Given a matrix of **M * N** elements (M rows, N columns), return all elements of the matrix in spiral order.

Problem Constraints

1 <= **M, N** <= 1000

Input Format

The first argument is a matrix A.

Output Format

Return an array of integers representing all elements of the matrix in spiral order.

Example Input

```
A =
  [
     [ 1, 2, 3 ],
     [ 4, 5, 6 ],
     [ 7, 8, 9 ]
  ]
```

Example Output

[1, 2, 3, 6, 9, 8, 7, 4, 5]

Code:

```
vector<int> Solution::spiralOrder(const vector<vector<int> > &A) {
   vector<int> result;
```

```
    int top = 0, bottom = A.size() - 1;
    int left = 0, right = A[0].size() - 1;

    while (top <= bottom && left <= right) {
        for (int i = left; i <= right; i++) {
            result.push_back(A[top][i]);
        }
        top++;

        for (int i = top; i <= bottom; i++) {
            result.push_back(A[i][right]);
        }
        right--;

        if (top <= bottom) {
            for (int i = right; i >= left; i--) {
                result.push_back(A[bottom][i]);
            }
            bottom--; // Move the bottom boundary up
        }

        if (left <= right) {
            for (int i = bottom; i >= top; i--) {
                result.push_back(A[i][left]);
            }
            left++; // Move the left boundary right
        }
    }

    return result;
}
```