Task: FAL
Falling Cats

AACPP SuSe 2025 Round 3 Memory: 128MiB 2025.05.27 – 2025.06.03

Dexter is taking a walk and encounters a peculiar sight. A cat is holding a branch of a very high tree with its paws. Below, there is n more cats holding each other in various ways. Each cat has **four paws** and can grasp a different cat with each, or have it free.

The cats don't have infinite strength, so as time goes on they release their paws. Naturally, if they lose connection to the topmost cat holding the branch they will fall to the ground due to Isaac Newton – don't worry, cats always fall on all fours and the tree is not that high.

Dexter knows his kind really well and so can predict exactly which cat will release which paw at which time. Knowing that, he'd like to predict **at which time each cat will fall to the ground**.

Input

The first line of input contains two integers n, m, the number of cats and the length of time.

The next $n-1$ lines contain the paw configuration. The i-th line contains four integers a_i, b_i, c_i, d_i, describing which other cats the $i+1$-th cat is holding onto – either the number of the cat, or -1 meaning the paw is free.

The final m lines contain description of paw releases. The i-th line contains two integers, v_i, e_i, where v_i is the number of a cat and e_i is equal to 1, 2, 3, or 4, and means that the cat v_i released his e_i-th paw at moment i.

Note that: cat number 1 never releases the tree branch; cats can still release their paws after they've fallen to the ground; a cat can't hold itself; a cat can hold the same cat in multiple paws; no cats are initially on the ground.

Output

Your program should write $n-1$ lines to the standard output. The i-th line should contain the timestamp at which cat $i+1$ falls to the ground, or -1 if it does not happen.

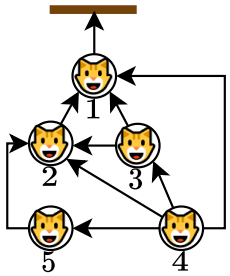
Example

For the input:

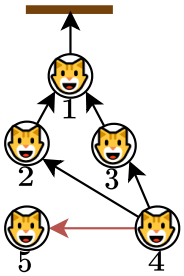
```
5 6
1 -1 -1 -1
1 2 -1 -1
5 2 3 1
2 -1 -1 -1
4 4
5 1
3 2
4 1
2 1
3 1
```

the correct output is:

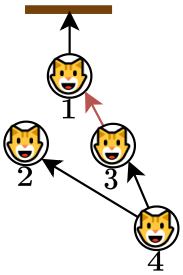
```
6
6
6
4
```



At the start ($t = 0$)



5 falls ($t = 4$)



$2, 3, 4$ fall ($t = 6$)

Additional examples

The following initial tests are also available:

- `0b` – small sample, $n = 6$,$m = 5$, some cats never fall;
- `0c` – $n = 200, m = 20$, cat i holds cats $2 \cdot i - 1$ and $2 \cdot i$ in its paws, cat 2 additionally holds the cat 3;
- `0d` – sample for Subtask 2, $n = 2\,000, m = 500$, cat $i + 1$ holds the cat i in its first paw, then each even cat let's go in order from the bottom;
- `0e` – sample for Subtask 3, $n = 250\,000, m = 999\,996$, all cats hold 1 in all their paws.

Limits

Your solution will be evaluated on a number of hidden test cases divided into groups. Points for a group are awarded if and only if the submission returns the correct answer for each of the tests in the group within the allotted time limit. These groups are organised into subtasks with the following limits and points awarded.

Subtask	Limits	Points
1.	$1 \leq n \leq 200$, $1 \leq m \leq 800$	1
2.	$1 \leq n \leq 2\,000$, $1 \leq m \leq 8\,000$	2
3.	$1 \leq n \leq 250\,000$, $1 \leq m \leq 1\,000\,000$	7