Shashank loves trees and math. He has a rooted tree, T, consisting of N nodes uniquely labeled with integers in the inclusive range $[1, N]$. The node labeled as 1 is the *root* node of tree T, and each node in T is associated with some positive integer value (all values are initially 0).

Let's define F_k as the k^{th} Fibonacci number. Shashank wants to perform 2 types of operations over his tree, T:

1. $U\ X\ k$
   Update the subtree rooted at node X such that the node at level 0 in subtree X (i.e., node X) will have F_k added to it, all the nodes at level 1 will have F_{k+1} added to them, and so on. More formally, all the nodes at a distance D from node X in the subtree of node X will have the $(k+D)^{th}$ Fibonacci number added to them.
2. $Q\ X\ Y$
   Find the sum of all values associated with the nodes on the unique path from X to Y. Print your sum modulo $10^9 + 7$ on a new line.

Given the configuration for tree T and a list of M operations, perform all the operations efficiently.

Note: $F_1 = F_2 = 1$.

Input Format

The first line contains 2 space-separated integers, N (the number of nodes in tree T) and M (the number of operations to be processed), respectively.
Each line i of the $N - 1$ subsequent lines contains an integer, P, denoting the parent of the $(i+1)^{th}$ node.
Each of the M subsequent lines contains one of the two types of operations mentioned in the *Problem Statement* above.

Constraints

- $1 \le N, M \le 10^5$
- $1 \le X, Y \le N$
- $1 \le k \le 10^{15}$

Output Format

For each operation of type 2 (i.e., Q), print the required answer modulo $10^9 + 7$ on a new line.

Sample Input

```
5 10
1
1
2
2
Q 1 5
U 1 1
Q 1 1
Q 1 2
Q 1 3
Q 1 4
Q 1 5
U 2 2
Q 2 3
Q 4 5
```

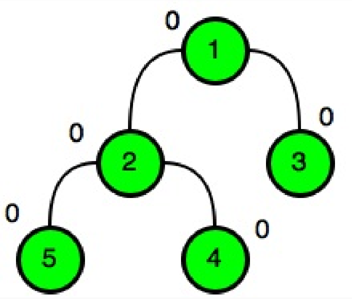
Sample Output

```
0
1
2
2
4
4
```

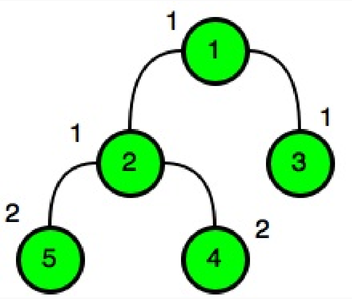
```
4
10
```

Explanation

Intially, the tree looks like this:



After update operation $1\ \ 1$, it looks like this:



After update operation $2\ \ 2$, it looks like this: