Brandeis University COSI 12B, Spring 2019

PA0: Warmup Problems

Due date: Monday, January 21, 2019, at 11:55 PM

Late due date (20 point penalty): Tuesday, January 22, 2019, at 6:30 PM

Overview

Eclipse, JUnit tests, abstract classes, recursion... these are just some of the things we're excited to teach you about in this rollercoaster of a course! First, however, there are some prerequisites we have to lay down, and what better way to enforce them than with the very first PA of the class?

This PA isn't meant to teach you any new Java skills, it's purely meant to test the skills you should know as you're coming into this course. This will test Java concepts such as **strings, for loops, and arrays**.

IMPORTANT - A note on difficulty

These problems were **designed to be easy** for a student out of COSI 11A. Some of them may make you think a bit, but they should all be attainable without too much work. If you find yourself struggling with these problems, **please talk to Pito or a TA about it immediately** - we won't judge you harshly for it, we just want to make sure you're fully prepared with these basic concepts when we pick up the pace and start introducing new concepts.

How do I do this PA?

The format of this PA is slightly different from the PAs you've done in 11A, but it still takes place in the same text editor you know and love. Here's what you have to do:

1. Download the `PA0Problems.java` and open it in your favorite text editor. This file contains four blank static methods, one for each problem we're asking you to do. **You'll be writing the code for those four methods** – more on that soon.
2. When you think you're done, compile and run that Java file in order to test your code! We've given you a few examples of expected inputs and outputs for each method, so you can use those examples to gauge

how your code is doing. Note that **those examples are not exhaustive** - as in, you will be graded on example cases other than the ones we gave you, so you should get creative and come up with your own test cases!

3. When you feel totally finished and you're happy with the results of testing your code, submit your `PA0Problems.java` file to LATTE.

The methods you have to write

Here are the four blank methods contained inside `PA0Problems.java`, and what we expect each one to do:

`int charCount(String str, char c)`

This method will take in a String, `str`, and a character, `c`, and return how many times the character appears within the given string. It should return 0 if the character does not appear in the string at all.

Examples:

- `charCount("banana", 'n')` should return 2, because the letter 'n' appears twice in the word "banana".
- `charCount("banana", 'q')` should return 0, because the letter 'q' does not appear in the word "banana".

`double calculator(int a, int b, String op)`

For this method, you'll be implementing a basic calculator that takes in two integers, `a` and `b`, and returns the mathematical result of `a [op] b`. `op` will be a string with one of four values: `ADD`, `SUB`, `MUL`, and `DIV`, representing addition, subtraction, multiplication, and division, respectively.

While this method takes two integers, it returns a double, so in division problems where the quotient has decimal places, we'll be expecting those to be returned too! You may assume that this method will never be asked to divide by zero, and it will never be asked to generate a number that exceeds the maximum possible value of a double.

Examples:

- `calculator(3, 4, "ADD")` should return 7, because 3 + 4 = 7
- `calculator(3, 4, "DIV")` should return 0.75, because 3 ÷ 4 = 0.75

`int[] arrayBuilder(int len, int val)`

This method takes in two integers, `len` and `val`, and returns an array of length `len`, in which every element of the array is set to `val`. You may assume that `len` will never be a negative integer.

Examples:

- `arrayBuilder(3, 7)` should return the array `[7, 7, 7]`, a length-3 array where each element is 7.
- `arrayBuilder(0, 2)` should return the empty array, `[]`.

`String messageTriangle(String str)`

This method takes in a string, `str`, and returns a string that consists of `str` being built up, one letter at a time, and then shrinking one letter at a time.

Examples:

- `messageTriangle("ab")` should return the following string:

```
a
ab
a
```

- `messageTriangle("pito")` should return the following string:

```
p
pi
pit
pito
pit
pi
p
```

Note that we're expecting that entire output in a single string. Recall that `nameOfString += '\n'` can be used to add a line break to a string!

But wait, don't I have to write a main method?

Great question! You'll notice that midway down the `PA0Problems.java` file, we've written a main method for you, which acts as an interactive console for you to call any of the four methods you've written and prompts you for their parameters. Since we've taken care of this for you, the only things you have to write are the four

methods we listed above.

Another thing to recall about this setup is that it means *the code you write is never asked to print anything* - you may add any print statements you want for debugging purposes, but **the only values that matter in the end are the values that your methods return**.

As a reminder, you can run the Java file as follows:

1. Open a terminal window to the folder where the file is.
2. Run `javac PA0Problems.java`. If the command runs without outputting any errors, then your code is free of syntax errors and has compiled successfully.
3. Run `java PA0Problems` to start the program.

Here's an example of how you'd run the program to test `charCount`, with user input in bold. Since we've already written the main method, all you need to do to get the correct answer in this example is write the code for `charCount`. You can test any of the four methods like this, by entering its name and then entering all the parameters when prompted.

```
> java PA0Problems

Methods: charCount | calculator | arrayBuilder | messageTriangle

Method to run (or "exit"): charCount

Testing: int charCount(String str, char c)

    str: banana

    c: n

charCount("banana", n) = 2

Methods: charCount | calculator | arrayBuilder | messageTriangle

Method to run (or "exit"): exit

> _
```

Submission and Grading

Test Grade

When you're satisfied with your code, submit the completed file to LATTE. **70%** of your grade will be based on

the output of an automated testing script, which will test your code against the examples we gave you as well as some other test cases.

Note that even though you know that the examples will comprise some of what you're going to be graded on, this is **not** an excuse to write code specifically tailored to those cases! This is known as *hardcoding*, and it's generally regarded in the computer science community as "not cool."

For instance, consider the following code, written by a student who saw that our examples for `charCount` ask for the frequencies of n and q in the string "banana":

```
if(str.equals("banana") && c == 'n') {
    return 2;
} else if(str.equals("banana") && c == 'q') {
    return 0;
}
```

While our testing script would report that this student's implementation of `charCount` passes our tests for those inputs, the TA reading their code would give them a substantial penalty for writing something like that - the student clearly didn't solve the problem we asked them to solve.

One-on-Ones

Over the week starting on Monday, January 21st, you will be assigned a TA who will email you asking to schedule a one-on-one meeting with you to discuss your work on the PA. Your "meeting grade," which in this case simply means scheduling a meeting and showing up for it, will comprise the remaining **30%** of your grade.

If you haven't received an email from any TA by **12:00 noon on Wednesday, January 23rd**, you must proactively inform Pito or any TA and we'll work to put you in touch with your assigned TA. Otherwise, we'll assume you've been getting our emails and are choosing to not schedule a meeting.

Welcome to COSI 12B!