Assignment 6 – Spring 2021

Due Date: by Friday May 7, 2021 11:59PM
How to submit: upload JAVA files to Blackboard

Please note:

- ✓ This is an individual assignment; please do your own work. Sharing and/or copying code in part or whole with/from others will result in a grade of 0 and disciplinary actions for all involved parties. If you run into problems and have done your best to solve them, please talk to me during office hours or by e-mail.
- ✓ There is a 25% grade deduction for every day the assignment is late unless prior permission is granted.

Preamble

In this assignment we will work with Java's Lambdas and Streams as an alternative to traditional procedural programming. None of the required work should use a *FOR*, *WHILE*, or *DO … WHILE* loops; you can only use functional programming reductions and intermediate operations.

There is only one file to complete for this assignment. Class *StackOperations* works with the provided test code. Update the methods of *StackOperations* as follows:

1. *toQueue*, converts a stack to a queue. Use the *LinkedList* class as your queue instance.
2. *reverseStack*, returns a new stack with items reversed from the original stack.
3. *sumBetween*, returns the sum of values in the Stack between the specified positions. Make sure the start and end positions are within the stack's valid range. If they're not, return -1
4. *readNumericFromFile*, the function returns a Stack containing the numeric characters found in the input file. Print *IOException* to the standard output. The function will:
   a. Read each line from the file
   b. For each line, convert to a character array
   c. For each character array, filter non-numeric digits
   d. For each numeric character, push on the stack

Hint: the following statement will print the characters of the string *line* to the standard output each on separate line.

```
"abc0123!@#$".chars()
                .filter(Character::isDigit)
                .mapToObj(Character::toChars)
                .forEach(System.out::println);
```

Grading:

Item	Points
toQueue	20
reverseStack	20
sumBetween	20
Handle errors	
readNumericFromFile	40
Exception	
	100

Figures:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ZYXWVUTSRQPONMLKJIHGFEDCBA
-1
10
15
746968613296025012334308579763599323945489576466873874973739536876471214130686128541071387618469739466238294740212639
850307443056394174638485348150640839113596649642679641614284529530168891495787839534119710777194146049645627362076273
496676863627220431117657162321840111016098347274885802004790632913916515863314287310416121347048482719400933221628145
565474279445106064708694092194286661285555661895345794132145076887788923211442313118672890488779408477011092231449003
815999029602505277198082699440866135327408547433669550494939998079324453668188171356249114740674647728598984509779502
240909013706497960250975249283281612799476010209484887095315979270091438480855263650181317532033797248082477032840842
817326131469678248582354176592473382170218443523312310782608678067847488045371509019681949914618363350531594205206581
870624555443925855291308452060l0
```

Figure 1: Test file’s output