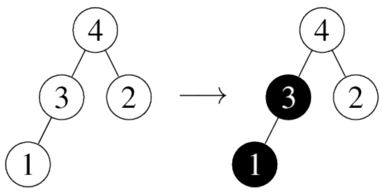
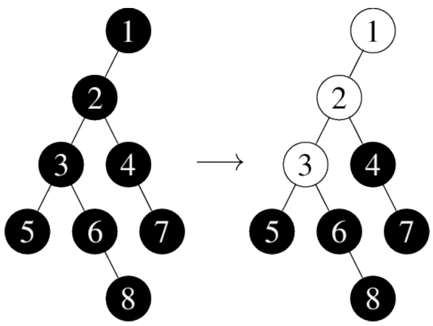
Ball Machine

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

Limits

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

Palindrome-Free Numbers

A string is a palindrome if it remains the same when it is read backwards. A number is palindrome-free if it does not contain a palindrome with a length greater than 1 as a substring. For example, the number 16276 is palindrome-free whereas the number 17276 is not because it contains the palindrome 727.

Your task is to calculate the total number of palindrome-free numbers in a given range.

Input

The input contains two integers, a and b.

Output

The output should contain one integer: the total number of palindrome-free numbers in the range $a, \ldots, b$ (including a and b).

Constraints

$0 \leq a \leq b \leq 10^{18}$

In test cases worth 25 points: $b - a \leq 100\,000$.

Examples

Input	Output
`123 321`	`153`
`123456789 987654321`	`167386971`

Limits

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

Pipes

The city of Hotham is once again attacked by its most prominent villain, the Jester. This time his target is Hotham's water supply. The fresh water of Hotham is stored in N reservoirs, which are connected by a set of M pipes. There is at least one path (potentially consisting of several pipes) from any reservoir to any other reservoir. Moreover, every pipe connects two different reservoirs, and there is at most one pipe between any pair of reservoirs.

The Jester has breached some of the pipes and has been draining water from them. Following his playful nature, the Jester ensured that water drained from any one pipe amounts to an even number of cubic meters per second (m^3/s). If $2d\ m^3/s$ of water is drained from a pipe joining reservoirs u and v, then u and v lose $d\ m^3/s$ of water each.

To make matters more confusing, the Jester actually pumps water into some of the breached pipes instead of draining from them. Again, the water pumped into any one pipe is an even number of m^3/s. If $2p\ m^3/s$ of water is pumped into a pipe joining reservoirs u and v, then u and v gain $p\ m^3/s$ of water each. The net change of water volume in each reservoir is the total sum of gains and losses acquired from the pipes connected to it. Formally, if a reservoir is connected to pipes from which $2d_1$, $2d_2$, $\ldots$, $2d_a\ m^3/s$ of water is drained and to pipes into which $2p_1$, $2p_2$, $\ldots$, $2p_b\ m^3/s$ of water is pumped, then the net change of water volume in this reservoir is $p_1+p_2+\ldots+p_b-d_1-d_2-\ldots-d_a$.

The mayor of Hotham has installed sensors in the reservoirs, but not in the pipes. Therefore, he can observe the net change of water in each reservoir but does not how much water is drained from or pumped into each pipe.

Your task is to write a program that helps the mayor. Given full information about the reservoir network and the net changes in each reservoir, your program should decide if this information is enough to uniquely determine the Jester's plan. The plan can be determined uniquely if there is exactly one possibility for how much water is drained from or pumped into each pipe. Note that these amounts of water need not be the same for all pipes. If there is exactly one possibility, your program should print it.

Input

The first line of the input contains two integers: N, the number of reservoirs in Hotham, and M, the number of pipes. The following N lines contain an integer c_i each: the net change in reservoir i ($1 \le i \le N$). Line i of these N lines contains c_i. The following M lines contain two integers u_i and v_i each ($1 \le i \le M$). Each such line indicates that there is a pipe between resevoirs u_i and v_i ($1 \le u_i, v_i \le N$). Line i of these M lines contains u_i and v_i.

The input always describes a set of reservoir changes that can be realized by the Jester.

Output

If the Jester's plan cannot be determined uniquely, your program should output a single line containing 0. Otherwise, your program should output M lines with one integer x_i each ($1 \le i \le M$). Line i should contain x_i. If the Jester drains $d_i\ m^3/s$ of water from the pipe between u_i and v_i, let $x_i = -d_i$. If the Jester pumps $p_i\ m^3/s$ of water into the pipe between u_i and v_i, let $x_i = p_i$. If the Jester does not add or remove water from the pipe between u_i and v_i, let $x_i = 0$.

Constraints

$1 \leq N \leq 100000$
$1 \leq M \leq 500000$
$-10^9 \leq c_i \leq 10^9$
If the Jester's plan can be determined uniquely, $-10^9 \leq x_i \leq 10^9$.

In test cases worth 30 points, the water network of Hotham is a tree.

Examples

Input	Output
4 3 -1 1 -3 1 1 2 1 3 1 4	2 -6 2
4 5 1 2 1 2 1 2 2 3 3 4 4 1 1 3	0

Limits

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