COMP 482 Project 2: Adjusting Raw Data
Due: 11/8
Points: 30 points possible

Overview: Often raw data needs to be processed/adjusted before analysis. Some cases are obviously necessary (others might be more suspect). For example, when given distance measurements in a mixture of English ft/in and Metric m/cm it would be reasonable to consistantly use one or the other. Similarly, if you expect an exponential relationship between two variables then it would be normal to take the log of the faster growing variable and investigate a linear relationship. Other times the adjustment might be made to compensate for the effects of another variable. A house rated 2/10 might cost more than a house rated 8/10 if the first house is on the beach in Malibu and the second is in rural Manitoba.

You will be given an array of ratings and moderately complicated requirements for the calculation of scores. Think of the scores as the (suspiciously complicated) adjusted values of the ratings. Your program will calculate the scores and sum them. The calculation process will require a simple $\Theta(n^2)$ greedy algorithm which you are to determine.

Details: The input will come from a file called input.txt which will be placed in the same directory as your java file. The first line of the file will have a single value which will be the value of N. The remainder of the file will be N integer rating values seperated by whitespace.

You program will read in this file, place the ratings in an array (or linked list or ArrayList or ...), create a second array to hold the values to be calculated by your algorithm, calculate the values, and print the sum of the calculated values.

The rules for calculating values are: all values must be integer, all values must be at least 1, if rating i is higher than the rating $i-1$ then value i must be higher than value $i-1$, if rating i is higher than rating $i+1$ then value i must be greater than value $i+1$, and values should be as low as possible subject to the previous requirements.

You can discuss the algorithm to be used with anyone and consult any source (books, internet, etc). However, for projects, you are expected to write the code on your own with limited or no assistance from the professor (using Project0.java is permitted), no assistance from others, and limited or no assistance from other sources (books, internet, etc).

Picky, but required specifications: Your project must:

- be submitted via canvas.
- consist of 1 or more dot-java files (no class files, zip files, input files or other files should be submitted). Each file must have your name and which project you are submitting as comments on the first 2 lines.
- not be placed into any package (for the java pedants, it must be in the default package).
- be designed and formatted reasonably (correct indentation, no excessively long lines, no excessively long methods, has useful method/variable names, etc)
- have one file called Project2.java.
- compile using the command 'javac Project2.java'.
- run using the command 'java Project2',
- accept input from a file called input.txt in the same directory as the java file(s) formatted precisely as described above.
- accomplishes the goal of the project. In other words, the output should be the correct answer, computed in the correct way, formatted correctly.
- be submitted on time (early and multiple times is fine).

For each listed item that you fail to follow, expect to lose at least 5 points. In particular, submitting via anything other than canvas will result in a 0.

Sample execution: If input.txt contains the ratings

```
10
4 2 0 3 6 4 5 2 7 1
```

then your program should calculate the values array
3 2 1 2 3 1 2 1 2 1
(it is easy to verify that all values are at least 1 and if a rating is higher than an neighbor then the value is larger than the neighboring value) and output the sum

```
18
```

If input.txt contains the ratings

```
20
40 20 13 94 16
32 27 15 76
43 51 59 71 63 92 30 34
51 84 32
```

then your program should calculate the values array
3 2 1 2 1 3 2 1 2 1 2 3 4 1 2 1 2 3 4 1
and output the sum

```
41
```

Stray Thoughts:
I will be using a recent version of Java (likely the current version Java SE 17, but if a new version is released I may upgrade).
You are generally allowed to use the standard features, classes, methods in Java. For example, I expect nearly all students will sometimes want to use either an array or java.util.ArrayList and the built in sort routine (either for arrays or ArrayLists). This is allowed as long as it doesn't violate a project requirement.
You can use as many or as few files as you feel appropriate, but the main method should be located in a file called Project2.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 Project2.java' and/or 'java Project2' 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.
It is likely that many students won't read this far, but there is no need to let me know you've read this.
I suggest you finish you project several days in advance. This way you have time and opportunity to ask any last questions, you don't get caught by a power outage or similar issue, and verify that what you upload satisfies the requirements.
Your project should be written and understood by you. Helping or receiving help from others to figure out what is allowed/required is fine, but copying code is not. 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. Both are unacceptable.
These project description files are getting excessively long, but I find that failing to state things which are common sense yields complaints.