

CSCI 2110 – Data Structures in Java

Lab exercise LabN2

- Deadline: Friday Jan 22nd at 5pm
- Please first read the assignment check list available in the following link. If you do not follow the check list, you lose many points unfortunately.
<https://web.cs.dal.ca/~nourashr/teaching/cs2110/assignments/assignmentchecklist/>
- If you have any question regarding the questions, please send an email to the instructor or TAs.
- Please submit your work on the Moodle portal:
<https://courses.cs.dal.ca/login/index.php>
- You probably need to review lecture slides which are available in the following link. Feel free to use the codes of slides.
<https://web.cs.dal.ca/~nourashr/teaching/cs2110/lectures/>
- We follow Dalhousie's policy of plagiarism.
http://www.dal.ca/dept/university_secretariat/academic-integrity.html

In the following questions, you can use the implementations of the *SinglyLinkedList* and *DoublyLinkedList* classes provided in the slides.

Q1) (25 points)

Implement the insertion sort algorithm for a doubly linked list of type Integer. Test your method using five unsorted linked lists. Print the lists before and after sort.

```
public static void inserionSort (DoublyLinkedList<Integer> L){...}
```

Q2) (25 points)

Given two sorted doubly linked lists L1 and L2 of type Integer, write a method to merge L2 in L1. The output list L1 remains sorted. The list L2 remains unchanged.

You cannot concatenate L2 to L1 and then sort the concatenated list. Test your method by three pairs of lists. Print the lists before and after merge.

```
public static void mergeDLists (DoublyLinkedList<Integer> L1,  
DoublyLinkedList<Integer> L2){...}
```

Q3) (25 points)

You have two numbers each represented by a singly linked list of Integer, where each node contains a single digit. The digits are stored in reverse order, such that the first digit is at the head of the list. Write a method that adds the two numbers and returns the sum as a singly linked list. Test your method by five examples. You need to print the sum linked lists.

```
public static SinglyLinkedList<Integer> addSLists (SinglyLinkedList<Integer> L1,  
SinglyLinkedList<Integer> L2){...}
```

Input: (3 → 1 → 5) + (5 → 9 → 2) [513 + 295]

Output: 8 → 0 → 8 [808]

Q4) (15 points)

Implement a method to find the n^{th} to the last element of a singly linked list. Test your method by creating a list and five different values for n. You should print the output. Assumption: The minimum number of nodes in the list is $n+1$.

```
public <E> findntoLast (SinglyLinkedList<E> L1, int n){...}
```

Example: 3 → 1 → 5 → 4 → 324 → 121 → 321

3rd to the last element is: 4

5th to the last element is: 1

Q5) (10 points)

What is the difference between Comparator¹ and Comparable² interfaces in Java? By giving a sample program, explain when we need to implement Comparator and when we need to implement Comparable.

References:

[1] <https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html>

[2] <https://docs.oracle.com/javase/7/docs/api/java/lang/Comparable.html>