NAME –KASYAP VARANASI

REGISTRATION NUMBER –20BCE7315

Q1. Suppose a multi-digit number n is represented by a set of nodes in SLL(i.e., each node contains a digit). For example, SLL with three nodes (1->2->3) represents numerical value 123. Now, write a program to create two linked lists which represents two numbers n1 and n2. also write a methods static int sum () to return sum of n1 and n2.

Example: list-1" 1->2->7

List-2: 2->3->4

sum () returns 361 (i.e. 127+234)

CODE-

```
public class AddTwoNumberLinkedList{

    private static Node head;

    private static class Node {
        private int value;
        private Node next;

        Node(int value) {
```

```
            this.value = value;

        }
    }

    public void addToTheLast(Node node) {

        if (head == null) {
            head = node;
        } else {
            Node temp = head;
            while (temp.next != null)
                temp = temp.next;

            temp.next = node;
        }
    }

    public void printList(Node printNode) {
        Node temp = printNode;
        while (temp != null) {
            System.out.format("%d ", temp.value);
            temp = temp.next;
```

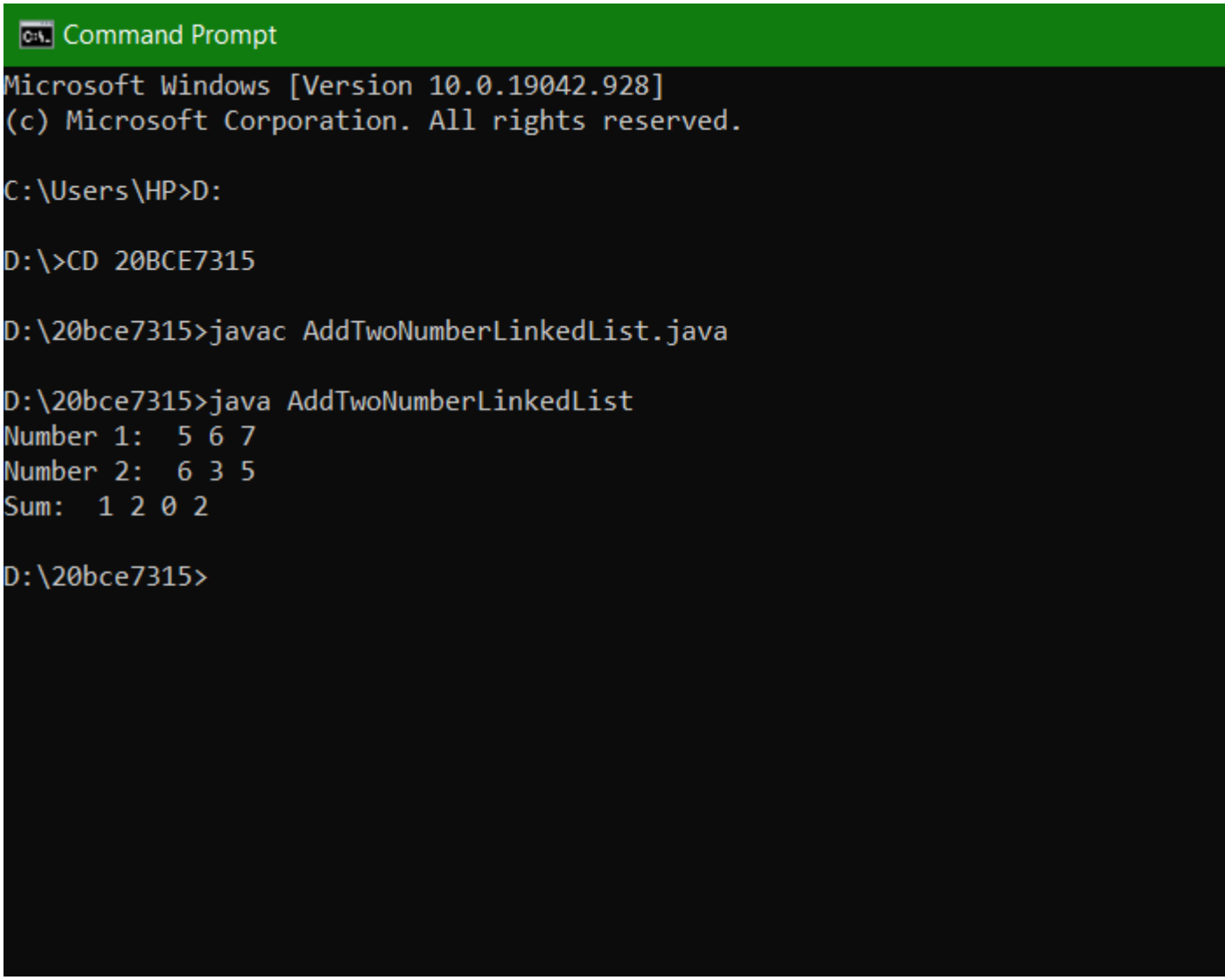
```
        }
        System.out.println();
    }

    public static Node reverseLinkedList(Node node) {
        if (node == null || node.next == null) {
            return node;
        }

        Node remaining = reverseLinkedList(node.next);
        node.next.next = node;
        node.next = null;
        return remaining;
    }


    public Node findSumOfNumbers(Node l1, Node l2) {
        int carry =0;

        Node newHead = null;
        Node tempNodeForIteration=null;
        int sum=0;
```

```
        int firstIter=0;
        while(l1!=null || l2!=null)
        {
            firstIter++;
            sum=carry;
            if(l1!=null)
            {
                sum=sum+l1.value;
                l1=l1.next;
            }

            if(l2!=null)
            {
                sum=sum+l2.value;
                l2=l2.next;
            }

            carry=sum/10;
            sum=sum%10;

            if(firstIter==1)
            {
                tempNodeForIteration = new Node(sum);
```

```
                newHead=tempNodeForIteration;
            }
            else
            {
                Node tempSumNode=new Node(sum);
                tempNodeForIteration.next=tempSumNode;

tempNodeForIteration=tempNodeForIteration.next;
            }


        }
        if(carry!=0)
        {
            Node tempNode=new Node(carry);
            tempNodeForIteration.next=tempNode;
        }
        return newHead;
    }

    public static void main(String[] args) {
        AddTwoNumberLinkedList list = new
AddTwoNumberLinkedList();
```

```
        Node head1=new Node(5);
        list.addToTheLast(head1);
        list.addToTheLast(new Node(6));
        list.addToTheLast(new Node(7));
        System.out.print("Number 1:  ");
        list.printList(head1);
        head=null;
        Node head2=new Node(6);
        list.addToTheLast(head2);
        list.addToTheLast(new Node(3));
        list.addToTheLast(new Node(5));
        System.out.print("Number 2:  ");
        list.printList(head2);
        head1=reverseLinkedList(head1);
        head2=reverseLinkedList(head2);
        Node result=
list.findSumOfNumbers(head1,head2);
        result=reverseLinkedList(result);
        System.out.print("Sum:  ");
        list.printList(result);
    }
}
```

OUTPUT-


Command Prompt
Microsoft Windows [Version 10.0.19042.928]
(c) Microsoft Corporation. All rights reserved.
C:\Users\HP>D:
D:\>CD 20BCE7315
D:\20bce7315>javac AddTwoNumberLinkedList.java
D:\20bce7315>java AddTwoNumberLinkedList
Number 1: 5 6 7
Number 2: 6 3 5
Sum: 1 2 0 2
D:\20bce7315>