COMP 482 Project 2
Due: Friday October 30 2355

Task: You will write a java program which

- reads a set of data (lattice points - ordered pairs with both coordinates integer) from a file,
- stores that data into a data structure, and
- finds a lattice point which is "most central" (technically there can be ties) to the data set and the total distance to a lattice point which is "most central" to the data using both the L_1 (ie Manhattan) and L_2 (ie normal distance) metrics.

Your program must:

- compile using the command 'javac Project2.java',
- run using the command 'java Project2',
- accept input from a file called input2.txt formatted precisely as described below,
- output results exactly as described below, and
- is reasonably efficient. The L_1 metric can be done in $O(n \lg n)$ (hmm, same time complexity as sorting?), but the L_2 metric may take significantly longer - I'm actually curious what techniques you come up with for solving this second part).

If your submission fails to do any of these things, expect a score no higher than 5/20.

Your program should:

- be properly formatted (eg using proper indentation)
- be structured reasonably (eg dividing the tasks up into methods), and
- be commented appropriately (eg containg your name at or near the beginning of each file and explaining how/why a method works just above the method).

If your submission fails to do any of these things, expect a score no higher than 15/20.

Metrics and Most Centeral: A metric is just a distance function. You should be familiar with the normal L_2 metric $\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$. The L_1 (ie Manhattan) metric is $|x_1-x_2|+|y_1-y_2|$ which represents the distance you would have to travel if you could only move along a grid of north/south or east/west streets.

The "most central" concept I want you to use in this project is the lattice point which minimizes the total distance between itself and the other points. For those looking online, this is not the centroid or most of the other concepts I found. The way I imagine the problem is that I have people at each of the locations who want to have 1 big meeting, but want to minimize the total travel distance.

Consider the first sample input below. The people could meet at many locations including (0,0), (3,0), or (100,100). In these cases, they would have total travel distance 0+1+2+3+4+5+6 = 21 or 3+2+1+0+1+2+3 = 12 or something huge.

Input Format: After downloading your submission, I will place a file called input2.txt into the same directory. Every line of the file will contain two strictly positive integers seperated by whitespace (spaces/tabs) each of which will be one integer lattice point. These are the values you should place into a data structure.

Two sample input files - input2.txt:

First

```
0 0
1 0
2 0
3 0
4 0
5 0
6 0
```

Second

```
0  0
10 0
0     10
```

Output Format: You will output 2 lines. The first line will be an ordered pair which is the location of a "most central" point according to the L_1 metric, a space, and the total distance from all data items to the "most central" point. The second line will be an ordered pair which is the location of a "most central" point according the the L_2 metric, a space, and the total distance from all data items to the "most central" point.

Sample output: For the first sample input above:

```
(3,0) 12
(3,0) 12
```

For the second sample input above:

```
(0,0) 20
(2,2) 19.32084962721683
```

Stray Thoughts:
I will be using a recent version of Java (likely the current version Java SE 15, but if Oracle releases a new version I may upgrade).
You'll be submitting only dot-java files (no class files or input files required or wanted).
You are allowed to use any of the standard features, classes, methods in Java. For example, I expect nearly all students will want to use either an array or java.util.ArrayList and the built in sort routine (either for arrays or ArrayLists). This is allowed.
You can use as many or as few files as you feel appropriate, but the main method should be located in a file called Project1.java. Otherwise the project won't compile/run with the required commands.
Some IDEs default to placing java files into packages. This will likely cause the commands 'javac Project1.java' and/or 'java Project1' to fail. Either use an IDE that does not place java files into packages OR learn your preferred IDE well enough to avoid this issue OR delete any package lines before submission.
Students often decide to change or modify the format of the input or ouput. Sometimes it makes the project easier for them. Other times a student thinks it is an improved design. You may or may not be right, but don't change the input or output format. Doing so will result in your project getting a low score.
It is likely that many students won't read this far. There is no need to let me know you've read this.
There is a very easy way to determine answers in time $O(n^3)$. You should not use this method. You should instead use a method which is $O(n^2)$.
I will likely use the sample input file above while grading, but I'm also likely to use other much larger input files (possibly containing millions of data items, but no more than the JVM limit on arrays $\approx 2^{30}$).
I suggest you finish you project several days in advance. This way you have time and opportunity to ask any last questions and verify that what you upload satisfies the requirements.
Your project should be written and understood by you. Helping or receiving help from others is allowed, but significant shared source code indicates that you either did not write/understand what you submitted or you assisted another in submitting code they did not write or understand.