CS 304 Homework Assignment 3

Due: 11:59pm, Thursday, February 20th

This assignment is scored out of 61. It consists of 6 questions. The first 5 questions are to be completed on D2L as an online quiz. There is a programming question and you will need to put all your Java programs (`*.java`) as well as output files for this question in the folder named LastName_FirstName_CS304_HW3. Zip this folder, and submit it as one file to Desire2Learn. Do not hand in any printouts. Triple check your assignment before you submit. **If you submit multiple times, only your latest version will be graded and its timestamp will be used to determine whether a late penalty should be applied.**

Short Answers

Complete the quiz for this homework on D2L by the due date. You might see there is a time limit of 120 minutes on this quiz but it is not enforced so you can ignore it. Before you complete all questions, DO NOT submit! Doing so will prevent any further changes to the answers. You can save your answers for as many times as you want before submission.

Programming Questions

(33pts)

a. Completing the `RecursiveMethods` class

In the **`RecursiveMethods`** class, you are required to implement the following methods using recursive solutions (**no looping statements**):

`sumSquaresRec(int[] A, int pos)` – This method takes an integer array as well as an integer (the starting index) and returns the sum of the squares of the elements in the region between indices **`pos`** and **`A.length-1`**. **Your implementation should NOT change the content of the array that is passed into the method.**

`upperStackRec(CharStack s)` – This method takes a character stack and converts all lower case letters to upper case ones. **Do NOT create any auxiliary data structure, including but not limited to array(s), queue(s), and list(s). Primitive variables are okay.**

`reverseStringRec(String s)` – This method reads a string and returns the string in the reversed order.

`reverseListRec(LNode head)` – This method takes a reference to the head of a linked list and returns the reference to the head of the linked list in the reversed order. **Do NOT create any additional nodes or any other auxiliary structures. Do NOT alter the data field of any nodes.**

Note that you are only supposed to touch the above methods. You are NOT allowed to create any other methods, instance variables, or make any changes to methods other

than these four methods or files other than "`RecursiveMethods.java`". Points will be taken off if you fail to follow this rule.

b. Code Testing

You are provided with a test driver implemented by "**`TestRecursiveMethods.java`**" (**Do not make any changes to this file!**) so there is no need to write your own.

Once you have completed the methods, you can run the test. You should create a plain text file named "**`output.txt`**", copy and paste the output (if your code crashes or does not compile, copy and paste the error messages) to this file and save it.

Grading Rubrics:

Code does not compile: -10
Code compiles but crashes when executed: -5
Changes were made to things other than the required methods: -5
`sumSquaresRec` was implemented in a non-recursive way: -4
`sumSquaresRec` changes the contents of the array parameter: -3
`sumSquaresRec` does not make use of its return value: -3
`upperStackRec` was implemented in a non-recursive way: -8
`upperStackRec` uses auxiliary data structures: -4
`reverseStringRec` was implemented in a non-recursive way: -8
`reverseStringRec` does not make use of its return value: -4
`reverseListRec` was implemented in a non-recursive way: -8
`reverseListRec` uses auxiliary data structures: -4
`reverseListRec` alters the data field of linked list nodes: -6
Has output file: 5
Code passes 14 test cases: 28 (each test case worth 2 points)

Sample output:

```
Test 1: sumSquares(10) ==> [Passed]
 Expected: 100
 Yours: 100

Test 2: sumSquares(10, 20, 30, 40, 50, 60) ==> [Passed]
 Expected: 9100
 Yours: 9100

Test 3: upperStackRec("a" (from top to bottom)) ==> [Passed]
 Expected: "A" (from top to bottom)
 Yours: "A" (from top to bottom)

Test 4: upperStackRec("p?7" (from top to bottom)) ==> [Passed]
 Expected: "P?7" (from top to bottom)
 Yours: "P?7" (from top to bottom) Yours: "cba"

...
```

```
Test 13: reverseListRec(20->30->40) ==> [Passed]
 Expected: head->40->30->20->null
 Yours: head->40->30->20->null

Test 14: reverseListRec(90->80->70->60->50->40) ==> [Passed]
 Expected: head->40->50->60->70->80->90->null
 Yours: head->40->50->60->70->80->90->null

Total test cases: 14
Correct: 14
Wrong: 0
```