Normal Friends

Input file: `standard input`
Output file: `standard output`
Time limit: 8 seconds
Memory limit: 1024 megabytes

Segment trees are a data structure that Little Q is very fond of. They have simple structures, excellent time complexities, and powerful functionalities, so Little Q once spent a long time studying some properties of segment trees.

Recently, Little Q has started studying segment trees again, but with a difference: she has focused on more generalized segment trees. In a normal segment tree, for an interval $[l, r]$, we would take $\text{mid} = \left\lfloor \frac{l+r}{2} \right\rfloor$, and then split this interval into $[l, \text{mid}]$ and $[\text{mid}+1, r]$. In generalized segment trees, mid is not required to be exactly the midpoint of the interval, but mid still must satisfy $l \le \text{mid} < r$. It is not hard to find that in generalized segment trees, the depth of the tree can reach $O(n)$ levels.

Little Q doesn't know how to implementer balanced BSTs, so she wants to use segment trees to implement all operations of balanced BSTs. One of the most famous operations that cannot be implemented using segment trees is reversing an interval, but Little Q is not convinced and wants to implement it using segment trees.

Specifically, when Little Q reverses an interval, she will first partition this interval into several maximal intervals represented by nodes in the segment tree; suppose from smallest to largest they are represented by the k nodes $a_1, a_2, \ldots, a_k$ in the segment tree. Then, Little Q will strip these k subtrees and reattach them to the segment tree in reverse order, that is, the subtree originally at the position of a_1 is replaced with the subtree of a_k, the position originally of a_2 is replaced with that of a_{k-1}, and so on; the position originally of a_k is replaced with the subtree of a_1. Finally, Little Q will swap the left and right children of all nodes in these k subtrees. It is easy to find that after such an operation, the tree is still a generalized segment tree, and the order of the elements from left to right is exactly the result after reversing the interval.

Given such a generalized segment tree, Little Q needs to perform m operations; each time, given a prefix $[1, x]$, she reverses it. At the same time, Little Q wants to know the value of k, that is, the number of intervals partitioned in the current generalized segment tree during the reverse operation.

Of course, Little Q easily solved this problem, which made her more confident not to learn how to write balanced BSTs. So, can you solve it?

Input

The first line contains two positive integers n and m ($2 \le n, m \le 3 \times 10^5$), representing the size of the tree and the number of operations.

The next line contains $n - 1$ positive integers, given in depth-first search (DFS) order, providing the splitting point mid for each interval.

The next m lines each contain a positive integer x, representing an operation of reversing the prefix $[1, x]$.

Output

Output m lines, each line containing a positive integer, representing the value of k during each operation.

Example

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