

E. Little Elephant and Inversions

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

The Little Elephant has array a , consisting of n positive integers, indexed from 1 to n . Let's denote the number with index i as a_i .

The Little Elephant wants to count, how many pairs of integers l and r are there, such that $1 \leq l < r \leq n$ and sequence $b = a_1 a_2 \dots a_l a_r a_{r+1} \dots a_n$ has no more than k inversions.

An **inversion** in sequence b is a pair of elements of the sequence b , that change their relative order after a stable sorting of the sequence. In other words, an inversion is a pair of integers i and j , such that $1 \leq i < j \leq |b|$ and $b_i > b_j$, where $|b|$ is the length of sequence b , and b_j is its j -th element.

Help the Little Elephant and count the number of the described pairs.

Input

The first line contains two integers n and k ($2 \leq n \leq 10^5$, $0 \leq k \leq 10^{18}$) — the size of array a and the maximum allowed number of inversions respectively. The next line contains n positive integers, separated by single spaces, a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — elements of array a .

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use `cin`, `cout` streams or the `%I64d` specifier.

Output

In a single line print a single number — the answer to the problem.

Examples

input
3 1 1 3 2
output
3

input
5 2 1 3 2 1 7
output
6