ECE 36800
PA 2: Collision Detection

Due Monday, July 15th at 11:59 PM

Goal.

In this assignment, you will implement a very simple vesion of collision detection. Your collision detector will load a "scene" consisting of a set of points in two-dimensional space, and then respond to a number of "collision queries". Each query will present you with the center and radius of a circle, and then ask you how many points in the scene collide with that circle.

Input & Output.

Your program should take as a command-line argument the name of a text file describing the scene, and then respond to queries on standard input. The text file will contain a list of points, with one point on each line. Each point will be represented by two integers, the x- and the y-coordinate respectively, separated by a space. A query will consist of three integers: the x- and y-coordinates of the center, and the radius.

Example.

Consider the following `points.txt` file:

```
100 -100
100 100
-100 100
-100 -100
100 0
0 100
-100 0
0 -100
```

An sample run of the program is shown below:

```
$ ./pa2 points.txt
> 0 0 100
4
> 0 0 142
8
> 50 50 100
3
```

Grading

For this assignment, you will be graded time *and* correctness, in addition to the usual memory management. 50% of the test cases will be relatively small, consisting of up to 100 points in the scene and up to 10 queries; for these, you will get full points just by returning the correct responses. The other 50% will be much larger, consisting of scenes with several million points, and potentially several thousand queries. In addition to checking for correctness, we will time these test cases, and your score will be based on your run time. As always, a 50% penalty will be applied to any test case with a memory leak or memory error.

Submission Instructions.

Submit any source/header files with your implementation, as well as a Makefile that builds a target called `pa2`, to Gradescope. Note that to receive points, your submission must work on `eceprog`. (See the syllabus for more details.)
If you choose to do the optional writeup, submit it to the assignment on Brightspace. The writeup can be formatted however you want, but here's a guideline of some questions you may want to think about when writing it:

- What data structures did you choose to use, and why? Are there different choices that would have made a more efficient solution?
- What other factors did you consider? How did they influence your final implementation?
- What, if anything, did you find most challenging about this project? Was there anything in particular you enjoyed/disliked?
- If you took a particularly creative approach to solving this problem, or ended up implementing something we didn't ask for, tell me about it!