Project #4

Prof. John P. Baugh – CIS 2353 – Oakland Community College – OR

Points: _____________ / 200
Due: December 5, 2018 at 11:59 p.m.

Objectives

- To apply sorting to an interesting problem
- To determine numerical distances

Instructions

Sorting is useful as the first step in many different tasks. The most common task is to make finding things easier, but there are other uses as well. In this case, it will make it easier to determine which pair or pairs of elements have the smallest absolute difference between them.

For example, if you've got the list [5, 2, 3, 4, 1], sort it as [1, 2, 3, 4, 5] to see that several pairs have the minimum difference of 1: [(1, 2), (2,3), (3,4), (4,5)]. The displayed solution should be (1,2), (2,3), (3,4), (4,5).

So, you must solve the following:

Given a list of unsorted integers, ***arrList,*** find the pair (or *pairs*) of elements that have the smallest absolute difference between them.

Method Description

You will create a closestNumbers method that will return an ArrayList of pairs of integers as discussed above. The closestNumbers method takes as a parameter, ***arrList***, an ArrayList of integers.

Input Format

The program will work by reading input files and producing output to the console. The format of the input file is to contain all elements, one per line. Both positive and non-positive integer values are acceptable.

Output Format

The output contains the pair (or pairs, one per line) with the smallest difference.

Example 1

```
-20
-3916237
-357920
-3620601
7374819
-7330761
30
```

```
6246457
-6461594
266854
```

Input for Example 1

```
(-20, 30)
```

Output for Example 1

Example 2

```
-20
-3916237
-357920
-3620601
7374819
-7330761
30
6246457
-6461594
266854
-520
-470
```

Input for Example 2

```
(-520, -470)
(-20, 30)
```

Output for Example 2

Note for Example 2 that the minimal distance is 50, and both pairs in the output have this minimal distance.

Deliverables

- **Create a zip file** of your .java files (in PDF or Word format) and turn in the zip file. Name the zip file "Project3" and D2L will take care of putting your name in it.
- You will also need **screen shots of your program working**, pasted inside of a PDF or Word (.doc or .docx) document (you can create PDF from Word documents using the Save As… option)
- Also, make sure your name is in comments on **each** Java file that you turn in. For example:

```
// Gunner Boomstikkz
// CIS 2353
// Fall 2018
// Prof. John P. Baugh
```