# Problem 20: Ghostbusters

| | |
|---|---|
| Source filename: | `ghosts.(cpp|java)` |
| Input filename: | `ghosts.in` |
| Output filename: | `ghosts.out` |

A group of *n* ghostbusters is battling *n* ghosts. Each ghostbuster is armed with a proton pack which shoots a stream at ghost, eradicating it. A stream goes in a straight line and terminates when it hits a ghost.

The ghostbusters decide upon the following strategy. They will pair off with the ghosts, forming *n* ghostbuster-ghost pairs, and then simultaneously each ghostbuster will shoot a stream at his or her chosen ghost. As we all know, it is *very* dangerous to let proton streams cross, and so the ghostbusters must choose pairings for which no two streams will cross.

Assume that the position of each ghostbuster and each ghost is a fixed point in the plane and that no three positions are collinear.

**Claim:** There exists a line passing through one ghostbuster and one ghost such that the number of ghostbusters on one side of the line equals the number ghosts on that same side.

Write a program that pairs ghostbusters with ghosts in such a way that no two proton streams will cross.

**Description of input file:** The 1st line contains *n* ($1 \le n \le 50$), the number of ghosts (as well as the number of ghostbusters). The next *n* lines contain pairs of floating point numbers (separated by a single space) that represent the coordinates of the positions of the *n* ghostbusters. The *n* lines after that contain the floating point numbers (separated by a single space) that represent the coordinates of the ghosts' positions. All floating point numbers will be represented with 1 digit to the right of the decimal point.

The next line will contain another integer, *m* ($0 \le m \le 50$). If the value for *m* is 0, then you have reached the end of the input file. Otherwise, this is the start of a new data set and the next *m* lines contain the coordinates of *m* new ghostbusters followed by *m* lines for the coordinates of *m* new ghosts.

**Description of output file:** For each data set in the input file, write the header: "`Data Set #k`" to the output file (where `k` is the data set number, beginning with `#1`). The next several lines in the output file should report a list of ghostbuster-ghost pairings (one pairing per line) that satisfy the stated strategy for the given data set – using the format: $(x_{ghostbuster},y_{ghostbuster})\text{->}(x_{ghost},y_{ghost})$ (Note: No Spaces!)

The list of ghostbuster-ghost pairings should be sorted in ascending order on the x-coordinates of the ghostbusters. If more than 1 ghostbuster share the same x-coordinate, sort them in ascending order on their y-coordinate.

**Note:** A solution of Ghostbuster-Ghost pairings that satisfy the stated strategy is *not* unique. You are free to select any solution that solves the problem.

**Example Input File** (`ghosts.in`)

```
2
0.0 0.0
3.0 4.5
0.0 3.1
2.5 0.0
3
0.0 4.0
3.0 0.0
-3.0 0.0
0.0 -4.0
1.0 4.0
2.5 -3.5
0
```

**Example Output File** (`ghosts.out`)

```
Data Set #1
(0.0,0.0)->(2.5,0.0)
(3.0,4.5)->(0.0,3.1)
Data Set #2
(-3.0,0.0)->(0.0,-4.0)
(0.0,4.0)->(1.0,4.0)
(3.0,0.0)->(2.5,-3.5)
```