Experiment No. 8

Title: Implementation of problem based on Computational Geometry

Batch:A2 **Roll No: 16010421063** **Experiment No.:8**

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

Resources needed: Text Editor, C/C++ IDE

Theory:

Rectangular grids of unit-spaced points (also called lattice points) are at the heart of any grid-based coordinate system. In general, there will be about one grid point per unit-area in the grid, because each grid point can be assigned to be the upper-right-hand corner of a different 1×1 empty rectangle. Thus the number of grid points within a given figure should give a pretty good approximation to the area of the figure. Pick's theorem gives an exact relation between the area of a lattice polygon (anon-intersecting figure whose vertices all lie on lattice points) and the number of lattice points on/in the polygon. Suppose there are I(P) lattice points inside of P and B(P) lattice points on the boundary of P. Then the area A(P) of P is given by

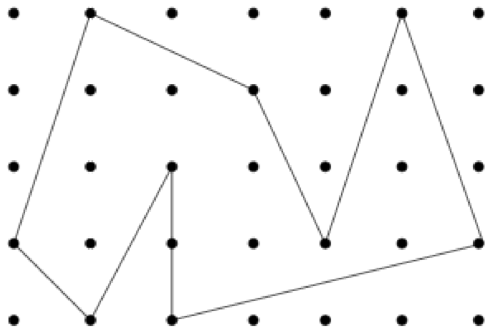
$$A(P)=I(P)+B(P)/2-1$$

as illustrated in Figure below. For example, consider a triangle defined by coordinates (x,1), (y,2), and (y+k,2). No matter what x, y, and k are there can be no interior points, because the three points lie on consecutive rows of the lattice. Lattice point (x,1) serves as the apex of the triangle, and there are k+ 1 lattice points on the boundary of the base. Thus I(P)=0,B(P)=k+ 2, and so the area is k/2, precisely what you get from the triangle area formula. As another example, consider a rectangle defined by corners (x1,y1) and (x2,y2). The number of boundary points is

$$B(P)=2|y2-y1+1|+2|x2-x1+1|-4 = 2(\Delta y-\Delta x)$$

with the 4-term to avoid double-counting the corners. The interior is the total number of points in or on the rectangle minus the boundary, giving

$$I(P)=(\Delta x+ 1)(\Delta y+1)-2(\Delta y-\Delta x)$$



Pick's theorem correctly computes the area of the rectangle as Δx Δy. Applying Pick's theorem requires counting lattice points accurately. This can in principle be done by exhaustive testing for small area polygons using functions that (1) test whether a point lies on a line segment and (2) test whether a point is inside or outside a polygon.

Example of Pick`s theorem:

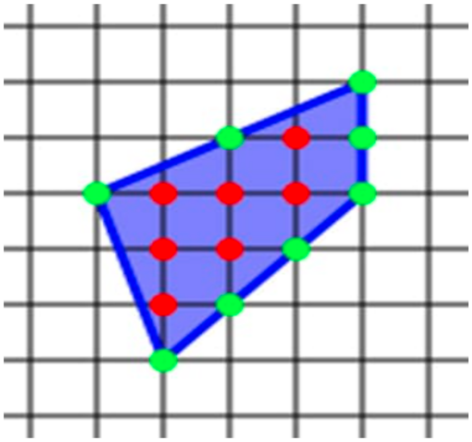
Pick's theorem is one of those theorems in mathematics which seems too simple to be true. Take any polygon and lay it on a lattice. (A lattice is a grid of points where every point has whole number (integer) coordinates.) According to Pick's Theorem all you need to do to find the area of a polygon is to count the points on the interior and on the boundary of the shape.

Pick's Theorem then states that:

$$Area = i + \frac{b}{2} - 1$$

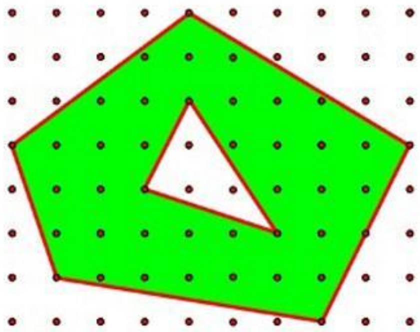
(i stands for the number of points in the interior of the shape, b stands for the number of points on the boundary of the shape.)

Take the polygon below as an example. In this polygon $i = 7$ and $b = 8$. According to Pick's Theorem, $Area = 7 + \frac{8}{2} - 1 = 10$ which is the correct area of the polygon.



Pick's Theorem does however only work for simple polygons – this means polygons which don't intersect themselves and don't have any holes, however it can be adapted to include holes.

Pick's Theorem for polygons with holes



To use Pick's Theorem on a shape like the one above you simply need to apply the theorem to the green shape without the hole and then subtract the area of the hole.

Area of Green shape (without hole) using Pick's Theorem: $38 + \frac{6}{2} - 1 = 40$

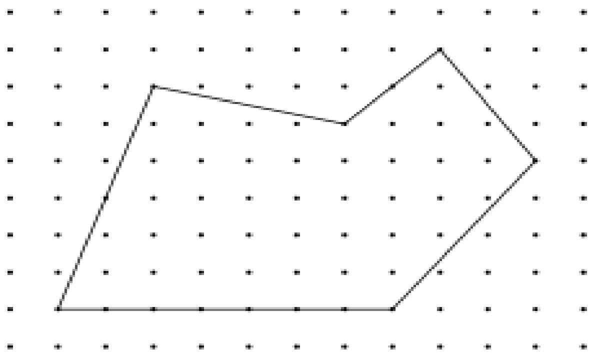
Area of the hole using Pick's Theorem: $3 + \frac{3}{2} - 1 = 3 + \frac{1}{2}$

Total Area of the shape = $40 - (3 + \frac{1}{2}) = 36 + \frac{1}{2}$

<u>Activity</u>:

Problem Statement: Green environment.

Mr. Cooper wants to plant trees on an island; however the island is not rectangular in shape (refer figure below). Using picks theorem he has to find a polygon area with vertices lying on the grid points and plant the trees strictly inside the grid points of this polygon with minimum distance from boundary of 1 unit. He has selected you for this job to write the program for above problem and state how many trees can be planted and what is area of the polygon mentioned below, distance between grid points is 1 unit.



Input format

A student has to consider input format as per their intelligence.

Output format
A student has to consider Output format as per their intelligence.

Constraints:
A student has to Design constraints as per your intelligence.

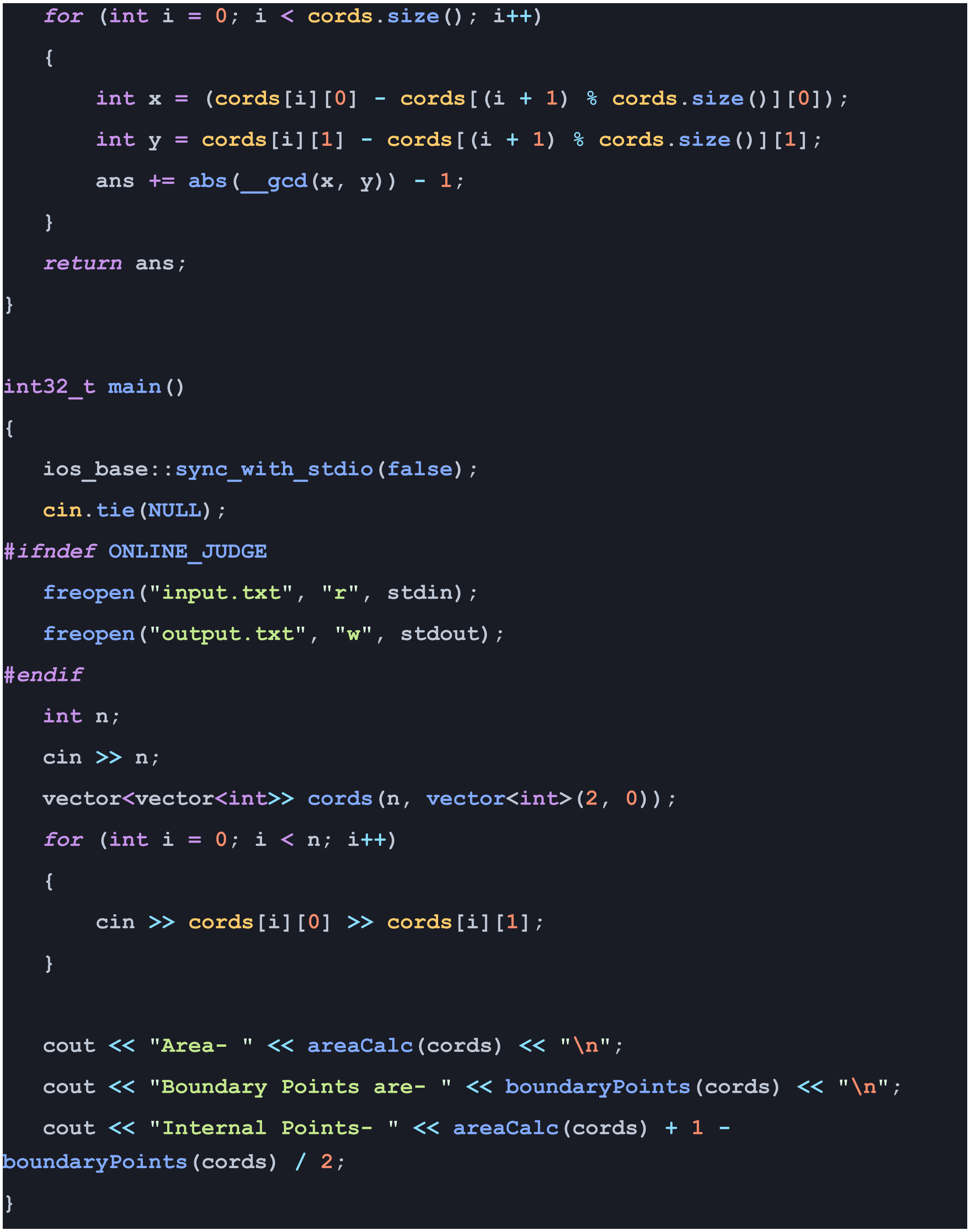
Program:

```
#include <bits/stdc++.h>
#define int long long
#define ld long double
using namespace std;

long double cross(ld x1, ld y1, ld x2, ld y2)
{
   return x1 * y2 - x2 * y1;
}

ld areaCalc(vector<vector<int>> &cords)
{
   ld ans = 0;
   for (int i = 2; i < cords.size(); i++)
   {
       ans += cross(cords[i][0] - cords[0][0], cords[i][1] -
cords[0][1], cords[i - 1][0] - cords[0][0], cords[i - 1][1] -
cords[0][1]);
   }
   return fabs(ans / 2);
}

int boundaryPoints(vector<vector<int>> &cords)
{
   int ans = cords.size();
```

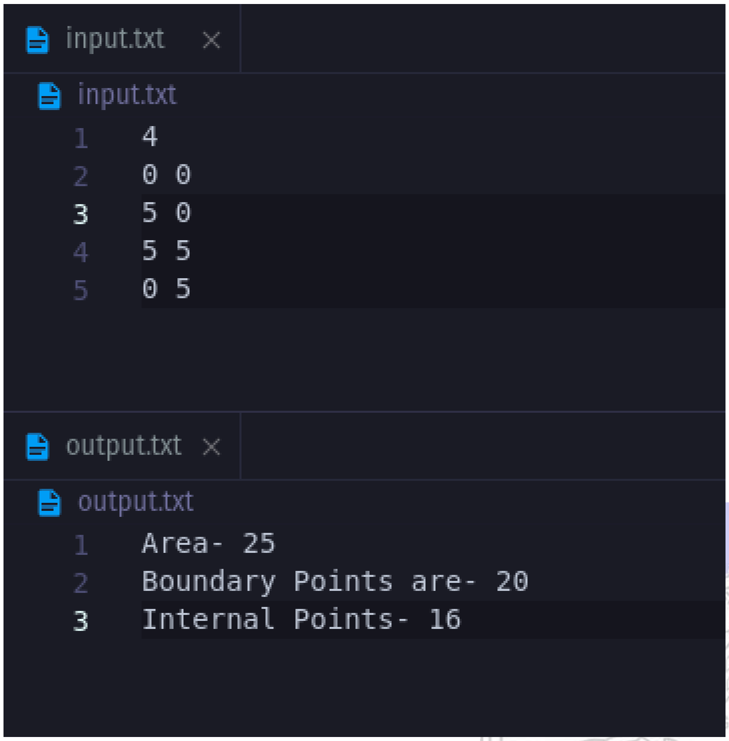


```
    for (int i = 0; i < cords.size(); i++)
    {
        int x = (cords[i][0] - cords[(i + 1) % cords.size()][0]);
        int y = cords[i][1] - cords[(i + 1) % cords.size()][1];
        ans += abs(__gcd(x, y)) - 1;
    }
    return ans;
}

int32_t main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
#ifndef ONLINE_JUDGE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
    int n;
    cin >> n;
    vector<vector<int>> cords(n, vector<int>(2, 0));
    for (int i = 0; i < n; i++)
    {
        cin >> cords[i][0] >> cords[i][1];
    }

    cout << "Area- " << areaCalc(cords) << "\n";
    cout << "Boundary Points are- " << boundaryPoints(cords) << "\n";
    cout << "Internal Points- " << areaCalc(cords) + 1 -
boundaryPoints(cords) / 2;
}
```

Output:



```
4
0 0
5 0
5 5
0 5
```

```
Area- 25
Boundary Points are- 20
Internal Points- 16
```

Outcomes:

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

Conclusion: (Conclusion to be based on the objectives and outcomes achieved)

Found the area of the entire polygon and the number of internal points using picks theorem

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

K. J. SOMAIYA COLLEGE OF ENGG.