Food delivery

|  |  |
| --- | --- |
| Time limit | 1 second |
| Memory limit | 244Mb |
| Input | standard input or input.txt |
| Output | standard output or output.txt |

“The Yellow Duckling” is a provider of duck food in Ducktown and he is serving n stores, numbered with the natural numbers 1 to n (*3≤ n≤ 1000*). Today “The Yellow Duckling” has enough goods to provide exactly k stores, but a lot of snow fell in Ducktown and led to a bad traffic situation. Moving around the town is very difficult: there are exactly m pairs of stores with cleared roads between them. All roads are bidirectional.

“The Yellow Duckling” wants to select its route in such way that starting from some store, moving only by cleared roads, finish at the beginning store, and it has visited exactly k stores. Also, it does not want to visit any store more than once. Now “The Yellow Duckling” wants to know how many unique routes meeting its requirements are there.

Routes differing only in the initial store or in the order of stores are counted as one.

Input format

The first line of the standard input contains two positive integer numbers – m and k (*1<m<105*, *3≤ k≤ 5*).

m pairs of separated by single space positive integer numbers *ai* and *bi* () are given in the next m lines. Each pair means that the road between *ai* and *bi* is cleared.

Output format

Print a single integer on the standard output – the number of unique routes which “The Yellow Duckling” can choose, such that he has visited exactly k stores, following the rules described above.

Sample

| **Input** | **Output** |
| --- | --- |
| 6 3  1 2  1 3  2 3  2 4  4 1  5 4 | 2 |