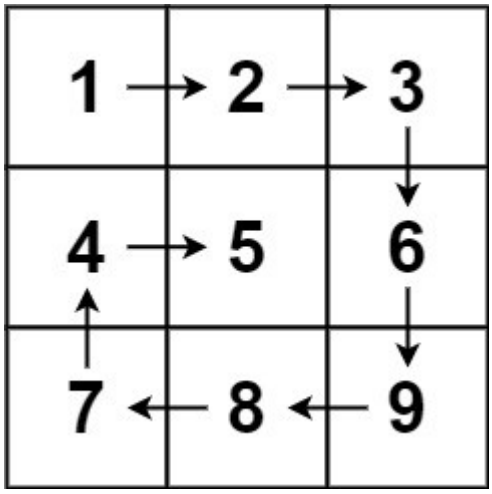
54. Spiral Matrix

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

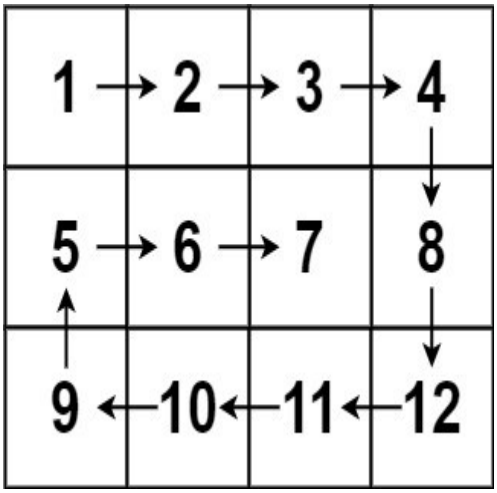
Example 1:

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]



54. Spiral Matrix

```
/*
   Time complexity: O(n^2)
   Space complexity: O(1)
*/
class Solution {
   public:
      std::vector<int> spiralOrder(std::vector<std::vector<int>>& matrix) {
         int m=matrix.size();
         int n=matrix[0].size();

         std::vector<int> ans;

         int left=0,right=n-1,top=0,bottom=m-1;
         bool ok=left<=right && top<=bottom;

         while(ok){
            if(ok){
               // Get all values in the top row
               for(int col=left;col<=right;++col) ans.push_back(matrix[top][col]);
               top++;

               // Get all values in the right column
               for(int row=top;row<=bottom;++row) ans.push_back(matrix[row][right]);
               right--;
            }

            ok=left<=right && top<=bottom;

            if(ok){
               // Get all values in the bottom row (reversed order)
               for(int col=right;col>=left;--col) ans.push_back(matrix[bottom][col]);
               bottom--;

               // Get all values in the left column (reversed order)
               for(int row=bottom;row>=top;--row) ans.push_back(matrix[row][left]);
               left++;
            }

            ok=left<=right && top<=bottom;

         }
         return ans;
      }
```

```
};
```