3 .Write a program to print all the LEADERS in the array. An element is a leader if it is greater than all the elements to its right side. And the rightmost element is always a leader.

```
public static void leaders(int [] input){
int  n=input.length;
for(int i=0;i<n;i++){
Int j;
for( j=i+1;j<n;j++){
if(input[i]<input[j]){
break;}
}
if(j==n){
System.out.print(input[i]+" ");
}
}
}
```

2 .Find the majority element in the array. A ***majority element*** in an array A[] of size n is an element that appears more than n/2 times (and hence there is at most one such element)

```
public  static int majorityElement(int [] nums){
int a=0,res=0;
HashMap<Integer,Integer> map=new HashMap<>();
for(int x:nums){
map.put(x,map.getOrDedault(x,0)+1);}
for(Map.Entry<Integer,Integer> e:map.entrySet()){
if(res<e.getValue()){
res=e.getValue();
a=e.getKey();}
return a;
}
```

4 .Given an array **arr[ ]** of size **N**, the task is to rotate the array by **d** position to the left.

```
public  static void shiftDpositions(int [] input,int d){
int  n=input.length;
int p=1;
while(p<=d){
```

```
int  last=input[0];
for(int i=0;i<n-1;i++){
input[i]=input[i+1];}
input[n-1]=last;
p++;
}
for(int i=0;i<n;i++){
System.out.print(input[i]+" ");}
}
```

1. Write a program that, given an array A[] of n numbers and another number x, determines whether or not there exist two elements in A[] whose sum is exactly x.

```
public static  boolean IsSum(int [] input,int x){
HashMap<Integer,Integer> map=new HashMap<>();
for(int i=0;i<input.length;i++){
if(map.containsKey(input[i])){
return true;}
else{
map.put(x-input[i],map.getOrDefaullt(nums[i],0)+1);
}
return false;}
```

5. Given an array arr[] of n integers, construct a Product Array prod[] (of same size) such that prod[i] is equal to the product of all the elements of arr[] except arr[i]. Solve it without division operator in O(n) time

```
public static  int[] productExceptItself(int [] arr){
int n=arr.length,prod=1;
int [] result=new int[n];
result[n-1]=arr[n-1];
for(int i=n-2;i>=0;i–){
result[i]=arr[i]*result[i+1];
```

```
}
for(int i=0;i<n-1;i++){
result[i]=result[i+1]*prod;
prod*=arr[i];}
result[n-1]=prod;
return result;
}
```