Codeforces Round #215 (Div. 2)

A. Sereja and Coat Rack

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

Sereja owns a restaurant for n people. The restaurant hall has a coat rack with n hooks. Each restaurant visitor can use a hook to hang his clothes on it. Using the i-th hook costs a_i rubles. Only one person can hang clothes on one hook.

Tonight Sereja expects m guests in the restaurant. Naturally, each guest wants to hang his clothes on an available hook with minimum price (if there are multiple such hooks, he chooses any of them). However if the moment a guest arrives the rack has no available hooks, Sereja must pay a d ruble fine to the guest.

Help Sereja find out the profit in rubles (possibly negative) that he will get tonight. You can assume that before the guests arrive, all hooks on the rack are available, all guests come at different time, nobody besides the m guests is visiting Sereja's restaurant tonight.

Input

The first line contains two integers n and d ($1 \le n, d \le 100$). The next line contains integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 100$). The third line contains integer m ($1 \le m \le 100$).

Output

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

Examples

input
2 1 2 1 2
output
3

input
2 1 2 1 10
output
-5

Note

In the first test both hooks will be used, so Sereja gets $1 + 2 = 3$ rubles.

In the second test both hooks will be used but Sereja pays a fine 8 times, so the answer is $3 - 8 = -5$.

B. Sereja and Suffixes

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

Sereja has an array a, consisting of n integers $a_1, a_2, \ldots, a_n$. The boy cannot sit and do nothing, he decided to study an array. Sereja took a piece of paper and wrote out m integers $l_1, l_2, \ldots, l_m$ $(1 \le l_i \le n)$. For each number l_i he wants to know how many distinct numbers are staying on the positions $l_i, l_i + 1, \ldots, n$. Formally, he want to find the number of distinct numbers among $a_{l_i}, a_{l_i+1}, \ldots, a_n$.?

Sereja wrote out the necessary array elements but the array was so large and the boy was so pressed for time. Help him, find the answer for the described question for each l_i.

Input

The first line contains two integers n and m $(1 \le n, m \le 10^5)$. The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^5)$ — the array elements.

Next m lines contain integers $l_1, l_2, \ldots, l_m$. The i-th line contains integer l_i $(1 \le l_i \le n)$.

Output

Print m lines — on the i-th line print the answer to the number l_i.

Examples

input
10 10 1 2 3 4 1 2 3 4 100000 99999 1 2 3 4 5 6 7 8 9 10

output
6 6 6 6 6 5 4 3 2 1

C. Sereja and Algorithm

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

Sereja loves all sorts of algorithms. He has recently come up with a new algorithm, which receives a string as an input. Let's represent the input string of the algorithm as $q = q_1 q_2 \ldots q_k$. The algorithm consists of two steps:

1. Find any continuous subsequence (substring) of three characters of string q, which doesn't equal to either string "zyx", "xzy", "yxz". If q doesn't contain any such subsequence, terminate the algorithm, otherwise go to step 2.
2. Rearrange the letters of the found subsequence randomly and go to step 1.

Sereja thinks that the algorithm works correctly on string q if there is a non-zero probability that the algorithm will be terminated. But if the algorithm anyway will work for infinitely long on a string, then we consider the algorithm to work incorrectly on this string.

Sereja wants to test his algorithm. For that, he has string $s = s_1 s_2 \ldots s_n$, consisting of n characters. The boy conducts a series of m tests. As the i-th test, he sends substring $s_{l_i} s_{l_i+1} \ldots s_{r_i}$ $(1 \le l_i \le r_i \le n)$ to the algorithm input. Unfortunately, the implementation of his algorithm works too long, so Sereja asked you to help. For each test (l_i, r_i) determine if the algorithm works correctly on this test or not.

Input

The first line contains non-empty string s, its length (n) doesn't exceed 10^5. It is guaranteed that string s only contains characters: 'x', 'y', 'z'.

The second line contains integer m $(1 \le m \le 10^5)$ — the number of tests. Next m lines contain the tests. The i-th line contains a pair of integers l_i, r_i $(1 \le l_i \le r_i \le n)$.

Output

For each test, print "YES" (without the quotes) if the algorithm works correctly on the corresponding test and "NO" (without the quotes) otherwise.

Examples

input
zyxxxxxxyyz 5 5 5 1 3 1 11 1 4 3 6
output
YES YES NO YES NO

Note

In the first example, in test one and two the algorithm will always be terminated in one step. In the fourth test you can get string "xzyx" on which the algorithm will terminate. In all other tests the algorithm doesn't work correctly.

D. Sereja ans Anagrams

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

Sereja has two sequences a and b and number p. Sequence a consists of n integers $a_1, a_2, \ldots, a_n$. Similarly, sequence b consists of m integers $b_1, b_2, \ldots, b_m$. As usual, Sereja studies the sequences he has. Today he wants to find the number of positions q $(q + (m - 1) \cdot p \le n;\ q \ge 1)$, such that sequence b can be obtained from sequence $a_q, a_{q+p}, a_{q+2p}, \ldots, a_{q+(m-1)p}$ by rearranging elements.

Sereja needs to rush to the gym, so he asked to find all the described positions of q.

Input

The first line contains three integers n, m and p $(1 \le n, m \le 2 \cdot 10^5, 1 \le p \le 2 \cdot 10^5)$. The next line contains n integers a_1, a_2, $\ldots$, a_n $(1 \le a_i \le 10^9)$. The next line contains m integers b_1, b_2, $\ldots$, b_m $(1 \le b_i \le 10^9)$.

Output

In the first line print the number of valid qs. In the second line, print the valid values in the increasing order.

Examples

input
5 3 1 1 2 3 2 1 1 2 3
output
2 1 3

input
6 3 2 1 3 2 2 3 1 1 2 3
output
2 1 2

E. Sereja and the Arrangement of Numbers

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

Let's call an array consisting of n integer numbers $a_1, a_2, \ldots, a_n$, *beautiful* if it has the following property:

- consider all pairs of numbers x, y $(x \neq y)$, such that number x occurs in the array a and number y occurs in the array a;
- for each pair x, y must exist some position j $(1 \leq j < n)$, such that at least one of the two conditions are met, either $a_j = x, a_{j+1} = y$, or $a_j = y, a_{j+1} = x$.

Sereja wants to build a beautiful array a, consisting of n integers. But not everything is so easy, Sereja's friend Dima has m coupons, each contains two integers q_i, w_i. Coupon i costs w_i and allows you to use as many numbers q_i as you want when constructing the array a. Values q_i are distinct. Sereja has no coupons, so Dima and Sereja have made the following deal. Dima builds some beautiful array a of n elements. After that he takes w_i rubles from Sereja for each q_i, which occurs in the array a. Sereja believed his friend and agreed to the contract, and now he is wondering, what is the maximum amount of money he can pay.

Help Sereja, find the maximum amount of money he can pay to Dima.

Input

The first line contains two integers n and m $(1 \leq n \leq 2 \cdot 10^6, 1 \leq m \leq 10^5)$. Next m lines contain pairs of integers. The i-th line contains numbers q_i, w_i $(1 \leq q_i, w_i \leq 10^5)$.

It is guaranteed that all q_i are distinct.

Output

In a single line print maximum amount of money (in rubles) Sereja can pay.

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

Examples

input
5 2 1 2 2 3
output
5

input
100 3 1 2 2 1 3 1
output
4

input
1 2 1 1 2 100
output
100

Note

In the first sample Sereja can pay 5 rubles, for example, if Dima constructs the following array: $[1, 2, 1, 2, 2]$. There are another optimal arrays for this test.

In the third sample Sereja can pay 100 rubles, if Dima constructs the following array: $[2]$.