Set - 1

Programming Problems for InfyTQ Round 2

Q1. Given an array arr of integers of size N and an integer K, the task is to find the K larger values of the array (repetition allowed)

Input:
1. The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.
2. The first line of each test case contains two space-separated integers N and K
3. The second line contains N space-separated positive integers represents array arr.

Output: For each test case, print K space-separated values in non-increasing order

Constraints:
1. 1 <= T <= 10
2. 1 <= K <= N <= 100000
3. 1 <= arr[i] <= 10^9

Example:
Input:
```
2
3 2
4 1 3
4 3
4 8 1 8
```

Output:
```
4 3
8 8 4
```

Q2. Given an array arr of integers of size N and an integer K, the task is to check if there exit K consecutive odd numbers or not if the elements are arranged in non-decreasing order.

Input:
1. The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.
2. The first line of each test case contains two space-separated integers N and K
3. The second line contains N space-separated positive integers represents array arr.

Output: For each test case, If exits print print "yes". Otherwise "no" (without quotes)

Constraints:
1. 1 <= T <= 10
2. 1 <= K <= N <= 100000
3. 1 <= arr[i] <= 10^9

Example:
Input:

```
3
4 2
4 1 3
7 3
2 5 1 5 3 7 8
5 2
5 4 3 2 1
```

Output:

```
yes
yes
no
```

Explanation:
Test case 1: Elements in non-decreasing order will be {1, 3, 4}. And there exit 2 consecutive odd numbers.
Test case 2: Elements in non-decreasing order will be {1, 2, 3, 5, 5, 7, 8}. And there exit 3 consecutive odd numbers.
Test case 3: Elements in non-decreasing order will be {1, 2, 3, 4, 5}. And there does not exit 2 consecutive odd numbers.

Q3. Given a number N, find least prime factors for all numbers from 1 to N. The least prime factor of an integer N is the smallest prime number that divides it.
Note : The least prime factor of all even numbers is 2. A prime number is its own least prime factor (as well as its own greatest prime factor).1 needs to be printed for 1.â€‹

Example 1:

Input: N = 6

Output: [1, 2, 3, 2, 5, 2]

Explanation: least prime factor of 1 = 1,

least prime factor of 2 = 2,

least prime factor of 3 = 3,

least prime factor of 4 = 2,

least prime factor of 5 = 5,

least prime factor of 6 = 2.

So answer is[1, 2, 3, 2, 5, 2].

Example 2:

Input: N = 4

Output: [1, 2, 3, 2]

Explanation: least prime factor of 1 = 1,

least prime factor of 2 = 2,

least prime factor of 3 = 3,

least prime factor of 4 = 2.

So answer is[1, 2, 3, 2].

Your Task:
You dont need to read input or print anything. Complete the function leastPrimeFactor() which takes N as input parameter and returns a list of integers containing all the least prime factor of each numbers from 1 to N.
Expected Time Complexity: O(NlogN)
Expected Auxiliary Space: O(N)
Constraints:
2<= n <=10^3

Q4. Write a function which takes a list sorted in non-decreasing order and deletes any duplicate nodes from the list. The list should only be traversed once.
For example if the linked list is 11->11->11->21->43->43->60 then removeDuplicates() should convert the list to 11->21->43->60.