ICA-8 Testing (individual)

Using GitHub, Java, and JUnit 5 (Jupiter), create a Test-Driven Development (TDD) program for counting urinals. 😊 The important thing here is the process, not the problem space. Simple tutorial -- [JUnit 5 Tutorial - HowToDoInJava]().

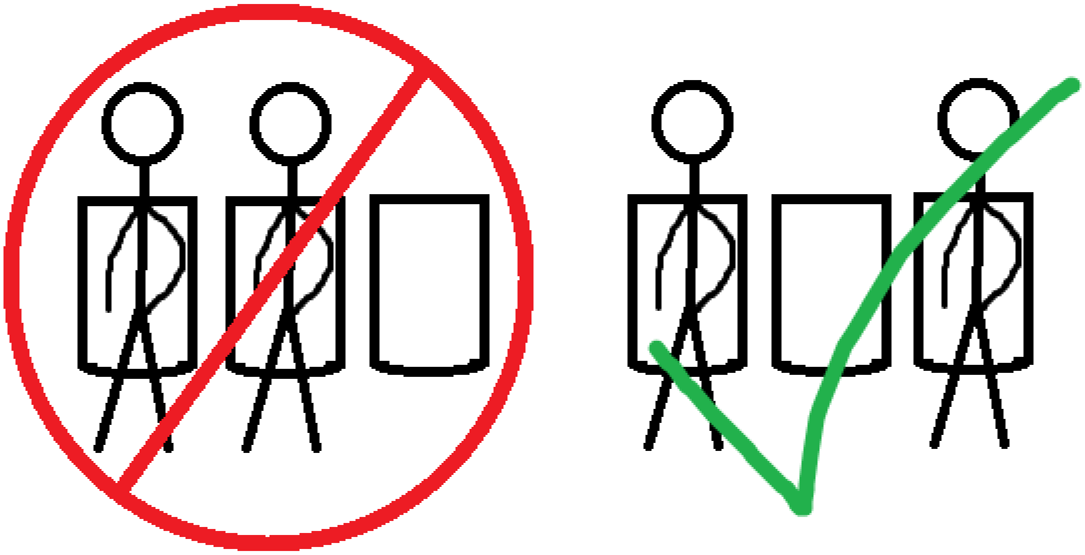
1. Create a **private** Workspace on GitHub. Make sure **ser-515-mjf** is invited to the workspace.
    a. Create a readme.md file. Describe the problem.
2. Create a shell of the class *urinals.java* with no functions except main. **Make sure your name is in the file as author**!
3. Add your (first) function skeleton.
    a. getString()? openFile()? countUrinals()? You decide the function names and the order to implement.
    b. The skeleton should have the function heading, i.e.,

```
Boolean goodString( String str ) {  // checks to see if valid string
     System.out.println (“Not yet implemented”);
     return true;
}
```

4. In your *urinalsTest.java*, create a JUnit test that fails. Add your name to the test string, i.e., `System.out.println("====== Mike Findler == TEST TWO EXECUTED =======");` (Eliminate the `System.out.println` (“`Not yet implemented`”); statement in your target source code.)
5. Correct your code to pass the test. Minimally change code, add nothing more than the amount needed to pass test.
6. Commit to GitHub! **Graders will be looking for these micro-commits**.
7. **Repeat 4-6** until that function test set is complete.
8. *Normally, you should commit here, not 6.*
9. Go back to **step 3** and create your next function.

How many urinals are free?

In men's public toilets with urinals, there is this unwritten rule that you leave at least one urinal free between you and the next person peeing. For example, if there are 3 urinals and one person is already peeing in the left one, you will choose the urinal on the right and not the one in the middle. That means that a maximum of 3 people can pee at the same time on public toilets with 5 urinals when following this rule (Only 2 if the first-person pees into urinal 2 or 4).



Your task:

You need to write a function that returns the maximum of free urinals as an integer according to the unwritten rule.

Examples

10001 returns 1 (10101)
1001 returns 0 (1001)
00000 returns 3 (10101)
0000 returns 2 (1001)
01000 returns 1 (01010 or 01001)
011 returns −1

Input

A String containing 1s and 0s (Example: 10001) (1 <= Length <= 20). This string can come from the keyboard, or a file called *urinal.dat*. The program will continue processing until a -1 or <eof> is reached. (This means you handle unusual input without crashing.)

A one stands for a taken urinal and a zero for a free one.

Output

If input is the keyboard, print the results to the screen. If input is from a file, output to *rule.txt*. If the file *rule.txt* already exists, increment a counter, and rename the file using the following rule pattern: *rule1.txt*, *rule2.txt*, etc. The output is JUST THE NUMBER of free urinals. Given the above output, your rule.txt file should contain:

1
0
3
2
1
-1

Submit

Submit a zip file with your java files and a text file with your name and GitHub repository link for the assignment. In the Canvas comments section also paste the GitHub repository address.

Note

When there is already a mistake in the input string (for example 011), then return −1. Should all errors return –1? (No, but this homework is okay. If you would like to have different error messages, note them in the GitHub Readme file.)

What are your equivalency tests? Are my examples enough? (NO 😊)

Have fun and don't pee into the wrong urinal! 😉