Problem Statement

Suggest Edit

You are given an array “A” of N integers. Your task is to find the maximum element in all K sized contiguous subarrays from left to right.

For Example:

```
If A = [3, 2, 3], and K = 2.
Then max of [3, 2] = 3 and max of [2, 3] = 3
So, the answer will be [3, 3]

If A = [3, 2, 3, 5, 1, 7] and K = 3.
Then max of [3, 2, 3] = 3
Then max of [2, 3, 5] = 5
Then max of [3, 5, 1] = 5
Then max of [5, 1, 7] = 7
So  the answer will be [3, 5, 5, 7]
```

Follow Up :

```
Can you solve the problem in O(N) time complexity and O(K) space complexity?
```

Input Format :

```
The first line of input contains a single integer T, representing the number of
test cases or queries to be run.
Then the T test cases follow.

The first line of each test contains two space-separated integers N and K.

The second line of each test contains N space-separated integers, denoting the
elements of array A.
```

Output Format :

```
For each test case, print a single line containing N -  K + 1 space-separated
integers denoting values of the maximum element in K size subarrays.
```

Note:

```
You do not need to print anything, it has already been taken care of. Just
implement the given function.
```

Constraints :

```
1 <= T <= 10
1 <= N <= 10^5
1 <= K <= N
1 <= A[i] <= 10^9

Time Limit: 1sec
```

Sample Input 1 :

```
2
3 1
2 1 1
3 2
1 1 3
```

Sample Output 1 :

```
2 1 1
1 3
```

Explanation For Sample Input 1:

```
For the first test case,the given A = [2, 1, 1]  and K = 1
Therefore, max([2]) = 2 , max([1]) = 1, max([1]) = 1
Hence our answer is [2, 1, 1]

For the second test case, the given A = [1, 1, 3] and K = 2
Therefore, max([1, 1]) = 1, max([1, 3]) = 3
Hence our answer is [1, 3].
```

Sample Input 2 :

```
2
3 2
1 3 1
5 3
1 2 3 4 5
```

Sample Output 2 :

```
3 3
3 4 5
```