Name – Yash Daga Roll Number – 20BCE7323

1.)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 method static int sum() to return sum of n1 and n2. Example: list-1" 1->2->7 List-2: 2->3->4 sum() return 361 (i.e 127+234)

Ans

Code :

public class AddtwoNumber

{

static Node head;

static class Node{

int value;

Node next;

Node(int value)

{

this.value=value;

}

}

public void addlast(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 reversell(Node node)

{

if(node ==null || node.next==null)

{

return node;

}

Node remain=reversell(node.next);

node.next.next=node;

node.next=null;

return remain;

}

public Node sumoftwo(Node l1, Node l2)

{

int carry=0;

Node newHead= null;

Node tempNode=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)

{

tempNode=new Node(sum);

newHead=tempNode;

}

else

{

Node tempsum=new Node(sum);

tempNode.next=tempsum;

tempNode=tempNode.next;

}

}

if(carry!=0)

{

Node tempN= new Node(carry);

tempNode.next=tempN;

}

return newHead;

}

public static void main (String[] args)

{

AddtwoNumber list = new AddtwoNumber();

Node head1= new Node(5);

list.addlast(head1);

list.addlast(new Node(6));

list.addlast(new Node(5));

System.out.print("List 1 :");

list.printList(head1);

head=null;

Node head2= new Node(5);

list.addlast(head2);

list.addlast(new Node(3));

list.addlast(new Node(5));

System.out.print("List 2 :");

list.printList(head2);

head1= reversell(head1);

head2= reversell(head2);

Node result = list.sumoftwo(head1,head2);

result=reversell(result);

System.out.print("Sum:");

list.printList(result);

}

}



