Problem G. Seven Nevers

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

The *longest increasing subsequence* problem is to find a subsequence of a given sequence in which the subsequence's elements are in sorted order, lowest to highest, and in which the subsequence is as long as possible. This subsequence is not necessarily contiguous.

You are given a permutation of the first n positive integers $a_1, a_2, \ldots, a_n$ and an integer k. Your task is to find, for each i from 1 to $n - k + 1$, the length of the longest increasing subsequence of the sequence $a_1, a_2, \ldots, a_{i-1}, a_{i+k}, a_{i+k+1}, \ldots, a_n$ (in other words, the sequence obtained by erasing $a_i, a_{i+1}, \ldots, a_{i+k-1}$ from a).

Input

The first line of the input contains two integers n and k ($1 \le k < n \le 3 \cdot 10^5$), denoting the length of the given permutation and the number of consecutive elements to be removed.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$; $a_i \ne a_j$ for $i \ne j$), denoting the elements of the permutation.

Output

Display $n-k+1$ integers, one per line, where the i-th integer denotes the length of the longest increasing subsequence of the sequence $a_1, a_2, \ldots, a_{i-1}, a_{i+k}, a_{i+k+1}, \ldots, a_n$ for $i = 1, 2, \ldots, n - k + 1$.

Example

standard input	standard output
8 3 6 5 3 1 8 2 4 7	4 3 3 3 2 2