

# Data Structure and Algorithm Practicum

## Quiz 2



**Name**

Muhammad Baihaqi Aulia Asy'ari

**NIM**

2241720145

**Class**

1I

**Department**

Information Technology

**Study Program**

D4 Informatics Engineering

---

# 1 Code

## 1.1 Node.java

```
/*
 * To change this license header, choose License Headers in Project
 *    ↪ Properties.
 * To change this template file, choose Tools / Templates
 * and open the template in the editor.
 */
package quiz2_baihaqi;

public class Node {
    int data;
    Node n;
    Node p;

    Node(){
        data=0;
        n = p = null;
    }

    Node(int data){
        this.data = data;
        n = p = null;
    }

    Node(Node prev, int data, Node next){
        this.data = data;
        this.n = next;
        this.p = prev;
    }
}
```

## 1.2 SingleLinkedList.java

```
package quiz2_baihaqi;

public class SingleLinkedList {
    Node head, tail;
    int size;
```

---

```
SingleLinkedList(){
    head = tail = null;
    size=0;
}
boolean isEmpty(){
    return size==0;
}
void addFirst(int data){
    Node nu = new Node(data);
    if(isEmpty()){
        head = tail = nu;
    }else{
        nu.n = head;
        head = nu;
    }
    size++;
}
void deleteFirst(){
    Node tmp = head.n;
    head = head.n;
    tmp = null;
    size--;
}
void print(){
    Node tmp = head;
    while(tmp!=null){
        System.out.print(""+tmp.data+"-");
        tmp = tmp.n;
    }
    System.out.println("");
}
//1.A. complete the missing code addLast
/**
 * this method will add new node at the last
 */
void addLast(int data){
    Node nu = new Node(data);
    if(isEmpty()){
        head = tail = nu;
    }else{
        //complete here
    }
}
```

---

```

        /* Add tmp node for head reference */
        Node tmp = head;
        /* loop through head to find tail */
        while (tmp.n != null) {
            tmp = tmp.n;
        }
        /* set the tmp.n and tail as nu */
        tmp.n = nu;
        tail = nu;
    }
    size++;
}
//1.B. complete the deleteLast
/**
 * this method will remove tail
 */
void deleteLast(){
    /* create tmp */
    Node tmp = head;
    /* check if empty */
    if (isEmpty()) {
        System.out.println("Linked list is empty. can't remove");
    } else if (head.n == null) {
        /* do as the delete first do */
        head = head.n;
        tmp = null;
        size--;
    } else {
        /* loop through tmp to find second to last node */
        while (tmp.n.n != null) {
            tmp = tmp.n;
        }
        /* use the second to last node to delete the next/last node
        ↪ */
        tmp.n = null;
        tail = tmp;
        size--;
    }
}
}
}

```

---

### 1.3 Main.java

```
/*
 * To change this license header, choose License Headers in Project
 * Properties.
 * To change this template file, choose Tools / Templates
 * and open the template in the editor.
 */
package quiz2_baihaqi;

public class Main {
    public static void main(String[] args){
        SingleLinkedList dll = new SingleLinkedList();
        dll.addFirst(45);
        dll.addFirst(10);
        dll.addFirst(10);
        dll.addFirst(15);
        dll.addFirst(150);
        dll.print();
        dll.deleteFirst();
        dll.print();
        //continue to call addLast, deleteLast,
        dll.addLast(69);
        dll.print();
        dll.deleteLast();
        dll.print();
        //merge, split,
        split(dll);
        merge(dll, dll);
    }
    //2.A.
    public static void merge(SingleLinkedList l1,
        SingleLinkedList l2){
        //complete this method
    }
    //2.B. this will split sll to be 2 sll
    public static void split(SingleLinkedList l){
        //ex: 2,3,4,34,2,3,45,4 (original list)
        //1-> 2,3,4,34
        //2-> 2,3,45,4

        /* Create list 1 and 2 */
        SingleLinkedList l1 = new SingleLinkedList();
```

```

SingleLinkedList l2 = new SingleLinkedList();
/* Create Temp to loop trough */
Node temp = l.head;
/* Loop trough half the list and add data using the add
→ function */
for (int i = 0; i < l.size/2; i++) {
    l1.addLast(temp.data);
    temp = temp.n;
}
/* Loop trough the rest of the list and add the the data
→ using the add function */
while (temp != null) {
    l2.addLast(temp.data);
    temp = temp.n;
}
/* Print the data */
l1.print();
l2.print();
}
}

```

## 2 Running the program

