Codeforces Round #316 (Div. 2)

A. Elections

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

The country of Byalechinsk is running elections involving n candidates. The country consists of m cities. We know how many people in each city voted for each candidate.

The electoral system in the country is pretty unusual. At the first stage of elections the votes are counted for each city: it is assumed that in each city won the candidate who got the highest number of votes in this city, and if several candidates got the maximum number of votes, then the winner is the one with a smaller index.

At the second stage of elections the winner is determined by the same principle over the cities: the winner of the elections is the candidate who won in the maximum number of cities, and among those who got the maximum number of cities the winner is the one with a smaller index.

Determine who will win the elections.

Input

The first line of the input contains two integers n, m $(1 \le n, m \le 100)$ — the number of candidates and of cities, respectively.

Each of the next m lines contains n non-negative integers, the j-th number in the i-th line a_{ij} $(1 \le j \le n, 1 \le i \le m, 0 \le a_{ij} \le 10^9)$ denotes the number of votes for candidate j in city i.

It is guaranteed that the total number of people in all the cities does not exceed 10^9.

Output

Print a single number — the index of the candidate who won the elections. The candidates are indexed starting from one.

Sample test(s)

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

input
3 4 10 10 3 5 1 6 2 2 2 1 5 7
output
1

Note

Note to the first sample test. At the first stage city 1 chosen candidate 3, city 2 chosen candidate 2, city 3 chosen candidate 2. The winner is candidate 2, he gained 2 votes.

Note to the second sample test. At the first stage in city 1 candidates 1 and 2 got the same maximum number of votes, but candidate 1 has a smaller index, so the city chose candidate 1. City 2 chosen candidate 3. City 3 chosen candidate 1, due to the fact that everyone has the same number of votes, and 1 has the smallest index. City 4 chosen the candidate 3. On the second stage the same number of cities chose candidates 1 and 3. The winner is candidate 1, the one with the smaller index.

B. Simple Game

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

One day Misha and Andrew were playing a very simple game. First, each player chooses an integer in the range from 1 to n. Let's assume that Misha chose number m, and Andrew chose number a.

Then, by using a random generator they choose a random integer c in the range between 1 and n (any integer from 1 to n is chosen with the same probability), after which the winner is the player, whose number was closer to c. The boys agreed that if m and a are located on the same distance from c, Misha wins.

Andrew wants to win very much, so he asks you to help him. You know the number selected by Misha, and number n. You need to determine which value of a Andrew must choose, so that the probability of his victory is the highest possible.

More formally, you need to find such integer a ($1 \le a \le n$), that the probability that $|c - a| < |c - m|$ is maximal, where c is the equiprobably chosen integer from 1 to n (inclusive).

Input

The first line contains two integers n and m ($1 \le m \le n \le 10^9$) — the range of numbers in the game, and the number selected by Misha respectively.

Output

Print a single number — such value a, that probability that Andrew wins is the highest. If there are multiple such values, print the minimum of them.

Sample test(s)

input
3 1
output
2

input
4 3
output
2

Note

In the first sample test: Andrew wins if c is equal to 2 or 3. The probability that Andrew wins is $2 / 3$. If Andrew chooses $a = 3$, the probability of winning will be $1 / 3$. If $a = 1$, the probability of winning is 0.

In the second sample test: Andrew wins if c is equal to 1 and 2. The probability that Andrew wins is $1 / 2$. For other choices of a the probability of winning is less.

C. Replacement

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

Daniel has a string s, consisting of lowercase English letters and period signs (characters '.'). Let's define the operation of *replacement* as the following sequence of steps: find a substring ".." (two consecutive periods) in string s, of all occurrences of the substring let's choose the first one, and replace this substring with string ".". In other words, during the replacement operation, the first two consecutive periods are replaced by one. If string s contains no two consecutive periods, then nothing happens.

Let's define $f(s)$ as the minimum number of operations of *replacement* to perform, so that the string does not have any two consecutive periods left.

You need to process m queries, the i-th results in that the character at position x_i ($1 \le x_i \le n$) of string s is assigned value c_i. After each operation you have to calculate and output the value of $f(s)$.

Help Daniel to process all queries.

Input

The first line contains two integers n and m ($1 \le n, m \le 300\,000$) the length of the string and the number of queries.

The second line contains string s, consisting of n lowercase English letters and period signs.

The following m lines contain the descriptions of queries. The i-th line contains integer x_i and c_i ($1 \le x_i \le n$, c_i — a lowercas English letter or a period sign), describing the query of assigning symbol c_i to position x_i.

Output

Print m numbers, one per line, the i-th of these numbers must be equal to the value of $f(s)$ after performing the i-th assignment.

Sample test(s)

input

```
10 3
.b..bz....
1 h
3 c
9 f
```

output

```
4
3
1
```

input

```
4 4
.cc.
2 .
3 .
2 a
1 a
```

output

```
1
3
1
1
```

Note

Note to the first sample test (replaced periods are enclosed in square brackets).

The original string is ".b..bz....".

- after the first query $f($hb..bz....$) = 4$ ("hb[..]bz...." → "hb.bz[..].." → "hb.bz[..]." → "hb.bz[..]" → "hb.bz.")
- after the second query $f($hbc.bz....$) = 3$ ("hbc.bz[..].." → "hbc.bz[..]." → "hbc.bz[..]" → "hbc.bz.")
- after the third query $f($hbc.bz..f.$) = 1$ ("hbc.bz[..]f." → "hbc.bz.f.")

Note to the second sample test.

The original string is ".cc.".

- after the first query: $f($..c.$) = 1$ ("[..]c." → ".c.")
- after the second query: $f($....$) = 3$ ("[..].." → "[..]." → "[..]" → ".")
- after the third query: $f($.a..$) = 1$ (".a[..]" → ".a.")
- after the fourth query: $f($aa..$) = 1$ ("aa[..]" → "aa.")

D. Tree Requests

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

Roman planted a tree consisting of n vertices. Each vertex contains a lowercase English letter. Vertex 1 is the root of the tree, each of the n - 1 remaining vertices has a *parent* in the tree. Vertex is connected with its parent by an edge. The parent of vertex i is vertex p_i, the parent index is always less than the index of the vertex (i.e., $p_i < i$).

The *depth* of the vertex is the number of nodes on the path from the root to v along the edges. In particular, the depth of the root is equal to 1.

We say that vertex u is in the *subtree* of vertex v, if we can get from u to v, moving from the vertex to the parent. In particular, vertex v is in its subtree.

Roma gives you m queries, the i-th of which consists of two numbers v_i, h_i. Let's consider the vertices in the subtree v_i located at depth h_i. Determine whether you can use the letters written at these vertices to make a string that is a *palindrome*. The letters that are written in the vertexes, can be rearranged in any order to make a palindrome, but all letters should be used.

Input

The first line contains two integers n, m ($1 \le n, m \le 500\,000$) — the number of nodes in the tree and queries, respectively.

The following line contains n - 1 integers $p_2, p_3, ..., p_n$ — the parents of vertices from the second to the n-th ($1 \le p_i < i$).

The next line contains n lowercase English letters, the i-th of these letters is written on vertex i.

Next m lines describe the queries, the i-th line contains two numbers v_i, h_i ($1 \le v_i, h_i \le n$) — the vertex and the depth that appear in the i-th query.

Output

Print m lines. In the i-th line print "`Yes`" (without the quotes), if in the i-th query you can make a palindrome from the letters written on the vertices, otherwise print "`No`" (without the quotes).

Sample test(s)

input
6 5 1 1 1 3 3 zacccd 1 1 3 3 4 1 6 1 1 2

output
Yes No Yes Yes Yes

Note

String s is a *palindrome* if reads the same from left to right and from right to left. In particular, an empty string is a palindrome.

Clarification for the sample test.

In the first query there exists only a vertex 1 satisfying all the conditions, we can form a palindrome "`z`".

In the second query vertices 5 and 6 satisfy condititions, they contain letters "`c`" and "`d`" respectively. It is impossible to form a palindrome of them.

In the third query there exist no vertices at depth 1 and in subtree of 4. We may form an empty palindrome.

In the fourth query there exist no vertices **in subtree of 6 at depth 1**. We may form an empty palindrome.

In the fifth query there vertices 2, 3 and 4 satisfying all conditions above, they contain letters "`a`", "`c`" and "`c`". We may form a palindrome "`cac`".

E. Pig and Palindromes

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Peppa the Pig was walking and walked into the forest. What a strange coincidence! The forest has the shape of a rectangle, consisting of n rows and m columns. We enumerate the rows of the rectangle from top to bottom with numbers from 1 to n, and the columns — from left to right with numbers from 1 to m. Let's denote the cell at the intersection of the r-th row and the c-th column as (r, c).

Initially the pig stands in cell $(1, 1)$, and in the end she wants to be in cell (n, m). Since the pig is in a hurry to get home, she can go from cell (r, c), only to either cell $(r + 1, c)$ or $(r, c + 1)$. She cannot leave the forest.

The forest, where the pig is, is very unusual. Some cells of the forest similar to each other, and some look very different. Peppa enjoys taking pictures and at every step she takes a picture of the cell where she is now. The path through the forest is considered to be *beautiful* if photographs taken on her way, can be viewed in both forward and in reverse order, showing the same sequence of photos. More formally, the line formed by the cells in order of visiting should be a *palindrome* (you can read a formal definition of a palindrome in the previous problem).

Count the number of beautiful paths from cell $(1, 1)$ to cell (n, m). Since this number can be very large, determine the remainder after dividing it by $10^9 + 7$.

Input

The first line contains two integers n, m $(1 \le n, m \le 500)$ — the height and width of the field.

Each of the following n lines contains m lowercase English letters identifying the types of cells of the forest. Identical cells are represented by identical letters, different cells are represented by different letters.

Output

Print a single integer — the number of beautiful paths modulo $10^9 + 7$.

Sample test(s)

input
3 4 aaab baaa abba
output
3

Note

Picture illustrating possibilities for the sample test.