Partsort

SEASON 10 – SECOND ROUND

Given is a numeric sequence, which is a permutation of the numbers from 1 to $\boldsymbol{N}$. We say that the sequence is sorted if for each $i = 1, 2, \ldots, N - 1$ the inequality $A_i < A_{i+1}$ is satisfied, where A_i denotes the i-th number in the sequence. In the not-so-distant past the computers had way less operational memory and namely because of this to sort a sequence was not a trivial task.

We know that we can load in the memory at most $\boldsymbol{K}$ of the elements of the sequence at a time. That is why, we first sort the numbers with indices between 1 and K, then those between 2 and $K + 1$ and so on until we sort the elements from index $N - K + 1$ to index N. Unfortunately, this is not always sufficient to sort the sequence. For instance, if $N = 5$, $K = 3$ and $A = \{4, 5, 3, 1, 2\}$, we ger the following changes: $\{\mathbf{3}, \mathbf{4}, \mathbf{5}, 1, 2\} \rightarrow \{3, \mathbf{1}, \mathbf{4}, \mathbf{5}, 2\} \rightarrow \{3, 1, \mathbf{2}, \mathbf{4}, \mathbf{5}\}$. So, after one such step, we will change the original sequence to $\{3, 1, 2, 4, 5\}$ and it will take us one more step to sort the sequence completely.

Write a program, which finds the number of steps that the algorithm will do, in order to sort the given sequence. The program has to process $\boldsymbol{T}$ test cases during a single execution.

Input

The first line of the input file `partsort.in` contains a single number T. Each of the following T lines describes one test case in the format – $N, K, A_1, A_2, \ldots, A_N$.

Output

On N lines of the output file `partsort.out` print one number equal to the required number of steps needed to sort the sequence from the corresponding test case.

Constraints

$2 \leq K \leq N \leq 10\,000$

The sum of $N \div K$ over all test cases will not exceed 100.

Example

Input	Output
3 5 3 4 5 3 1 2 3 2 1 2 3 7 7 7 6 5 4 3 2 1	2 0 1