Homework 2, CSCE 240, Fall 2018

Points

This is a 40 point assignment. The first 5 points are for part A, and the remaining 40 points are for part B.

The basic assignment is this. You are given an input file comprising 10 pairs (HARD-CODED 10) of floating point numbers. These are to be taken as (x, y) coordinates of 10 points in the plane. You are to run a double loop on pairs of points to find the pair of points that are closest together in Euclidean distance.

Part A

You are to read the input data into an appropriate
`vector<pair<double, double> >`
or two
`vector<double>`
instances and then dump the data using an appropriate function. You will get 5 points for doing this and for having your name and other boilerplate attribution at the top of each program file.

You will get zero points if you don't do the right read/write or if you don't have your name in each file.

Part B Assignment

You are to read the 10 pairs of numbers into a
`vector<pair<double, double> >`
or into two instances of a
`vector<double>`.

You are then to find, among all the pairs of (x, y) coordinate points, the two points that are closest together in Euclidean distance.

This will require you to have a function that is passed two pairs (x_1, y_1) and (x_2, y_2) and computes the Euclidean distance

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}.$$

COMMENT: The pairs of numbers are x, y coordinates in the plane. The Right Way to do this is to have a `vector` of values, each value of which is a member of the `Pair` class. HOWEVER, you may if you wish use one `vector` for the x coordinates and a second `vector` for the y coordinates.

Indeed, one might argue that this is the right way to get the program written. First write a program that gets you the right answer, and then improve it to use one `vector` of `Pair` instances instead of two `vector` structures. Getting the right answer requires knowing the logic of how the program should be written. Using the `Pair` class is a matter of working with the details of C++. Maybe it would be better to do things one step at a time.

IMPORTANT OBSERVATION: One of the reasons that "structures" like the `Pair` exist is the following. If you do separate `vector` instances for the x and the y values, then you have to subscript with the same subscript value into two different `vector`s and you have to ensure that the two instances are kept parallel in what they hold. This kind of thing has always led to programming errors. The use of an instance of `vector<pair<double, double> >` allows you to keep the values together and have only one list of things to look at.

Your program will have a header file `main.h` and a program file `main.cc` and will be compiled with a simple makefile.

You should read from standard input and write to standard output.