Our Solution(s) Run Code

Solution 1 Solution 2

```cpp
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

using namespace std;

// O(n) time | O(n) space - where n is the total number of elem
vector<int> spiralTraverse(vector<vector<int>> array) {
  if (array.size() == 0)
    return {};

  vector<int> result = {};
  int startRow = 0;
  int endRow = array.size() - 1;
  int startCol = 0;
  int endCol = array[0].size() - 1;

  while (startRow <= endRow && startCol <= endCol) {
    for (int col = startCol; col <= endCol; col++) {
      result.push_back(array[startRow][col]);
    }

    for (int row = startRow + 1; row <= endRow; row++) {
      result.push_back(array[row][endCol]);
    }

    for (int col = endCol - 1; col >= startCol; col--) {
      if (startRow == endRow)
        break;
      result.push_back(array[endRow][col]);
    }

    for (int row = endRow - 1; row > startRow; row--) {
      if (startCol == endCol)
        break;
```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```cpp
using namespace std;

vector<int> spiralTraverse(vector<vector<int>> array) {
  // Write your code here.
  return {};
}

```

Our Tests

Custom Output Submit Code

```
package
  void Run() {
    RunTests("Test Cases #", [](){
      vector<vector<int>> input = {
          {},
      };
      vector<int> expected = {};
      vector<int> actual = solution(input);
      assert(expected == actual);
    });
    RunTests("Test Cases #", [](){
      vector<vector<int>> input = {
          {3, 3},
          {4, 5},
      };
      vector<int> expected = {3, 3, 4, 4};
      vector<int> actual = solution(input);
      assert(expected == actual);
    });
    RunTests("Test Cases #", [](){
      vector<vector<int>> input = {
```

Run or submit code when you're ready.