G - Stock Chase

Input: `standard input`
Output: `standard output`

I have to admit, the solution I proposed last year for solving the bank cash crisis didn't solve the whole economic crisis. As it turns out, companies don't have that much cash in the first place. They have assets which are primarily shares in other companies. It is common, and acceptable, for one company to own shares in another. What complicates the issue is for two companies to own shares *in each other at the same time*. If you think of it for a moment, this means that each company now (indirectly) controls its own shares.

New market regulation is being implemented: No company can control shares in itself, whether directly or indirectly. The Stock Market Authority is looking for a computerized solution that will help it detect any buying activity that will result in a company controlling its own shares. It is obvious why they need a program to do so, just imagine the situation where company A buying shares in B, B buying in C, and then C buying in A. While the first two purchases are acceptable. The third purchase should be rejected since it will lead to the three companies controlling shares in themselves. The program will be given all purchasing transactions in chronological order. The program should reject any transaction that could lead to one company controlling its own shares. All other transactions are accepted.

Input

Your program will be tested on one or more test cases. Each test case is specified on $T + 1$ lines. The first line specifies two positive numbers: $(0 < N \leq 234)$ is the number of companies and $(0 < T \leq 100{,}000)$ is the number of transactions. T lines follow, each describing a buying transaction. Each transaction is specified using two numbers A and B where $(0 < A, B \leq N)$ indicating that company A wants to buy shares in company B.

The last line of the input file has two zeros.

Output

For each test case, print the following line:

k. R

Where k is the test case number (starting at one) and R is the number of transactions that should be rejected.

Sample Input

```
3 6
1 2
1 3
3 1
```

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

Sample Output

```
1. 2
```