

Our Solution(s)

Run Code

Your Solutions

Run Code

Solution 1

Solution 2

Solution 3

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 #include <unordered_map>
5 #include <algorithm>
6 using namespace std;
7
8 struct Point {
9     int x;
10    int y;
11
12    bool operator==(Point point2) { return x == point2.x && y == point2.y; }
13 };
14
15 string UP = "up";
16 string RIGHT = "right";
17 string DOWN = "down";
18 string LEFT = "left";
19
20 unordered_map<string, unordered_map<string, vector<Point>>>
21 getCoordsTable(vector<Point> coords);
22 string getCoordDirection(Point coord1, Point coord2);
23 int getRectangleCount(
24     vector<Point> coords,
25     unordered_map<string, unordered_map<string, vector<Point>>> coord:
26 int clockwiseCountRectangles(
27     Point coord,
28     unordered_map<string, unordered_map<string, vector<Point>>> coord:
29     string direction, Point origin);
30 string getNextClockwiseDirection(string direction);
31 string coordToString(Point coord);
32
33 // O(n^2) time | O(n^2) space - where n is the number of coordinates
```

Solution 1

Solution 2

Solution 3

```
1 #include <vector>
2
3 using namespace std;
4
5 struct Point {
6     int x;
7     int y;
8 };
9
10 int rectangleMania(vector<Point> coords) {
11     // Write your code here.
12     return -1;
13 }
14
```

Our Tests

Custom Output

Submit Code

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 #include <vector>
4 #include <unordered_map>
5 #include <algorithm>
6 using namespace std;
7
8 struct Point {
9     int x;
10    int y;
11
12    bool operator==(Point point2) { return x == point2.x && y == point2.y; }
13 };
14
15 string UP = "up";
16 string RIGHT = "right";
17 string DOWN = "down";
18 string LEFT = "left";
19
20 unordered_map<string, unordered_map<string, vector<Point>>>
21 getCoordsTable(vector<Point> coords);
22 string getCoordDirection(Point coord1, Point coord2);
23 int getRectangleCount(
24     vector<Point> coords,
25     unordered_map<string, unordered_map<string, vector<Point>>> coord:
26 int clockwiseCountRectangles(
27     Point coord,
28     unordered_map<string, unordered_map<string, vector<Point>>> coord:
29     string direction, Point origin);
30 string getNextClockwiseDirection(string direction);
31 string coordToString(Point coord);
32
33 // O(n^2) time | O(n^2) space - where n is the number of coordinates
```

```
1 #include <vector>
2
3 using namespace std;
4
5 struct Point {
6     int x;
7     int y;
8 };
9
10 int rectangleMania(vector<Point> coords) {
11     // Write your code here.
12     return -1;
13 }
14
```

```
24
25
26 def test_case_27, 1000000
27   answer = 0
28   for i in 1..1000000
29     answer += i
30   end
31
32   answer
33 end
34
35 def test_case_28, 1000000
36   answer = 0
37   for i in 1..1000000
38     answer += i
39   end
40
41   answer
42 end
43
44 def test_case_29, 1000000
45   answer = 0
46   for i in 1..1000000
47     answer += i
48   end
49
50   answer
51 end
```

Run or submit code when you're ready.