1)Class Main {

public static void main(String[] args) {

int arr[]={-1,2,3,5};

int sum=5;

int left=0;

int right=arr.length-1;

int[] res=new int[2];

while(left<right)

{

if(arr[left]+arr[right]==sum)

{

res[0]=left;

res[1]=right;

break;

}

else if(arr[left]+arr[right]<sum)

{

left++;

}

else

{

right--;

}

}

System.out.println(res[0]);

System.out.println(res[1]);

}

}

Output:-

1

2

class LinkedList{

Node head;

class Node

{

int data;

Node next;

Node(int d)

{

data=d;

next=null;

}

}

public void addf(int data)

{

Node first=new Node(data);

first.next=head;

head=first;

}

public void printList()

{

Node current=head;

while(current!=null)

{

System.out.println(current.data);

current=current.next;

}

}

}

2) import java.util.HashMap;

class Main {

public static int subarray(int[] arr,int k)

{

int sum=0;

int res=0;

HashMap<Integer,Integer> map=new HashMap<>();

for(int i=0;i<arr.length;i++)

{

sum=sum+arr[i];

if(sum==k)

{

res++;

}

if(map.containsKey(sum-k))

{

res=res+map.get(sum-k);

}

map.put(arr[i],map.getOrDefault(sum,0)+1);

}

return res;

}

public static void main(String[] args) {

int arr[]={10,2,-2,-20,10};

int k=-10;

int res=subarray(arr,k);

System.out.println(res);

}

}

Output:2

3)

class Main {

public static reverselist(Node head)

{

Node current=head;

Node prev=null;

Node temp;

while(current!=null)

{

temp=current.next;

current.next=prev;

prev=curr;

curr=temp;

}

}

public static addlinkedlist(Node l1,Node l2)

{

l1=reverselist(l1);

l2=reverselist(l2);

Node current=head;

int sum=0;

Node prev=null;

while(ll!=null||l2!=null)

{

if(l1!=null)

{

sum=sum+l1.data;

l1=l1.next;

}

if(l2!=null)

{

sum=sum+l2.data;

l2=l2.next;

}

Node res=new Node(sum%10);

int carry=sum/10;

current.next=res;

res=res.next;

}

}

public static void main(String[] args) {

LinkedList l1=new LinkedList();

l1.addf(5);

l1.addf(6);

l1.addf(5);

//System.out.println("created linked list is +",l1.printList());

LinkedList l2=new LinkedList();

l2.addf(1);

l2.addf(1);

l2.addf(1);

}

}