Swirly Sort

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You're given an array A containing n integers, and an integer k ($1 \leq k \leq n$). You would like to transform it into a sorted array.

You can perform the following operations on A any number of times:

- Choose k indices $1 \leq i_1 < i_2 < i_3 < \ldots < i_k \leq n$ and cyclically shift the values at these indices.
  - That is, $A[i_1]$ moves to index i_2, $A[i_2]$ moves to index i_3, $\ldots$, $A[i_k]$ moves to index i_1.
  - Note that the indices you choose for the cyclic shift must always be increasing.
  - This operation has cost 0.
- Choose an index i, and either increment or decrement $A[i]$ by 1.
  - This operation has cost 1.

Find the minimum total cost of operations to transform the array into a sorted array.

Notes:

- The array A is one-indexed.
- An array X is sorted if $X_1 \leq X_2 \leq X_3 \leq \ldots$.

Input

The first line of input contains a single integer t, denoting the number of test cases.

Each test case consists of two lines of input.

- The first line contains two space-separated integers n and k.
- The second line contains n space-separated integers $A_1, A_2, \ldots, A_n$, the initial values of array A.

- $1 \leq t \leq 10^5$
- $1 \leq n \leq 3 \cdot 10^5$
- $1 \leq k \leq n$
- $1 \leq A[i] \leq 10^9$
- The sum of $n \cdot k$ across all test cases is $\leq 3 \cdot 10^5$.

Output

For each test case, print a line containing an integer: the minimum cost to transform A into a sorted array.

Example

standard input	standard output
4 4 1 6 4 3 7 4 2 6 4 3 7 4 3 6 4 3 7 4 4 6 4 3 7	3 0 1 2

Note

In all samples, the initial array is $A = [6, 4, 3, 7]$.

- For $k = 1$, we subtract 2 from the first element and add 1 to the third element, turning the array into $A = [4, 4, 4, 7]$.
- For $k = 2$, we can choose $i_1 = 1$ and $i_2 = 3$, transforming the array into $[3, 4, 6, 7]$ which is sorted. This has a cost of 0.
- For $k = 3$, the following process is optimal:
- Choose $i_1 = 1$, $i_2 = 2$, $i_3 = 3$. The array becomes $[3, 6, 4, 7]$.
- Choose $i_1 = 1$, $i_2 = 2$, $i_3 = 3$ again. The array becomes $[4, 3, 6, 7]$.
- Subtract 1 from the first element to obtain $[3, 3, 6, 7]$.
- For $k = 4$, the following is optimal:
- Add 1 to the first element. The array becomes $[7, 4, 3, 7]$.
- Subtract 1 from the second element. The array becomes $[7, 3, 3, 7]$.
- Choose all four elements and cyclic shift to obtain $[7, 7, 3, 3]$.
- Cyclic shift again and obtain $[3, 7, 7, 3]$.
- Cyclic shift again and obtain $[3, 3, 7, 7]$.