Codeforces Round #466 (Div. 2)

A. Points on the line

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

We've got no test cases. A big olympiad is coming up. But the problemsetters' number one priority should be adding another problem to the round.

The **diameter** of a multiset of points on the line is the largest distance between two points from this set. For example, the diameter of the multiset $\{1, 3, 2, 1\}$ is 2.

Diameter of multiset consisting of one point is 0.

You are given n points on the line. What is the minimum number of points you have to remove, so that the diameter of the multiset of the remaining points will not exceed d?

Input

The first line contains two integers n and d ($1 \le n \le 100$, $0 \le d \le 100$) — the amount of points and the maximum allowed diameter respectively.

The second line contains n space separated integers ($1 \le x_i \le 100$) — the coordinates of the points.

Output

Output a single integer — the minimum number of points you have to remove.

Examples

input
3 1 2 1 4
output
1

input
3 0 7 7 7
output
0

input
6 3 1 3 4 6 9 10
output
3

Note

In the first test case the optimal strategy is to remove the point with coordinate 4. The remaining points will have coordinates 1 and 2, so the diameter will be equal to $2 - 1 = 1$.

In the second test case the diameter is equal to 0, so its is unnecessary to remove any points.

In the third test case the optimal strategy is to remove points with coordinates 1, 9 and 10. The remaining points will have coordinates 3, 4 and 6, so the diameter will be equal to $6 - 3 = 3$.

B. Our Tanya is Crying Out Loud

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Right now she actually isn't. But she will be, if you don't solve this problem.

You are given integers n, k, A and B. There is a number x, which is initially equal to n. You are allowed to perform two types of operations:

1. Subtract 1 from x. This operation costs you A coins.
2. Divide x by k. Can be performed only if x is divisible by k. This operation costs you B coins.

What is the minimum amount of coins you have to pay to make x equal to 1?

Input

The first line contains a single integer n ($1 \le n \le 2 \cdot 10^9$).

The second line contains a single integer k ($1 \le k \le 2 \cdot 10^9$).

The third line contains a single integer A ($1 \le A \le 2 \cdot 10^9$).

The fourth line contains a single integer B ($1 \le B \le 2 \cdot 10^9$).

Output

Output a single integer — the minimum amount of coins you have to pay to make x equal to 1.

Examples

input
9 2 3 1
output
6

input
5 5 2 20
output
8

input
19 3 4 2
output
12

Note

In the first testcase, the optimal strategy is as follows:

- Subtract 1 from x ($9 \rightarrow 8$) paying 3 coins.
- Divide x by 2 ($8 \rightarrow 4$) paying 1 coin.
- Divide x by 2 ($4 \rightarrow 2$) paying 1 coin.
- Divide x by 2 ($2 \rightarrow 1$) paying 1 coin.

The total cost is 6 coins.

In the second test case the optimal strategy is to subtract 1 from x 4 times paying 8 coins in total.

C. Phone Numbers

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

And where the are the phone numbers?

You are given a string s consisting of lowercase English letters and an integer k. Find the lexicographically smallest string t of length k, such that its set of letters is a subset of the set of letters of s and s is lexicographically smaller than t.

It's guaranteed that the answer exists.

Note that the set of letters is a set, not a multiset. For example, the set of letters of `abadaba` is $\{a, b, d\}$.

String p is lexicographically smaller than string q, if p is a prefix of q, is not equal to q or there exists i, such that $p_i < q_i$ and for all $j < i$ it is satisfied that $p_j = q_j$. For example, `abc` is lexicographically smaller than `abcd` , `abd` is lexicographically smaller than `abec`, `afa` **is not** lexicographically smaller than `ab` and `a` **is not** lexicographically smaller than `a`.

Input
The first line of input contains two space separated integers n and k ($1 \le n, k \le 100\,000$) — the length of s and the required length of t.

The second line of input contains the string s consisting of n lowercase English letters.

Output
Output the string t conforming to the requirements above.

It's guaranteed that the answer exists.

Examples

input
3 3 abc
output
aca

input
3 2 abc
output
ac

input
3 3 ayy
output
yaa

input
2 3 ba
output
baa

Note
In the first example the list of strings t of length 3, such that the set of letters of t is a subset of letters of s is as follows: `aaa`, `aab`, `aac`, `aba`, `abb`, `abc`, `aca`, `acb`, ... Among them, those are lexicographically greater than `abc`: `aca`, `acb`, ... Out of those the lexicographically smallest is `aca`.

D. Alena And The Heater

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

"We've tried solitary confinement, waterboarding and listening to Just In Beaver, to no avail. We need something extreme."

"Little Alena got an array as a birthday present…"

The array b of length n is obtained from the array a of length n and two integers l and r ($l \le r$) using the following procedure:

$b_1 = b_2 = b_3 = b_4 = 0$.

For all $5 \le i \le n$:

- $b_i = 0$ if $a_i, a_{i-1}, a_{i-2}, a_{i-3}, a_{i-4} > r$ and $b_{i-1} = b_{i-2} = b_{i-3} = b_{i-4} = 1$
- $b_i = 1$ if $a_i, a_{i-1}, a_{i-2}, a_{i-3}, a_{i-4} < l$ and $b_{i-1} = b_{i-2} = b_{i-3} = b_{i-4} = 0$
- $b_i = b_{i-1}$ otherwise

You are given arrays a and b' of the same length. Find two integers l and r ($l \le r$), such that applying the algorithm described above will yield an array b equal to b'.

It's guaranteed that the answer exists.

Input
The first line of input contains a single integer n ($5 \le n \le 10^5$) — the length of a and b'.

The second line of input contains n space separated integers $a_1, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$) — the elements of a.

The third line of input contains a string of n characters, consisting of `0` and `1` — the elements of b'. Note that they are not separated by spaces.

Output
Output two integers l and r (- $10^9 \le l \le r \le 10^9$), conforming to the requirements described above.

If there are multiple solutions, output any of them.

It's guaranteed that the answer exists.

Examples

input
5 1 2 3 4 5 00001
output
6 15

input
10 -10 -9 -8 -7 -6 6 7 8 9 10 0000111110
output
-5 5

Note
In the first test case any pair of l and r pair is valid, if $6 \le l \le r \le 10^9$, in that case $b_5 = 1$, because $a_1, ..., a_5 < l$.

E. Cashback

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Since you are the best Wraith King, Nizhniy Magazin «Mir» at the centre of Vinnytsia is offering you a discount.

You are given an array a of length n and an integer c.

The value of some array b of length k is the sum of its elements except for the $\lfloor \frac{k}{c} \rfloor$ smallest. For example, the value of the array $[3, 1, 6, 5, 2]$ with $c = 2$ is $3 + 6 + 5 = 14$.

Among all possible partitions of a into contiguous subarrays output the smallest possible sum of the values of these subarrays.

Input
The first line contains integers n and c ($1 \le n, c \le 100\,000$).

The second line contains n integers a_i ($1 \le a_i \le 10^9$) — elements of a.

Output
Output a single integer — the smallest possible sum of values of these subarrays of some partition of a.

Examples

input
3 5 1 2 3
output
6

input
12 10 1 1 10 10 10 10 10 10 9 10 10 10
output
92

input
7 2 2 3 6 4 5 7 1
output
17

input
8 4 1 3 4 5 5 3 4 1
output
23

Note

In the first example any partition yields 6 as the sum.

In the second example one of the optimal partitions is $[1, 1], [10, 10, 10, 10, 10, 10, 9, 10, 10, 10]$ with the values 2 and 90 respectively.

In the third example one of the optimal partitions is $[2, 3], [6, 4, 5, 7], [1]$ with the values 3, 13 and 1 respectively.

In the fourth example one of the optimal partitions is $[1], [3, 4, 5, 5, 3, 4], [1]$ with the values 1, 21 and 1 respectively.

F. Machine Learning

time limit per test: 4 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

You come home and fell some unpleasant smell. Where is it coming from?

You are given an array a. You have to answer the following queries:

1. You are given two integers l and r. Let c_i be the number of occurrences of i in $a_{l:\,r}$, where $a_{l:\,r}$ is the subarray of a from l-th element to r-th inclusive. Find the **Mex** of $\{c_0, c_1, \ldots, c_{10^9}\}$
2. You are given two integers p to x. Change a_p to x.

The **Mex** of a multiset of numbers is the smallest non-negative integer **not in** the set.

Note that in this problem all elements of a are positive, which means that $c_0 = 0$ and 0 is never the answer for the query of the second type.

Input

The first line of input contains two integers n and q ($1 \le n, q \le 100\,000$) — the length of the array and the number of queries respectively.

The second line of input contains n integers — $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$).

Each of the next q lines describes a single query.

The first type of query is described by three integers $t_i = 1, l_i, r_i$, where $1 \le l_i \le r_i \le n$ — the bounds of the subarray.

The second type of query is described by three integers $t_i = 2, p_i, x_i$, where $1 \le p_i \le n$ is the index of the element, which must be changed and $1 \le x_i \le 10^9$ is the new value.

Output

For each query of the first type output a single integer — the **Mex** of $\{c_0, c_1, \ldots, c_{10^9}\}$.

Example

input
10 4 1 2 3 1 1 2 2 2 9 9 1 1 1 1 2 8 2 7 1 1 2 8
output
2 3 2

Note

The subarray of the first query consists of the single element — 1.

The subarray of the second query consists of four 2s, one 3 and two 1s.

The subarray of the fourth query consists of three 1s, three 2s and one 3.