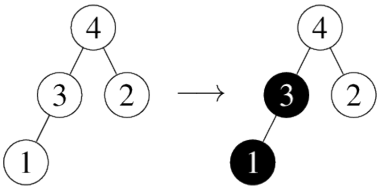
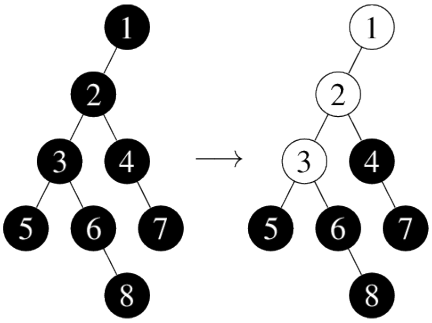
Ball Machine

We have a “ball machine” that can be visualized as a rooted tree. The nodes of the tree are numbered from 1 to N. Each node is either empty or contains one ball. Initially, all nodes are empty. While running, the machine can perform operations of two different types:

1. Add k balls to the ball machine: Balls are put one by one into the root node. As long as a ball has an empty node directly beneath it, it will roll down. If there are multiple empty child nodes, the ball will choose the one which has the node with the smallest number in its subtree. So if the ball rolls down multiple levels, it makes a choice at each level. For example: If we add two balls to the machine in the picture below, they will go to nodes 1 and 3: The first ball rolls from node 4 to node 3 because node 3 is empty and it contains node 1 in its subtree (which consists of nodes 3 and 1); it further rolls from node 3 to node 1. The second ball rolls from node 4 to node 3 as well and stops there.



2. Remove a ball from a specified node: This node becomes empty and balls from above (if there are any) roll down. Whenever a parent of an empty node contains a ball, this ball will roll down. If we remove balls from nodes 5, 7 and 8 (in this order) from the machine in the picture below, nodes 1, 2 and 3 will become empty.



Input

The first line contains two integers N and Q – the number of tree nodes and the number of operations. The next N lines describe the ball machine. Each of these lines contains one integer, the number of a node: the i-th of these lines contains the number of node i’s parent node, or 0 if node i is the tree root. Each of the next Q lines contains two integers and describes an operation to be performed. An operation of type 1 is denoted by 1 k where k is the number of balls to be added to the machine. An operation of type 2 is denoted by 2 x where x is the number of the node from which a ball is to be removed. It is guaranteed that all performed operations are correct: Operations will not require to add more balls than there are empty nodes in the ball machine or to remove a ball from an empty node.

Output

For each operation of type 1, output the number of the node where the last inserted ball ended up. For each operation of type 2 output the number of balls that rolled down after removing the ball from the specified node.

Constraints

It is always $N, Q \leq 100\,000$.

In test cases worth 25 points, each node has either 0 or 2 children. Furthermore, all nodes with 0 children have the same distance from the root.

In test cases worth 30 points, the operations will be requested in such a way that no balls ever roll down after an operation of type 2.

In test cases worth 40 points, there is exactly one operation of type 1, and it is the very first operation.

These three sets of test cases are pairwise disjoint.

Example

Input	Output
8 4	1
0	3
1	2
2	2
2	
3	
3	
4	
6	
1 8	
2 5	
2 7	
2 8	

Limits

Time limit:	1 sec per test case
Memory limit:	128 MB per test case