Problem E: Feeding the Cats

One of the cats calls for your attention.

Having a side job to make a little extra money is always appreciated, especially if the job description sounds easy. “Help wanted to feed my cats. – Ms Purr” – How hard can it be?

Turns out that it is a bit more complicated than it appeared. Firstly, Ms Purr has a lot of cats. Polite as you are, you listened to her listing off all their individual names, but after a short while you gave up and just numbered the cats from 0 to $n-1$. Secondly, cats can get really jealous if they see that other cats are fed before they are, so they might start stealing food. However, you noticed that there is a cat rank order. Cats of lower rank will never dare to steal food from cats of higher rank. Therefore, the only way to keep peace between the cats is to feed them in descending order of rank (conveniently, no two cats have the same rank).

In the past couple of days, while already performing your job, you made a large number of memos to remind yourself which cats must be fed before others. Every memo describes exactly two different cats and which of those has the higher rank. You know for sure that with those memos you can reconstruct the correct rank order and now you are wondering if you always have to bring all those heavy notes with you. Wouldn’t it be easier to just leave the unnecessary ones at home? You start by giving the memos consecutive indices from 0 to $m-1$. Which memos can be removed without making it impossible to reconstruct the correct rank order?

Input

The input consists of:

- One line with two integers n and m ($1 \leq n \leq 100\,000, n-1 \leq m \leq 200\,000$), which describe the number of cats and the number of memos.
- m lines of the form a `>` b ($0 \leq a, b < n; a \neq b$), which means cat a is of higher rank than cat b. The memos are listed with ascending index, from memo 0 to memo $m-1$. It is guaranteed that every memo is unique and that the unique correct rank order can be reconstructed with those m memos.

Output

Output the indices of all the memos that are not needed to reconstruct the correct rank order. These indices should be printed in strictly ascending order, one per line. If no such index exists, print `meow` instead.

Sample Input 1

```
3 3
0 > 1
0 > 2
1 > 2
```

Sample Output 1

```
1
```

Sample Input 2

```
2 1
1 > 0
```

Sample Output 2

```
meow
```

Sample Input 3

```
6 8
1 > 5
0 > 3
4 > 3
1 > 3
2 > 3
5 > 4
0 > 1
2 > 0
```

Sample Output 3

```
1
3
4
```