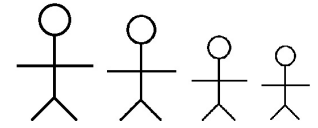


## Problem B: Classrooms

You're a professional photographer. Sadly you're fairly uncreative with the way that you arrange the participants in the photo - you think that a photograph is only acceptable if the subjects are lined up by height in descending order, from left to right. As last night you were out late doing some 'questionable' activities, today you're wearing very dark sunglasses and you can't really see. You'll therefore have to try arranging them in many different ways, so that you can guarantee yourself that you'll get an acceptable picture by the time you're done.



You aren't completely in the dark though. You know that you are organizing an even number of people, that initially came from two classrooms of equal size. You've heard some gossip from the subjects talking among themselves, and you know some relationships about the student's heights: some of the students compared their own heights to other students in their same class. Taking these into account will let you eliminate some incorrect permutations.

How many different pictures do you need to take?

### Input Specification:

Each testcase takes up several lines. On the first line of a testcase is are non-negative integers  $2 \leq N \leq 16$ , the number of people, and an integer  $R$ , the number of known relationships. Each of the next  $R$  lines contains two positive integers  $A$  and  $B$ : person number  $A$  is taller than person number  $B$ . Students numbered  $[1 \dots \frac{N}{2}]$  are in one class, and students numbered  $[\frac{N}{2} + 1 \dots N]$  are in the other class. The only relationships given are ones where both  $A$  and  $B$  are from the same class.

The input ends on a testcase where  $N = 0, R = 0$ . Do not process this case.

### Output Specification:

For each testcase, output a single integer on a line by itself: the number of times you must photograph these subjects to make sure you get the perfect picture. Remember to use a 64-bit integer data type (long in Java, long long in C/C++).

### Sample Input:

```
8 3
2 1
3 1
4 2
0 0
```

### Sample Output:

```
5040
```