AC Automaton

Input file:	`standard input`
Output file:	`standard output`
Time limit:	8 seconds
Memory limit:	512 megabytes

JB is trying to get more "AC" on problems with his AC Automaton.

The AC Automaton is a rooted tree with n nodes and node 1 as root. Each node i except the root has its unique parent p_i and a character s_i on it, which is one of '`A`', '`C`', or '`?`'. A node x is called an ancestor of y if and only if $x = p_y$ or x is an ancestor of p_y. The number of "AC" JB can get equal to the number of ordered pairs (x, y) that x is an ancestor of y, $s_x =$ '`A`' and $s_y =$ '`C`'. JB can replace '`?`' arbitrarily with '`A`' or '`C`'. His goal is to get more "AC" after replacing all '`?`'.

However, the problem always changes. JB will change his AC Automaton q times. Each time he will modify the character on one of the nodes x to one of '`A`', '`C`', or '`?`' (the character will possibly not change). JB wants you to answer the maximum number of "AC" he can get if he replaces all '`?`' on his AC Automaton after each modification. Note that JB **will not** actually modify the AC Automaton while calculating the maximum number of "AC". You can refer to the sample to help you understand.

Input

The first line contains two integers n and q ($1 \le n, q \le 300\ 000$).

The second line contains a string consisting of n characters, representing the characters on each node. It's guaranteed that the characters in the string is one of '`A`', '`C`', and '`?`'.

The third line contains $n-1$ integers, the i-th integer p_i ($1 \le p_i \le i$) represents the parent of node $i+1$.

The next q lines describe the modifications. The i-th of the following line consists one integer x ($1 \le x \le n$) and a character y ($y \in \{$'`A`', '`C`', '`?`'$\}$), representing one modification.

Output

Output q lines.

In the i-th line, output one integer representing the maximum number of "AC" after the i-th modification.

Example

standard input	standard output
5 3 AC??C 1 2 2 1 1 ? 4 A 2 ?	4 3 3

Note

After the first modification, one of the best way of replacing characters is "`ACCCC`", and there are 4 pairs $(1, 2), (1, 3), (1, 4)$, and $(1, 5)$.

After the second modification, one of the best way of replacing characters is "`ACCAC`", and there are 3 pairs $(1, 2), (1, 3)$, and $(1, 5)$.

After the third modification, one of the best way of replacing characters is "`AACAC`", and there are 3 pairs $(1, 3), (2, 3)$, and $(1, 5)$.