Agent 007700 has a tree with $n$ vertices and a root vertex with number $1$. In each vertex there is a value - $x\_{1}, x\_{2}, … , x\_{n}$. Unfortunately for him, the DoS (Department of Security) agent Bozhil gained access to it with the aim of sabotage. Events of 2 types occur:

1. $Update$ with parameters $node$ and $value$. Then agent Bozhil changes $x\_{node}$ and its values becomes equal to $value$
2. $Query$ with parameter $node$. Then agent 007700 wants to find the bitwise “excluding or” (xor) value of the values $x$ of $node$ and all its direct and indirect ancestors.

**Input**

The first line of the file **xor.in** contains the numbers $n$ and $q$. The next line contains$ n$ numbers - $x\_{1}, x\_{2}, … , x\_{n}$ – the values of the vertices. The next $n-1$ lines contain 2 natural numbers each - $u, v$ – the edges of the tree. The next $q$ lines contain 2 numbers each - $type and node$

If $type=1$, then the query type is $Update$ and the line contains a third number $value$

If $type=2$, then the query type is $Query$

**Output**

For each query of type 2, print on a new line in the **xor.out** file the answer for that query.

**Constraints**

$$1\leq n\leq 20 000$$

$$1\leq q\leq 50 000$$

$$1\leq x\_{i}, value\leq 10^{9}$$

$$1\leq type\leq 2$$

$$1\leq node\leq n$$

 **Time limit: 0.7 sec.**

 **Memory limit: 256 MB.**

**Sample test**

|  |  |
| --- | --- |
| **Input (xor.in)** | **Output (xor.out)** |
| 4 31 2 4 81 22 33 42 41 2 52 3 | 150 |