COMP 482 Project 4: Largest Non-consecutive Subset Of Array
Due: 12/15
Points: 30 points possible

Overview: You'll be given an array of (positive and negative) integers with size N. Your goal is to find the subset of the entries with the largest sum with 1 important restriction: no 2 elements of the subset can be consecutive. You will find an $O(N)$ time dynamic programming algorithm.

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 values seperated by whitespace.

You program will read in this file, place the integers in an array (or linked list or ArrayList or ...), and then use a dynamic programming technique (ie you'll create an array, fill in base cases, fill in the remaining cells using a recurrence). As a hint realize that the optimal solution either includes the last item in the array or it does not include the last item in the array.

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 Project4.java.
- compile using the command 'javac Project4.java'.
- run using the command 'java Project4',
- 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

```
10
9 8 7 -100 3 2 4 1 5 -4
```

then your program should ouput

```
28
```

because 9+7+3+4+5 = 28.

If input.txt contains

```
10
5 7 9 2
-6
-4
12 10 14 15
```

then your program should output

```
41
```

because 5+9+12+15 = 41.

If input.txt contains

```
12
-2 1 7 -3
4 5 6 -2
8 -5 1 2
```

then your program should output

```
21
```

because 7+4+6+8+2=27.

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 Project4.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 Project4.java' and/or 'java Project4' 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.