Our Solution(s) Run Code

Solution 1 Solution 2

```python
# Copyright © 2020 AlgoExpert, LLC. All rights reserved.

# O(n) time | O(n) space - where n is the total number of elemer
def spiralTraverse(array):
    result = []
    startRow, endRow = 0, len(array) - 1
    startCol, endCol = 0, len(array[0]) - 1

    while startRow <= endRow and startCol <= endCol:
        for col in range(startCol, endCol + 1):
            result.append(array[startRow][col])

        for row in range(startRow + 1, endRow + 1):
            result.append(array[row][endCol])

        for col in reversed(range(startCol, endCol)):
            if startRow == endRow:
                break
            result.append(array[endRow][col])

        for row in reversed(range(startRow + 1, endRow)):
            if startCol == endCol:
                break
            result.append(array[row][startCol])

        startRow += 1
        endRow -= 1
        startCol += 1
        endCol -= 1

    return result

```

Your Solutions Run Code

Solution 1 Solution 2 Solution 3

```python
def spiralTraverse(matrix):
    # Write your code here.
    pass

```

Our Tests

Custom Output Submit Code

```
import unittest


class TestProgram(unittest.TestCase):
    def test_case_0(self):
        matrix = [[0]]
        expected = [0]
        self.assertEqual(program.spiralOrder(matrix), expected)

    def test_case_1(self):
        matrix = [[1, 2], [3, 4]]
        expected = [1, 2, 4, 3]
        self.assertEqual(program.spiralOrder(matrix), expected)

    def test_case_2(self):
        matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
        expected = [1, 2, 3, 6, 9, 8, 7, 4, 5]
        self.assertEqual(program.spiralOrder(matrix), expected)

    def test_case_3(self):
        matrix = [[1, 2, 3, 4], [10, 11, 12, 5], [9, 8, 7, 6]
```

Run or submit code when you're ready.