Sukkur IBA University
Department of Computer Science

DATA STRUCTURES
Lab02 – Single, Double, Circular Linked List

Saif Hassan

READ IT FIRST

Prior to start solving the problems in these assignments, 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 –

3. SUBMISSION – This assignment needs to be submitted in a soft copy.

4. WHERE TO SUBMIT – Please visit your LMS.

5. WHAT TO SUBMIT – .docx and pdf file.

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

<u>**Note: Keep this code with you till the course ends.**</u>

Task 01: (Double Linked List)

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

Node.java

```java
public class Node {

    String name;
    Node prev, next;

    Node (String name)
    {
        this.prev = null;
        this.next = null;
        this.name = name;
    }
}
```

DoubleLinkedList.java

```java
public class DoubleLinkedList {

    Node head;

    // Add node with name in beginning of linkedlist, name as param
    public void insertAtBeginning(String name)
    {

    }
    // Add node in beginning of linedlist, node as param
    public void insertAtBeginning(Node node)
    {

    }
    // Add node in end of linedlist, name as param
    public void insertAtEnd(String name)
    {

    }
    // Add node in end of linedlist, node as param
    public void insertAtEnd(Node node)
    {

    }
    // Add node after name which is provided as param , name and node as params
    public void insertAfterName(String name, Node node)
    {

    }
    // Add node before name which is provided as param , name and node as params
    public void insertBeforeName(String name, Node node)
    {

    }

    // Make double linkedlist as Circular Double LinkedList
    public void makeCircular()
    {

    }

    // Print all the nodes in linkedlist, make sure it works on circular double linkedlist
    public void printAll()
    {

    }
    // Test the class
    public static void main(String[] args) {
        // Test all above methods

    }

}
```

Task02

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

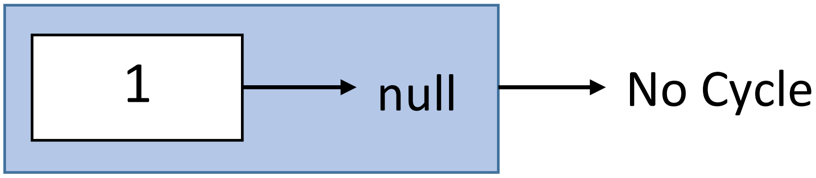
Now your task is to implement following methods (Single/Double LL) but this time you have to make another variable say **`tail`** for accessing last element directly.

- 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.


1
null
No Cycle