B Road Trip (`roadtrip.{c,cc,java}`)

B.1 Description

Julie, who lives in City 1 on the West Coast, is about to embark on a cross-continental road trip. Her goal is to travel to City n on the East Coast, and then back to City 1. On her journey to the East Coast, she wishes to travel only on roads going from west to east; on her journey home, she wishes to travel only on roads going from east to west. Furthermore, on her journey, she does not want to visit any city twice (other than her home, City 1).

Given a list of n cities, numbered in order from west to east, and a list of roads connecting them, determine the total number of unique journeys satisfying Julie's constraints. Here, two journeys are considered different if the *ordered* list of city indices visited are not the same; that is,

$$\text{City } 1 \rightarrow \text{City } 2 \rightarrow \text{City } n \rightarrow \text{City } 3 \rightarrow \text{City } 1 \tag{3}$$

and

$$\text{City } 1 \rightarrow \text{City } 3 \rightarrow \text{City } n \rightarrow \text{City } 2 \rightarrow \text{City } 1 \tag{4}$$

are two different journeys.

B.2 Input

The input file will contain multiple test cases. Each test case will begin with a single line containing two integers n m, where n is the total number of cities (where $2 \leq n \leq 100$), and m is the total number of roads (where $1 \leq m \leq \binom{n}{2}$). The next m lines each contain two integers i j (where $1 \leq i < j \leq n$) indicating the presence of a bidirectional road between City i and City j. The end-of-input is denoted by a single line containing "0 0" and should not be processed.

```
4 6
1 2
1 3
1 4
2 3
2 4
3 4
0 0
```

B.3 Output

For each test case, print the last 9 digits of the total number of paths satisfying the constraints described above. If the last 9 digits contain any leading zeroes, remove those before printing.

```
9
```