

It is a well known fact that some people do not have their social abilities completely enabled. One example is the lack of talent for calculating distances and intervals of time. This causes some people to always choose the longest way to go from one place to another, with the consequence that they are late to whatever appointments they have, including weddings and programming contests. This can be highly annoying for their friends.

César has this kind of problem. When he has to go from one point to another he realizes that he has to visit many people, and thus always chooses the longest path. One of César's friends, Felipe, has understood the nature of the problem. Felipe thinks that with the help of a computer he might be able to calculate the time that César is going to need to arrive to his destination. That way he could spend his time in something more enjoyable than waiting for César.

Your goal is to help Felipe developing a program that computes the length of the longest path that can be constructed in a given graph from a given starting point (César's residence). You can assume that there is at least one path from this starting point to any destination. Also, the graph has no cycles (there is no path from any node to itself), so César will reach his destination in a finite time. In the same line of reasoning, nodes are not considered directly connected to themselves.

Input

The input consists of a number of cases. The first line on each case contains a positive number n ($1 < n \leq 100$) that specifies the number of points that César might visit (i.e., the number of nodes in the graph).

A value of $n = 0$ indicates the end of the input.

After this, a second number s is provided, indicating the starting point in César's journey ($1 \leq s \leq n$). Then, you are given a list of pairs of places p and q , one pair per line, with the places on each line separated by white-space. The pair " $p q$ " indicates that César can visit q after p .

A pair of zeros ('0 0') indicates the end of the case.

As mentioned before, you can assume that the graphs provided will not be cyclic and that every place will be reachable from the starting place.

Output

For each test case you have to find the length of the longest path that begins at the starting place. You also have to print the number of the final place of such longest path. If there are several paths of maximum length, print the final place with smallest number.

Print a new line after each test case.

Sample Input

```
2
1
1 2
0 0
5
3
1 2
3 5
3 1
2 4
4 5
0 0
5
5
5 1
5 2
5 3
5 4
4 1
4 2
0 0
0
```

Sample Output

Case 1: The longest path from 1 has length 1, finishing at 2.

Case 2: The longest path from 3 has length 4, finishing at 5.

Case 3: The longest path from 5 has length 2, finishing at 1.