You are shown a graph with N vertices and M edges, which connect two vertices each. It is guaranteed that a path exists from every vertex to every other vertex using the edges. It is also known that every pair of vertices is connected by at most 1 edge and there is no edge connecting a vertex to itself.

The owner of the graph plans to remove **exactly one** vertex from the graph and that is why he is interested in how many pairs of vertices there are for which a path will always exist between them, no matter what edge is removed.

Write a program which answers his question.

**Input**

The first line of the file **connection.in** contains N and M – the number of vertices and edges. Each of the next M lines contains 2 numbers а and b, describing that there is an edge between vertices a and b.

**Output**

On the only line of the file **connection.out** print 1 number: the desired number of pairs.

**Constraints**

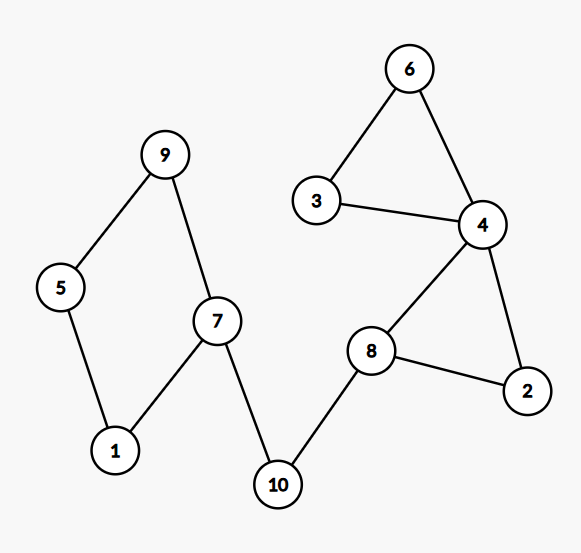
**Time Limit: 1 sec.**

**Memory Limit: 256 MB.**

**Sample Test**

|  |  |
| --- | --- |
| **Input (connection.in)** | **Output (connection.out)** |
| 10 12  10 8  4 3  2 8  4 6  5 9  5 1  1 7  10 7  2 4  6 3  4 8  7 9 | 16 |

**Sample test explanation**

Example pairs, which don’t follow the requirements are:

* 4 and 9: if the edge 10-7 or 10-8 is removed, there won’t be a path between them anymore.
* 10 and 7: if the edge 10-7 is removed, there won’t be a path between them anymore.

Example pairs, which follow the requirements are:

* 3 and 4
* 7 and 9