

Name: Yash Daga

Registration Number: 20BCE7323

Program-1

Q1. Write a Java program to concatenate two singly linked list.

Code

```
import java.util.*;

public class Concatenate {

    class Node{
        int data;
        Node next;

        public Node(int data) {
            this.data = data;
            this.next = null;
        }
    }

    public Node addNode(int data, Node head) {
        Node newNode = new Node(data);
        Node temp = head;
        if(head == null) {
            head = newNode;
            return head;
        }
        while(temp.next != null) {
```

```
        temp = temp.next;
    }
    temp.next = newNode;
    return head;
}
```

```
public Node concatenateList(Node top1, Node top2) {
    if(top1 == null) {
        top1 = top2;
        return top1;
    }
    Node temp = top1;
    while(temp.next != null) {
        temp = temp.next;
    }
    temp.next = top2;
    return top1;
}
```

```
public void display(Node head) {
    Node current = head;
    if(head == null) {
        System.out.println("List is empty");
        return;
    }
    while(current != null) {
```

```
        System.out.print(current.data + " ");  
        current = current.next;  
    }  
    System.out.println();  
}
```

```
public static void main(String[] args) {
```

```
    Concatenate list = new Concatenate();
```

```
    Scanner sc = new Scanner(System.in);
```

```
        System.out.println("Enter the number of nodes in top1 list");
```

```
        int n= sc.nextInt();
```

```
        Node top1 = null;
```

```
        for(int i=1;i<=n;i++)
```

```
        {
```

```
            top1 = list.addNode(i, top1);
```

```
        }
```

```
        System.out.println("Top1 Nodes - ");
```

```
        list.display(top1);
```

```
        System.out.println("Enter the number of nodes in top2 list");
```

```
        n= sc.nextInt();
```

```
        Node top2 = null;
```

```
        for(int i=1;i<=n;i++)
```

```
        {
```

```
        top2 = list.addNode(i, top2);
    }
    System.out.println("Top2 Nodes - ");
    list.display(top2);

    Node mergedList = list.concatenateList(top1, top2);
    System.out.println("Concatenated List - ");
    list.display(mergedList);
}

}
```

```
Command Prompt
D:\20BCE7323>javac Concatenate.java
D:\20BCE7323>java Concatenate
Enter the number of nodes in top1 list
5
Top1 Nodes -
1 2 3 4 5
Enter the number of nodes in top2 list
10
Top2 Nodes -
1 2 3 4 5 6 7 8 9 10
Concatenated List -
1 2 3 4 5 1 2 3 4 5 6 7 8 9 10
D:\20BCE7323>
```

```
D:\20BCE7323\Concatenate.java - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Reverse.java concatenate1.java CountNodes.java Concatenate.java
40
41 public void display(Node head) {
42     Node current = head;
43     if(head == null) {
44         System.out.println("List is empty");
45         return;
46     }
47     while(current != null) {
48         System.out.print(current.data + " ");
49         current = current.next;
50     }
51     System.out.println();
52 }
53
54 public static void main(String[] args) {
55
56     Concatenate list = new Concatenate();
57     Scanner sc = new Scanner(System.in);
58
59     System.out.println("Enter the number of nodes in top1 list");
60     int n= sc.nextInt();
61     Node top1 = null;
62     for(int i=1;i<=n;i++)
63     {
64         top1 = list.addNode(i, top1);
65     }
66     System.out.println("Top1 Nodes - ");
67     list.display(top1);
68
69     System.out.println("Enter the number of nodes in top2 list");
70     n= sc.nextInt();
71     Node top2 = null;
72     for(int i=1;i<=n;i++)
73     {
74         top2 = list.addNode(i, top2);
75     }
76     System.out.println("Top2 Nodes - ");
77     list.display(top2);
78
79     Node mergedList = list.concatenateList(top1, top2);
80     System.out.println("Concatenated List - ");
81     list.display(mergedList);
82 }
83
Java length: 1,942 lines: 84 Ln: 76 Col: 33 Pos: 1,764 Windows (CR LF) UTF-8 INS
```

Q2. Write a Java program to insert a node at given specified location in a doubly linked list.

Code

```
import java.util.*;

public class Nodeinsert {

    static class Node {

        public int data;

        public Node nextNode;

        public Node(int data)
        {
            this.data = data;
        }
    }

    static Node GetNode(int data)
    {
        return new Node(data);
    }

    static Node add_DLL(Node headNode, int position, int data)
    {
        Node head = headNode;
        if (position < 1)
            System.out.print("Invalid position");
        if (position == 1)
        {
            Node newNode = new Node(data);
            newNode.nextNode = headNode;
        }
    }
}
```

```
        head = newNode;
    }
    else
    {
        while (position-- != 0) {
            if (position == 1) {

                Node newNode = GetNode(data);

                newNode.nextNode = headNode.nextNode;
                headNode.nextNode = newNode;
                break;
            }
            headNode = headNode.nextNode;
        }
        if (position != 1)
            System.out.print("Position out of range");
    }
    return head;
}

static void PrintList(Node node)
{
    while (node != null)
    {
        System.out.print(node.data);
```

```
        node = node.nextNode;
        if (node != null)
            System.out.print(",");
    }
    System.out.println();
}
```

```
public static void main(String[] args)
{
    Scanner obj=new Scanner(System.in);
    Node head = GetNode(8);
    head.nextNode = GetNode(6);
    head.nextNode.nextNode = GetNode(2);
    head.nextNode.nextNode.nextNode = GetNode(13);

    System.out.print("Linked list before insertion: ");
    PrintList(head);

    int data, pos;

    System.out.print("Enter data : ");
    data=obj.nextInt();

    System.out.print("Enter Position : ");
    pos=obj.nextInt();
```



```
        head = add_DLL(head, pos, data);
        System.out.print("linked list after " +data+ " insertion at "+pos+" is :
");
        PrintList(head);
    }
}
```

```
Command Prompt
D:\208CE7323>javac Nodeinsert.java
D:\208CE7323>java Nodeinsert
Linked list before insertion: 8,6,2,13
Enter data : 1
Enter Position : 3
linked list after 1 insertion at 3 is : 8,6,1,2,13
D:\208CE7323>
```

```
Nodeinsert.java
41      System.out.print("Position out of range");
42  }
43  }
44  }
45  }
46  static void PrintList(Node node)
47  {
48      while (node != null)
49      {
50          System.out.print(node.data);
51          node = node.nextNode;
52          if (node != null)
53              System.out.print(",");
54      }
55      System.out.println();
56  }
57  }
58  public static void main(String[] args)
59  {
60      Scanner obj=new Scanner(System.in);
61      Node head = GetNode(8);
62      head.nextNode = GetNode(6);
63      head.nextNode.nextNode = GetNode(2);
64      head.nextNode.nextNode.nextNode = GetNode(13);
65
66      System.out.print("Linked list before insertion: ");
67      PrintList(head);
68
69      int data, pos;
70
71      System.out.print("Enter data : ");
72      data=obj.nextInt();
73
74      System.out.print("Enter Position : ");
75      pos=obj.nextInt();
76
77      head = add_DLL(head, pos, data);
78      System.out.print("linked list after " +data+ " insertion at "+pos+" is : ");
79      PrintList(head);
80  }
81  }
82  }
```