

# Problem C

## Maximizing Your Pay


**Problem ID:** maximizingyour

**CPU Time limit:** 1 second

**Memory limit:** 1024 MB

**Author:** Greg Hamerly

**Source:** Baylor Competitive Learning course

**License:** 

It's tougher to find work these days than it has been in the past. However, you are fortunate to have a job leading people on walking tours around various areas in your city. There are two interesting things about your job. First, your contract requires you to take people on a walking circuit that never visits the same place more than once, since otherwise your clients get bored (the notable exception is that you must deliver them to the same place they started). Second, you get paid in proportion to the amount of time you are leading each tour. Thus, you are trying to find the *longest* tours possible that don't repeat any location (other than finishing where you started). You've found that this is a difficult problem to solve by hand, even when there are a moderate number of places you can visit, so you want to write a program to solve it for you.

### Input

Input consists of a set of up to 20 test cases. Each test case starts with a line containing two integers  $n$  and  $m$ , where  $2 \leq n \leq 10$  indicates the number of locations in the city, and  $0 \leq m \leq n(n-1)/2$  indicates the number of unique streets that directly connect pairs of locations in the city. Locations are numbered 0 through  $n-1$ . The following  $m$  lines each contain a pair of integers  $x$  and  $y$ , describing that there is a street that directly connects those two locations. Each street can be walked in either direction. The last valid test case is followed by a line containing a single zero.

### Output

For each test case, output the largest number of locations you can visit on the longest walking circuit. Tours always begin and end at intersection 0.

#### Sample Input 1

```
5 6
0 1
0 2
1 2
2 3
3 4
4 0
8 8
0 1
0 2
0 3
0 4
0 5
5 6
5 7
6 7
0
```

#### Sample Output 1

```
5
2
```