Data Structures & Applications
Spring 2020
Lab 02 – Single, Double, Circular Linked List

Instructor: Saif Hassan **Date: 28th September 2020**

READ IT FIRST

Prior to start solving the problems in this assignment, please give full concentration on following points.

1. WORKING – This is individual lab. If you are stuck in a problem contact your teacher, but, in mean time start doing next question (don't waste time).
2. DEADLINE – One Week after assigned this (Deadline is mentioned on LMS).
3. SUBMISSION – This assignment needs to be submitted in a soft copy.
4. WHERE TO SUBMIT – Please visit your LMS.
5. HOW TO SUBMIT – rename all .java/.cpp files as task number and make .zip/.rar file of all files, name it as 000-00-0000_Lab02.zip.

KEEP IT WITH YOU!

1. Indent your code inside the classes and functions. It's a good practice!
2. It is not bad if you keep your code indented inside the loops, if and else blocks as well.
3. Comment your code, where it is necessary.

Read the entire question. Don't jump to the formula directly.

Double LinkedList

Note: Keep this code with you till the course ends.

Task 01: (Double Linked List)

Understand provided code and implement all required methods (with all possible exceptions) in DoubleLinkedList

Node.java

```
1.  public class Node {
```

```
2.
3.      String name;
4.      Node prev, next;
5.
6.      Node (String name)
7.      {
8.          this.prev = null;
9.          this.next = null;
10.         this.name = name;
11.     }
12. }
```

DoubleLinkedList.java

```
1.  public class DoubleLinkedList {
2.
3.      Node head;
4.
5.      // Add node with name in beginning of linkedlist, name as param
6.      public void insertAtBeginning(String name)
7.      {
8.
9.      }
10.     // Add node in beginning of linedlist, node as param
11.     public void insertAtBeginning(Node node)
12.     {
13.
14.     }
15.     // Add node in end of linedlist, name as param
16.     public void insertAtEnd(String name)
17.     {
18.
19.     }
20.     // Add node in end of linedlist, node as param
21.     public void insertAtEnd(Node node)
22.     {
23.
24.     }
25.     // Add node after name which is provided as param , name and node as params
26.     public void insertAfterName(String name, Node node)
27.     {
28.
29.     }
30.     // Add node before name which is provided as param , name and node as params
31.     public void insertBeforeName(String name, Node node)
32.     {
33.
34.     }
35.
36.     // Make double linkedlist as Circular Double LinkedList
37.     public void makeCircular()
38.     {
39.
40.     }
```

```
41.
42.     // Print all the nodes in linkedlist, make sure it works on circular double linkedl
    ist
43.     public void printAll()
44.     {
45.
46.     }
47.     // Test the class
48.     public static void main(String[] args) {
49.         // Test all above methods
50.
51.     }
52.
53. }
```

Task02

In previous labs, you have designed single/double linkedlist with all possible common methods with only head.

Now your task is to implement following methods (Single/Double LL) but this time you have to make another variable say **`tail`** for accessing last element directly.

- All types of methods for inserting (Beginning, End)
- All types of methods for removing (Beginning, End)

Compare these methods with those which were designed without **`tail`**.

Task03

Design a method that takes head as param and detect whether linked list contains cycle or not? Cycle exists in a linked list if any node is visited twice while traversing whole traversing.