CSE 174 Fall 2018

PROGRAM #11: 30 points – Due November 10, by 11:59 p.m.

Outcomes:

- Write programs that use arrays
- Format and comment source code that adheres to a given set of formatting guidelines

Scoring:

At a bare minimum, the program you submit must have the assigned source code, and your source code must compile and run without crashing.

- If you submit source code that does not compile, your score will be zero.
- If you submit source code that roughly resembles the requirements and it compiles, but it crashes under normal operating conditions (nice input from the user), your score will be reduced by 75%.
- Deductions will be made for not meeting the usual formatting requirements (commenting, indentation, appropriate variable names, and so on.

	Full credit	Partial credit
Solve the specified problem (20 points)	Program correctly solves the specified problem for any number of lockers	There are errors in the solution.
Use appropriate programming techniques (5 points)	Program correctly uses appropriate data types (such as a boolean array to store the state of the lockers) and uses multiple short methods to break the larger problem into smaller parts.	Inappropriate data types are used, and/or the program should have been broken into smaller methods.
Format console output, Comment, file names, short methods, etc. (5 points)	You formatted output as specified, including aligning numbers. Your code has comprehensive comments	Screen output does not match specifications, or not enough comments.

Background:

Imagine a school with 10 students, each with her own locker. At the beginning of the day, all 10 lockers are closed. Then, the 10 students do the following:

- Student #1 opens all lockers
- Student #2 goes through and changes lockers 2, 4, 6, 8, and 10 (so, closes those lockers)
- Student #3 goes through and changes lockers 3, 6, and 9. That is, changes all lockers that are a multiple of 3 (closing those that were open, and opening those that were closed)

- Student #4 goes through and changes all lockers that are a multiple of 4.
- and so on up to and including student #10.

The big question is: after all ten students have gone through, which lockers are open?

Requirements:

You are to write a program that can solve this for any number of students (the number of students will always be the same as the number of lockers), showing the lockers along the way, and then listing which lockers are open after each student. You should match the output format shown below as closely as possible.

A few specifics:

- Name your class **Program11**.
- Number of lockers should be a number **bigger than 2**, otherwise you need to display the "invalid input" message and ask the user to re-enter the number.
- When printing the ten lockers, print them at the end of each student going through and changing the lockers. So, the first stage that gets printed is the state of the lockers after person 1 has gone through and opened them all.
- Use `O` to represent an open locker, and `X` or `-` to represent a closed locker. Do not put spaces between the lockers. So, for example, after stage 3, we see `OXXXOOOXXX` or `O---OOO---` which indicates that lockers 1, 5, 6, and 7 are open.
- At the end of all the stages, list which lockers remain open.
- Use a boolean array to keep track of which lockers are open and which are closed.
- Break this problem into small single-purpose methods. Any method should not be longer than **12 lines** long (comments, empty lines, brackets are not included).
- The user has the option of showing the lockers after each stage. This is useful for larger numbers. But the program will always show the comma-separated list of open lockers.
- When the program has run once, the program prompts the user to see if the user wants to run the program again. If the answer is yes, the program needs to be run again.
- Submit Program11.java.

Sample run:

Notice that 10 students means 10 lockers and 10 stages. After student #1, each locker is open. After student #2, the evens are closed, and so on. After student #10, lockers 1, 4, and 9 are still open. In sample run 2, notice that the user did not want to see the stages, and so only the list of open lockers was printed at the end.