A. Simulating Move to Front

time limit per test: 3 seconds
memory limit per test: 256 megabytes

Start with a list $[0, 1, 2, \ldots, n-1]$. Now you get a sequence of m **move to front** requests, where $\texttt{mtf}(i)$ moves element i to the front of the list. When it does the move, your program should output the index of the element being moved. (Zero-based indexing is used, so the front of the list has index 0.)

In this way, a sequence of m numbers in the range $[0, n-1]$ is transformed into another such sequence. This transformation (and its inverse) is useful in implementing a certain data compression algorithm.

So, for example, say $n = 5$. The initial list is $[0, 1, 2, 3, 4]$. If $\texttt{mtf}(3)$ is executed, then the list becomes $[3, 0, 1, 2, 4]$, and 3 is output. If the next request is 3, then the list stays the same and 0 is output. Finally if the next request is $\texttt{mtf}(4)$ then the list becomes $[4, 3, 0, 1, 2]$ and 4 is output.

Input

The first line contains blank-separated numbers n and m with $1 \le n \le 10^6$ and $1 \le m \le 5 \times 10^5$. The second line consists of the numbers $p_1, p_2, \ldots, p_m$, each of which is in the range $[0, n-1]$. These are the items to which the move-to-front operation is applied.

Output

The output consists of one line containing m space-separated numbers. The ith of these is the index of the item p_i when it is requested.

Examples

input
```
5 3
3 3 4
```
output
```
3 0 4
```

input
```
6 8
5 0 4 0 3 0 2 0
```
output
```
5 1 5 1 5 1 5 1
```

B. Letters

time limit per test: 2 seconds
memory limit per test: 256 megabytes

Little Johnny has a very long surname. Yet he is not the only such person in his milieu. As it turns out, one of his friends from kindergarten, Mary, has a surname of the same length, though different from Johnny's. Moreover, their surnames contain precisely the same number of letters of each kind – the same number of letters A, same of letters B, and so on.

Johnny and Mary took to one another and now often play together. One of their favorite games is to gather a large number of small pieces of paper, write successive letters of Johnny's surname on them, and then shift them so that they obtain Mary's surname in the end.

Since Johnny loves puzzles, he has begun to wonder how many swaps of adjacent letters are necessary to turn his surname into Mary's. For a child his age, answering such question is a formidable task. Therefore, soon he has asked you, the most skilled programmer in the kindergarten, to write a program that will help him.

Input

In the first line of the standard input there is a single integer n ($2 \le n \le 10^6$), denoting the length of Johnny's surname. The second line contains Johnny's surname itself, i.e., contains its n successive letters (without spaces). The third line contains Mary's surname in the same format: a string of n letters (with no spaces either). Both strings consist only of capital (upper-case) letters of the English alphabet.

Output

Your program should print a single integer to the standard output: the minimum number of swaps of adjacent letters that transforms Johnny's surname into Mary's.

Examples

input	output
3 ABC BCA	2

input	output
4 AACB BACA	4

C. Closest Equals

time limit per test: 3 seconds

memory limit per test: 256 megabytes

You are given sequence $a_1, a_2, ..., a_n$ and m queries l_j, r_j ($1 \le l_j \le r_j \le n$). For each query you need to print the minimum distance between such pair of elements a_x and a_y ($x \ne y$), that:

- both indexes of the elements lie within range $[l_j, r_j]$, that is, $l_j \le x, y \le r_j$;
- the values of the elements are equal, that is $a_x = a_y$.

The text above understands distance as $|x - y|$.

Input

The first line of the input contains a pair of integers n, m ($1 \le n, m \le 5 \cdot 10^5$) — the length of the sequence and the number of queries, correspondingly.

The second line contains the sequence of integers $a_1, a_2, ..., a_n$ ($-10^9 \le a_i \le 10^9$).

Next m lines contain the queries, one per line. Each query is given by a pair of numbers l_j, r_j ($1 \le l_j \le r_j \le n$) — the indexes of the query range limits.

Output

Print m integers — the answers to each query. If there is no valid match for some query, please print `-1` as an answer to this query.

Examples

input

```
5 3
1 1 2 3 2
1 5
2 4
3 5
```

output

```
1
-1
2
```

input

```
6 5
1 2 1 3 2 3
4 6
1 3
2 5
2 4
1 6
```

output

```
2
2
3
-1
2
```

D. Elections

time limit per test: 2 seconds

memory limit per test: 256 megabytes

You are running for a governor in a small city in Russia. You ran some polls and did some research, and for every person in the city you know whom he will vote for, and how much it will cost to bribe that person to vote for you instead of whomever he wants to vote for right now. You are curious, what is the smallest amount of money you need to spend on bribing to win the elections. To win elections you need to have strictly more votes than any other candidate.

Input

First line contains one integer n ($1 \leq n \leq 10^5$) — number of voters in the city. Each of the next n lines describes one voter and contains two integers a_i and b_i ($0 \leq a_i \leq 10^5$; $0 \leq b_i \leq 10^4$) — number of the candidate that voter is going to vote for and amount of money you need to pay him to change his mind. You are the candidate 0 (so if a voter wants to vote for you, a_i is equal to zero, in which case b_i will also be equal to zero).

Output

Print one integer — smallest amount of money you need to spend to win the elections.

Examples

input

```
5
1 2
1 2
1 2
2 1
0 0
```

output

```
3
```

input

```
4
1 2
1 2
2 1
0 0
```

output

```
2
```

input

```
1
100000 0
```

output

```
0
```

E. Count Subarrays

time limit per test: 1 second
memory limit per test: 512 megabytes

Given an array of N integers A_1, A_2 ... A_N and an integer K, count number of subarrays of A such that number of inversions in those subarrays is at least K.

Inversions in a subarray $A_i, A_{i+1}, ..., A_j$ is defined as number of pairs (a, b) such that $i \le a < b \le j$ and $A_a > A_b$.

Input

First line consists of N and K in single line. Next line contains N space separated integers denoting array A.

Output

Print the required answer in one line.

Constraints

- $1 \le N \le 10^5$
- $0 \le A_i \le 10^9$
- $0 \le K \le N * (N - 1) / 2$

Examples

input

```
4 1
1 2 4 0
```

output

```
3
```

input

```
2 0
1 2
```

output

```
3
```

Note

Let's denote by $A[i, j]$ the subarray A_i, A_{i+1} ... A_j.

Example 1. $A[1, 4]$, $A[2, 4]$, $A[3, 4]$ have at least 1 inversion.

Example 2. All subarrays are valid.