Assignment no. 1

given: February 28th, 2000
due: March 20th, 2000

The first two exercises are warm-up exercises to get you acquainted with the CGAL geometric algorithms library, which will be the platform for all the exercises in the course.

A boldface **(r)** following the exercise number indicates that a program should be submitted as part of the exercise.

In order to run the programs you will need an account on *nova*. You can also get a full working environment to install on your Linux or Windows machine at home. However, you will have to submit the homework on nova here.

Links to the necessary libraries, documentation and example program files will appear in the coming days on the course's web site:
http://www.math.tau.ac.il/~halperin/courses/appGC-spring2000.html

Exercise 1.1 **(r)** The input file to a program consists of integers. It starts with the number n of plane point coordinates followed by n pairs of (x, y) coordinates.

(a) The first two points describe a non-vertical line ℓ that passes through them. The rest of the pairs describe point no. 1 through point no. $(n-2)$. Write a program that reads such a file and identifies all the points in the file that lie **above** the line ℓ.

(b) We number the points 1 through n. Write a program that identifies all the collinear triples of points.

Both programs should print out the indices of the relevant points.

Exercise 1.2 **(r)** Write a program that gets as input two convex polygons and computes their Minkowski sum. In the course's web site you'll find a 'frame' program that does the input and output in CGAL, and calls a procedure
`bool mink_convex_convex(const Polygon &poly1, const Polygon &poly2, Polygon &sum);`
that you need to complete. The procedure returns `true` upon successful completion and `false` otherwise.

Exercise 1.3 A two-dimensional scene consists of k pairwise-disjoint convex polygons with a total of n edges. We are concerned with the *aspect graph* of this scene for parallel (or orthographic) projection.

(a) Give upper and lower bounds on the size of the aspect graph.

(b) Devise an efficient algorithm to compute this aspect graph.

Exercise 1.4 We are given two overlapping simple polygons P and Q with m and n vertices respectively. Describe an algorithm that will find the minimum length translation of Q such that the interiors of P and Q are disjoint.