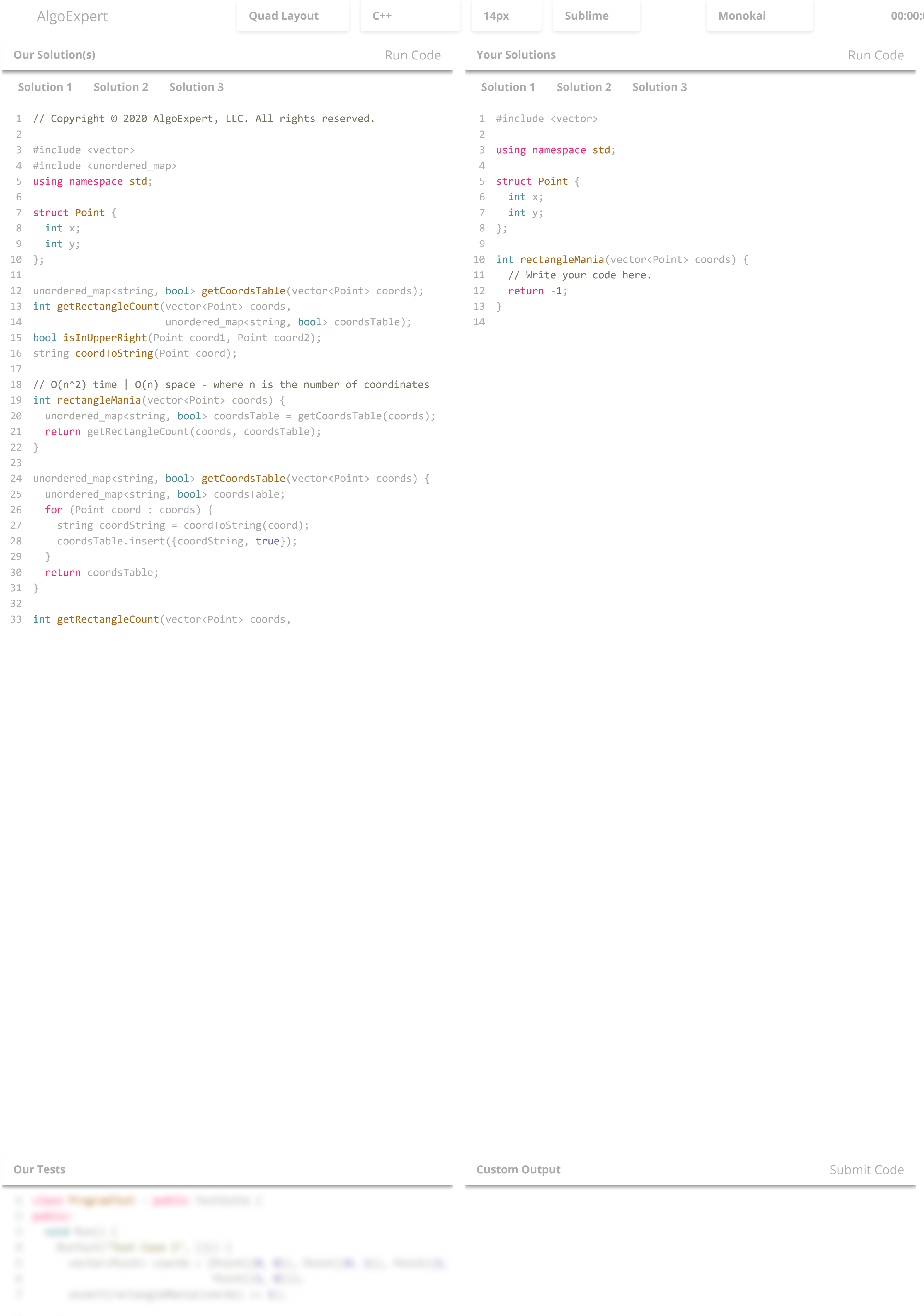

```cpp
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

#include <vector>
#include <unordered_map>
using namespace std;

struct Point {
  int x;
  int y;
};

unordered_map<string, bool> getCoordsTable(vector<Point> coords);
int getRectangleCount(vector<Point> coords,
                      unordered_map<string, bool> coordsTable);
bool isInUpperRight(Point coord1, Point coord2);
string coordToString(Point coord);

// O(n^2) time | O(n) space - where n is the number of coordinates
int rectangleMania(vector<Point> coords) {
  unordered_map<string, bool> coordsTable = getCoordsTable(coords);
  return getRectangleCount(coords, coordsTable);
}

unordered_map<string, bool> getCoordsTable(vector<Point> coords) {
  unordered_map<string, bool> coordsTable;
  for (Point coord : coords) {
    string coordString = coordToString(coord);
    coordsTable.insert({coordString, true});
  }
  return coordsTable;
}

int getRectangleCount(vector<Point> coords,
```

```cpp
#include <vector>

using namespace std;

struct Point {
  int x;
  int y;
};

int rectangleMania(vector<Point> coords) {
  // Write your code here.
  return -1;
}

```

[illegible]

Run or submit code when you're ready.