

CS 304 Homework Assignment 3

Due: 11:59pm, Thursday, October 12th

This assignment is scored out of 65. It consists of 6 questions. When you submit, you are required to create a folder with your name (Last name first, then First name), CS304, HW3, e.g., LastName_FirstName_CS304_HW3. Type your answers into a text file (**only .txt, .doc, and .pdf file formats are accepted**) and save it in this folder. Put all your Java programs (***.java**) as well as output files in the same folder. 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

P1. (11pts, 1pt each) Q12 on page 149

P2. (5pts) Q43 on page 157

P3. (4pts) Q50 on page 159

P4. (4pts, 2pts each) Q4 – a, b on page 203 (You don't need to answer part c.)

P5. (6pts, 2pts each) Q5 on page 203

Programming Questions

P6. (35pts + 12pts bonus)

a. Completing the RecursiveMethods class

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

`largestRec(int[] arr, int pos)` – This method takes an integer array as well as an integer (the starting index) and returns the largest number in the array.

`sumRec(int[] arr, int pos)` – This method takes an integer array as well as an integer (the starting index) and returns the sum of the values in the array.

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

The following method is for bonus points so its implementation is not mandatory:

`reverseListRec(INode 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.

Note that you are only supposed to touch the above four 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". You should also NOT change the content of the array that is passed into each method. 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

largestRec was implemented in a non-recursive way: -5

largestRec changes the content of the array parameter: -3

largestRec does not make use of its return value: -3

sumRec was implemented in a non-recursive way: -5

sumRec changes the contents of the array parameter: -3

sumRec does not make use of its return value: -3

reverseStringRec was implemented in a non-recursive way: -5

reverseStringRec does not make use of its return value: -3

Has output file: 5

Code passes 10 test cases: 30 (each test case worth 3 points)

BONUS:

reverseListRec was implemented in a non-recursive way: -12

Code passes 4 bonus test cases: 12 (each test case worth 3 points)

Sample output:

```
Test 1: sum(10) ==> [Passed]
```

```
Expected: 10
```

```
Yours: 10
```

```
Test 2: sum(10, 20, 30, 40, 50, 60) ==> [Passed]
```

```
Expected: 210
```

```
Yours: 210
```

```
Test 3: largest(10) ==> [Passed]
```

```
Expected: 10
```

```
Yours: 10
```

```
...
```

Test 9: reverseStringRec("abc") ==> [Passed]

Expected: "cba"

Yours: "cba"

Test 10: reverseStringRec("Hello, Data Structures!") ==> [Passed]

Expected: "!serutcurtS ataD ,olleH"

Yours: "!serutcurtS ataD ,olleH"

Total test cases: 10

Correct: 10

Wrong: 0

===== Bonus Question =====

Test 1: reverseListRec(null) ==> [Passed]

Expected: head->null

Yours: head->null

Test 2: reverseListRec(20) ==> [Passed]

Expected: head->20->null

Yours: head->20->null

Test 3: reverseListRec(20->30->40) ==> [Passed]

Expected: head->40->30->20->null

Yours: head->40->30->20->null

Test 4: 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 bonus test cases: 4

Correct: 4

Wrong: 0