Locker Puzzle

Pseudo-code:

1. Create Boolean array of lockers with 100 elements
2. Make every Boolean value true in the array for open lockers
   1. Use a for loop to go through each array element.
3. Nested for-loop
   1. Start with second student and closes every other locker
      1. The pattern is the locker number you are the number of lockers you skip
      2. You continue this going through students 2-100
      3. Inverse locker Boolean value as you continue through
4. Iterate through each value and only print lockers that remain open after all 100 students are done.

Test Plan:

When solving this problem by hand you find out that all numbers that are perfect squares are the ones that remain open. There is only one expected result.

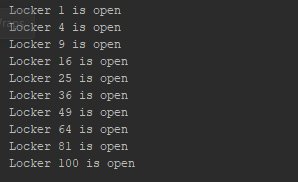
Those numbers are 1, 4, 9, 16, 25, 36, 49, 64, 81, 100.

|  |  |  |  |
| --- | --- | --- | --- |
| **Cases** | **Input** | **Expected Result** | **Actual Result** |
| 1 | NA | Locker 1 is open  Locker 4 is open  Locker 9 is open  Locker 16 is open  Locker 25 is open  Locker 36 is open  Locker 49 is open  Locker 64 is open  Locker 81 is open  Locker 100 is open | 1, 4, 9, 16, 25, 36, 49, 64, 81, 100 |

UML:

|  |
| --- |
| **Class Name:**  **LockerPuzzle** |
| +boolean[] lockers |
| **+** for (int i = 0; i < 100; i++)  **+**for (int s = 2; s <= 100; s++)  for (int locker = s - 1; locker < 100; locker += s)  +for (int i = 0; i < 100; i++) |

Code Screenshot:



FlowChart:

