COCI '13 Contest 4 #3 Sumo

Time limit: 1.0s **Memory limit:** 32M

In a Japanese monastery, otherwise known for serious fasting and ascetic life, the Head of the sumo wrestling section has decided to organise training competitions for his N fighters. He determined the exact sequence of M fights and its participants (two fighters face each other per fight).

Just moments before the competition, the Head realised he could easily stir things up a bit! He could divide his fighters into two teams so that only fighters of different teams face each other in each fight. Since the fighting schedule has already been made and it doesn't meet this condition, and we mustn't change it for whatever zen reason there is, the Head is left with only one option. That is to divide the fighters into two teams so that the fighters from the same team face each other in a fight as late as possible.

Help the Head! For a given fighting schedule, determine the ordinal number of the first fight where two fighters from the same team have to face each other, under the condition that we divide them in the best possible way, so that the required fight takes place as late as possible. In all test data, such fight will definitely occur.

Input Specification

The first line of input contains the integer N $(1 \leq N \leq 100\,000)$, the number of fighters. The fighters are marked with numbers from 1 to N.

The second line of input contains the integer M $(1 \leq M \leq 300\,000)$, the number of fights.

Each of the following M lines contains fights in the order which they must take place. Each line contains two different integers from the interval $[1, N]$: the labels of fighters who are going to face each other.

Output Specification

The first and only line of output must contain the ordinal number (from 1 to M) of the required fight.

Sample Input 1

```
5
5
1 2
2 3
3 4
4 5
5 1
```

Sample Output 1

```
5
```

Sample Input 2

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

Sample Output 2

```
6
```