Batch: SY-IT(B3) **Experiment Number:8**

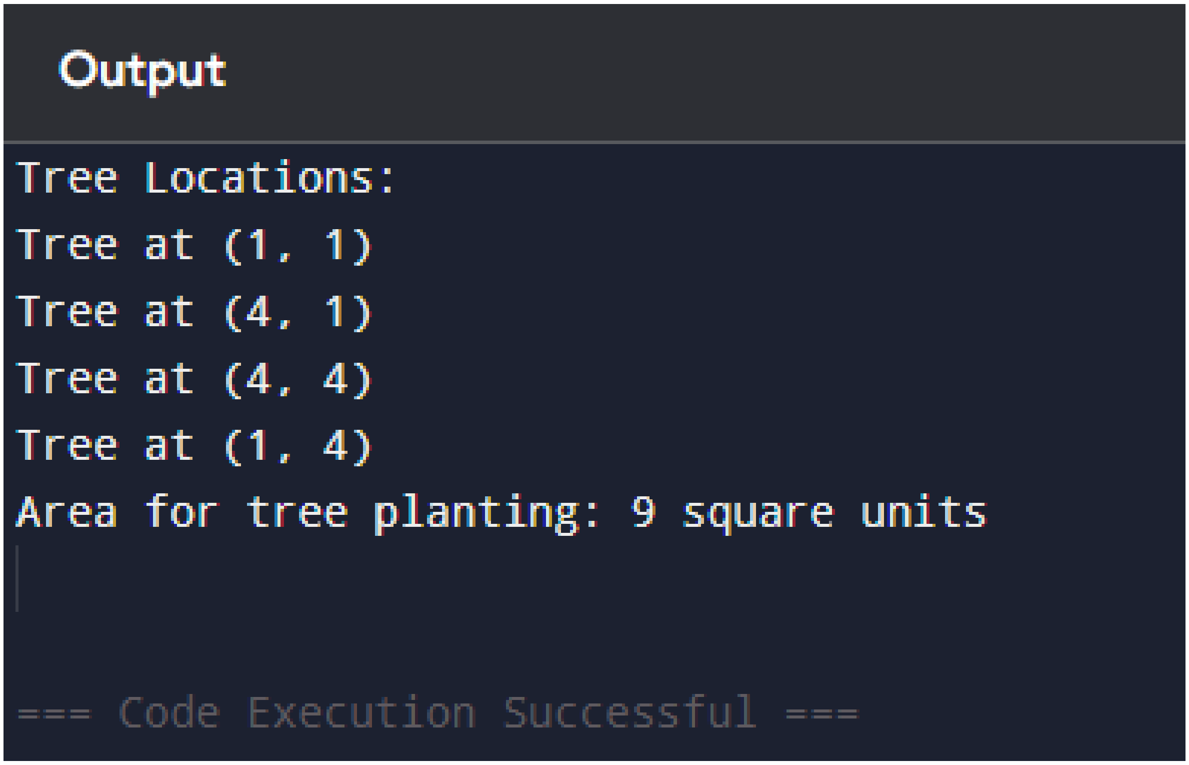
Roll Number: 16010423076 **Name: Ritesh Jha**

Aim of the Experiment: To study Lattice Polygons and Pick's Theorem for implementation of problem statement that is based on finding area of concave shape Polygon.

Program/ Steps:

```
#include <iostream>
#include <vector>
#include <cmath>

// Structure to represent a point on the grid
struct Point {
    int x, y;
};

// Function to calculate distance between two points
double calculateDistance(Point a, Point b) {
    return sqrt(pow(a.x - b.x, 2) + pow(a.y - b.y, 2));
}

// Function to check if points are valid for tree planting
bool canPlantTrees(std::vector<Point>& points) {
    // Check minimum distance between points
    for (int i = 0; i < points.size(); i++) {
        for (int j = i + 1; j < points.size(); j++) {
            if (calculateDistance(points[i], points[j]) < 1) {
                return false; // Points are too close
            }
        }
    }
    return true;
}

// Calculate area of polygon using shoelace formula
double calculatePolygonArea(std::vector<Point>& points) {
    double area = 0.0;
    int n = points.size();

    for (int i = 0; i < n; i++) {
        int j = (i + 1) % n;
        area += (points[i].x * points[j].y) - (points[j].x * points[i].y);
    }

    return std::abs(area) / 2.0;
}

int main() {
    // Example polygon vertices (tree planting locations)
    std::vector<Point> treeLocations = {
        {1, 1},   // First tree
        {4, 1},   // Second tree
        {4, 4},   // Third tree
        {1, 4}    // Fourth tree
    };

    // Check if trees can be planted
    if (!canPlantTrees(treeLocations)) {
        std::cout << "Cannot plant trees. Some locations are too close." << std::endl;
        return 1;
    }

    // Calculate and display polygon area
    double polygonArea = calculatePolygonArea(treeLocations);

    // Print tree locations
    std::cout << "Tree Locations:" << std::endl;
    for (const auto& tree : treeLocations) {
        std::cout << "Tree at (" << tree.x << ", " << tree.y << ")" << std::endl;
    }

    // Print polygon area
    std::cout << "Area for tree planting: " << polygonArea << " square units" << std::endl;

    return 0;
}
```

Output/Result:



```
Tree Locations:
Tree at (1, 1)
Tree at (4, 1)
Tree at (4, 4)
Tree at (1, 4)
Area for tree planting: 9 square units


=== Code Execution Successful ===
```

Outcomes:

CO4. Learn effective computation and programming practices for numeric and string operations and computation geometry

Conclusion (based on the Results and outcomes achieved):

From this experiment, I learned how to apply Pick’s Theorem and the Shoelace Theorem to compute the area of a concave polygon formed by lattice points. Additionally, I implemented a method to verify whether trees can be planted while maintaining a minimum distance constraint. This experiment helped in understanding computational geometry concepts and their real-world applications in problem-solving.

References:

1. Steven S. Skiena Miguel A. Revilla,”Programming challenges, The Programming Contest Training Manual”, Springer, 2006
2. T.H. Coreman ,C.E. Leiserson,R.L. Rivest, and C. Stein, " Introduction to algorithms", 3rd Edition 2009, Prentice Hall India Publication
3. Antti Laaksonen, “Guide to Competitive Programming”,Springer,2018
4. Gayle Laakmann McDowell,” Cracking the Coding Interview”,CareerCup LLC,2015
5. Antti Laaksonen, “Competitive Programmer’s Handbook”, Hand book, 2018
6. Steven Halim and Felix Halim, “Competitive Programming 3: The Lower Bounds of Programming Contests”, Handbook for ACM ICPC