### Operations on PriorityQueue

Given an array **A** of **N** integers, your task is to **add**these elements to the **PriorityQueue**. Also, given an array **B**of **M** integers, the task is to check**if the given element is present in the PriorityQueue** or not.  
If the element is present, **then 1 is printed by the driver code, after that the** **max element**of priority queue is printed. Then the driver code deletes the max element.  
**Note:** Here the driver code has implemented the PriorityQueue as a max-heap.

**Example:**

**Input:**

N = 8

A = {1, 2, 3, 4, 5, 2, 3, 1}

M = 5

B = {1, 3, 2, 9, 10}

**Output:**

1

5

1

4

1

3

-1

-1

**Explanation:**

After inserting elements present in A, when we find B[0]=1,

which is present, so 1 gets printed, and then the

top element of the PriorityQueue which is 5 gets

printed, and then it gets deleted. Similarly, when

element is not present, **just -1 is printed.**

**Example 2:**

**Input:**

N = 4

A = {1, 2, 3, 4}

M = 2

B = {1, 10}

**Output:**

1

4

-1

**Explanation:**

After inserting elements present in A, when we find B[0]=1,

which is present, so 1 gets printed, and then the

top element of the PriorityQueue which is 4 gets

printed, and then it gets deleted. Similarly, when

element is not present, **just -1 is printed.**

//{ Driver Code Starts

//Initial Template for Java

import java.io.\*;

import java.util.\*;

class helper{

// Function to insert element into the queue

static void insert(PriorityQueue<Integer> q, int k){

q.add(k);

}

// Function to find an element k

static boolean find(PriorityQueue<Integer> q, int k){

for(int x:q){

if(x==k)

return true;

}

return false;

}

// Function to delete the max element from queue

static int delete(PriorityQueue<Integer> q){

//Queue<Integer> qu=new LinkedList<>();

if(q.isEmpty())

return -1;

return q.poll();

}

}

//{ Driver Code Starts.

// Driver class with driver code

class CodingMaxima {

// Driver code

public static void main (String[] args) {

// Taking input using scanner class

Scanner sc = new Scanner(System.in);

int testcase = sc.nextInt();

while(testcase-- > 0){

// Priority Queue with comparator

PriorityQueue<Integer> p\_queue = new PriorityQueue<Integer>(new Comparator<Integer>() {

public int compare(Integer w1, Integer w2) {

return w2.compareTo(w1);

}

});

int n = sc.nextInt();

// Invoking object of Geeks class

helper obj = new helper ();

for(int i = 0;i<n;i++){

int k = sc.nextInt();

obj.insert(p\_queue, k);

}

//Taking total number queries

int x = sc.nextInt();

for(int i = 0;i<x;i++){

int k = sc.nextInt();

boolean f = obj.find(p\_queue, k);

if(f != false){

System.out.println("1");

System.out.println(obj.delete(p\_queue));

}

else{

System.out.println("-1");

}

}

}

}

}

// } Driver Code Ends