Lab 06 – Input Validation, Loops

Part 1: Group Brainstorm (NO computers during this time)

Good programmers think before they begin coding. Part I of this assignment involves brainstorming with a group of peers with absolutely no computers to talk about a strategy for solving this week's lab. Breakup into groups based on your seating (3-4 people per group) and brainstorm about how to solve the problem in Part 2 below. Make sure everyone understands the problem and sketch out potential ways to move toward a solution. Write up a detailed description of your approach in English / pseudocode (English that is structured like a program). This should be 1-2 paragraphs in length (enough to convince the grader that you have a plan for moving toward a solution). You may find it helpful to look over the required readings for this week. Make sure to get the last names of those you have worked with – you will need to provide it in your write-up for Part 2.

Include in your write-up which course topics you think will be useful for the lab and a detailed plan for completing the exercise. You will want to discuss an algorithm for computing the reverse of an integer using only mathematical operations.

Part 2: Submit Individual Brainstorm (You can now use a computer)

Login to eLC and submit a version of your group's brainstorm, **written in your own words** under the Assignment Dropbox titled "Lab 6 Brainstorm". *Note this is different than the Lab 6 Dropbox where you will submit your .java file for this assignment. Feel free to add any additional information that was not specified to your write-up. We prefer that you submit the brainstorm before the end of the first lab period for the week. However, you can submit until **Tuesday night at 9PM**.

Note: Brainstorms that are submitted without a student attending the lab will not be graded unless the student provides a documented excuse to the graduate TA.

Introduction

In this lab, the user will enter an integer input and your program will determine whether or not the given input is a *palindrome*, and if the given input is a *palindrome*, then your program will compute a summation involving the *palindrome*. For this lab, **`loop`** statements will be needed.

Lab Objectives

By the end of the lab, you should:

- validate user input and account for invalid user input
- use loops to repeatedly perform calculations
- use the modulus operator (%) to calculate remainders

Prerequisites

The lab deals mainly with material from Chapter 4 (loops and debugging loops) but builds on material learned in Chapters 2-3.

What to Submit

The file **`Palindrome.java`** should be submitted to eLC for grading.

Palindromes

A string (or any other sequence of characters) is called a *palindrome* if it reads the same way in reverse as it does when read normally. That is, a string is a palindrome if it is identical to the string obtained by writing its characters in reverse. In the same way, a positive integer is a palindrome if reversing the order of the digits yields the same integer.

Write a program called **`Palindrome`** that allows the user to enter a positive integer $n > 0$ and then determines whether n is a palindrome. If the user enters a negative integer or zero, then the program should display an error message and terminate.

Your program should determine whether n is a palindrome by obtaining the reverse of n. In order to obtain the reverse, you should store the variable n as an int and use only mathematical operations (no String methods allowed) to reverse the integer. Discuss this in your group brainstorm & provide an outline in your write-up of how this can be done.

Requirement

In order to receive credit, only arithmetic operations can be used to compute the reverse of a number.

Summation involving a palindrome

Once we've determined whether the number is a palindrome, write another loop that adds up all of the integers from 1 to that number (the summation). For instance, if the palindrome is 11, write a loop that calculates:

$$(1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + 11) = 66$$

Output the result of that calculation as seen in the samples below. Please note that if the number entered is not a palindrome, then you should not compute the summation.

Sample Input and Output

These are sample runs of the program. Your output should be consistent with what is shown here.

```
Please enter an integer > 0: -190
Sorry, you must enter an integer greater than zero.
```

```
Please enter an integer > 0: 1991
The integer 1991 is a palindrome.
The sum of the numbers from 1 to 1991 is 1983036
```

```
Please enter an integer > 0: 9
The integer 9 is a palindrome.
The sum of the numbers from 1 to 9 is 45
```

```
Please enter an integer > 0: 11
The integer 11 is a palindrome.
The sum of the numbers from 1 to 11 is 66
```

```
Please enter an integer > 0: 12562
The integer 12562 is not a palindrome.
```

eLC Submission and Grading

After you have completed and thoroughly tested **`Palindrome.java`** submit it to ***eLC*** in order to receive credit for the lab. Always double check that your submission was successful on ***eLC***!

The lab will be graded according to the following guidelines.

- A score between 0 and 100 will be assigned.
- If the source file(s) are not submitted before the specified deadline's late period ends (48 hours after the deadline) or if they do not compile, then a grade of 0 will be assigned.
- If the required comment for all labs describing the program and the academic honesty statement is not included at the top of the file, then 10 points will be deducted. Note: this required comment can be found in Lab 02.
- The program will be evaluated using the inputs below and additional inputs. For each test case, the output must be correct in order to receive credit for that test case.