

Dictionary 2

Input file:	standard input
Output file:	standard output
Time limit:	4 seconds
Memory limit:	1024 megabytes

You are given a rooted tree with n vertices, where the root is vertex 1. Each vertex u is assigned a lowercase English character c_u . For any vertex u , let $S(u)$ denote the string formed by concatenating the characters on the path from u up to the root. More formally:

- If $u = 1$, then $S(u) = c_u$.
- If v is a child of u , then $S(v) = c_v + S(u)$, where $+$ denotes string concatenation.

Let \mathcal{V} be the set of all vertices in the tree. Let \mathcal{D} be the set of all nonempty substrings of the strings in $\{S(u) \mid u \in \mathcal{V}\}$. That is, $w \in \mathcal{D}$ if and only if w is a nonempty substring of $S(u)$ for some vertex u . Recall that sets do not contain duplicate elements.

We define the lexicographical order \leq_{lex} as follows: for two strings A and B , $A \leq_{lex} B$ if and only if A is a prefix of B or, at the first position where they differ, the character in A is smaller than the corresponding character in B .

You need to process q operations sequentially. For each operation, you are given an encoded integer u' and a character c . Let N be the number of vertices before this operation, and let l be the integer output by the previous operation (for the first operation, $l = 0$). Perform the following steps:

1. Calculate the real parent vertex $u = ((u' + l) \bmod N) + 1$. Create a new vertex with index $(N + 1)$ as a child of u with character c , forming the new string $S(N + 1) = c + S(u)$. Update the set \mathcal{D} to include all nonempty substrings of $S(N + 1)$ that were not previously present in \mathcal{D} .
2. Output the number of strings in the updated set \mathcal{D} that are lexicographically less than or equal to $S(N + 1)$. More formally, output the value:

$$|\{w \in \mathcal{D} \mid w \leq_{lex} S(N + 1)\}|$$

Input

There is only one test case in each test file.

The first line contains two integers n and q ($1 \leq n, q \leq 10^5$).

The second line contains a string $c_1 c_2 \dots c_n$ consisting of lowercase English letters, where c_i is the character of vertex i .

For the following $(n - 1)$ lines, the i -th line contains two integers u_i and v_i ($1 \leq u_i, v_i \leq n$), indicating an edge connecting vertices u_i and v_i .

For the following q lines, the i -th line contains an integer u'_i ($1 \leq u'_i \leq N$, where N is the number of vertices before this operation) and a lowercase English letter c_i , indicating the i -th operation.

Output

For each operation, output one line containing an integer, indicating the answer.

Example

standard input	standard output
2 3	5
ab	2
1 2	8
1 b	
1 a	
2 c	

Note

We explain the sample test case below.

Initially, $S(1) = a$, $S(2) = ba$, and $\mathcal{D} = \{a, b, ba\}$.

1. Input $u' = 1, c = b$. The decrypted parent is $u = 2$. Create vertex 3, $S(3) = bba$. Added substrings bb, bba . There are 5 strings in \mathcal{D} with $\leq_{\text{lex}} bba$.
2. Input $u' = 1, c = a$. The decrypted parent is $u = 1$. Create vertex 4, $S(4) = aa$. Added substring aa . There are 2 strings in \mathcal{D} with $\leq_{\text{lex}} aa$.
3. Input $u' = 2, c = c$. The decrypted parent is $u = 1$. Create vertex 5, $S(5) = ca$. Added substrings c, ca . There are 8 strings in \mathcal{D} with $\leq_{\text{lex}} ca$.