Bangladesh University of Engineering & Technology
Department of Computer Science & Engineering
CSE 464: Computational Geometry Sessional
Session January 2018

Assignment 1
Computation of Convex Hull

April 7, 2018

1 Introduction

In this assignment you will implement *Graham Scan* algorithm to compute convex hull of a set of points in 2D. Thus you will implement an $O(n \log n)$ time algorithm to compute 2D convex hull.

2 Input

You will have to take input from a file . The first line of the input is a number indicating number of points n. Each of the next n lines contains a pair of numbers indicating the x-coordinate and y-coordinate of each point.

3 Output

Output of your program consists of two parts: Console output and Graphical output.

Console Output: In console you have to print the length of the boundary of the convex hull in the first line. In the following h lines, you will have to print the extreme points of the convex hull in anti-clockwise order where h is the number of extreme points in the convex hull.

Graphical Output: You have to show the points and the convex hull graphically (may be in OpenGL).

4 Sample I/O

The Table 1 contains one sample input output. The Figure 1 demonstates sample graphical output

Input	Output
5	16
0 0	0 0
4 0	4 0
0 4	4 4
4 4	0 4
2 2	

Table 1: Sample Input Output

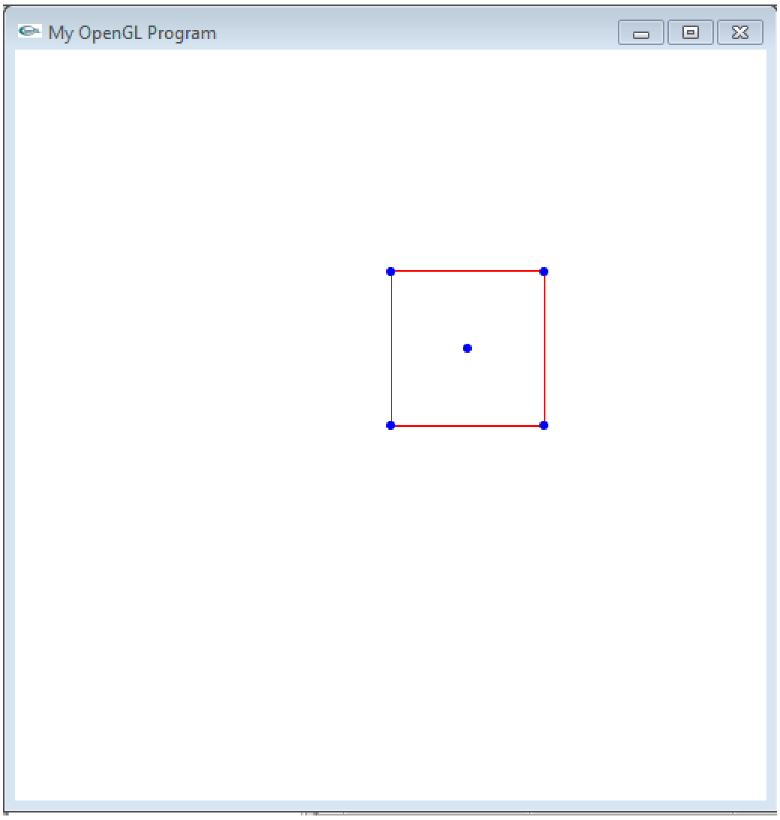

Figure 1: Sample Graphical Output

5 Test Cases:

You have to generate sufficient test cases including some tricky cases. As the complexity of *Graham Scan* algorithm is determined by the sorting step, you cannot design test case that will directly influence running time for that particular case. However, you can design test cases where more points need to be popped from the stack or no points needs to be popped from the stack. Include these cases in your sample test cases.

6 Report

You will have to submit a report in a doc/docx file containing the following:

- Complexity of *Graham Scan* algorithm.
- Two test cases where *Graham Scan* performs better and worse respectively than *Gift Wrapping* algorithm.
- Two test cases where *Graham Scan* performs better and worse respectively than *Quick Hull* algorithm.

7 Important Notes

Please try to follow the instructions listed below while implementing your assignment:

- Implement using C++/Java programming language
- Be cautious about floating point arithmetic

8 Marks Distribution

- Implementation: 11
- Test Case: 3
- Console Output: 1
- Graphical Output: 2
- Report: 3

9 Rules

- You have to submit all your source codes via moodle. All the file name will be in following format

  <your 7 digit student id>_<additional name>

  For example, the submitted file name would look like 1305999_GrahamScan.cpp if it is submitted by a student having 1305999 as student id. Name your input file as <your 7 digit student id>_input1.txt, <your 7 digit student id>_input2.txt and so on. Put all your source files, input files and report in a folder (even if you put all your codes in only one file) named after your 7 digit student id and create a **zipped** archive of the folder. Then submit the zip file in moodle. Failure to submit properly will cause 10% deduction.

- Any type of plagiarism is strongly forbidden. -100% marks will be given to the students who will be found to be involved in plagiarism (from book, internet, from senior/class-mates code etc.). It does not matter who is the server and who is the client.
- Prepare for an online evaluation.

10 Deadline

Deadline is set at 11:55 pm, April 20, 2018 for all lab groups.