Task: LOO
Loopies – Zoomie Loop Challenge

AACPP WiSe 2025/26 Round 3 Memory: 512MiB 2025.11.19 – 2025.12.03

Dexter the Cat has the zoomies again. But this time they loop.

Your apartment contains n rooms, numbered from 1 to n, connected by m directed corridors. A corridor from room u to room v means Dexter can sprint only from u to v, not the other way. The apartment is fully connected: **starting in room 1, Dexter can reach every other room** (though not necessarily return).

Dexter wants to perform a **loopie**, a proper zoomie race:

- He chooses a starting room s.
- He sprints through corridors, always following their direction.
- He must move through at least one corridor.
- Eventually, he must zoom back into room s.

A room s from which Dexter can perform such a loop is called **loopie-friendly**.

Before Dexter begins his run, he may decide to spice things up in his usual chaotic manner: Dexter may choose any **sequence of pairwise distinct corridors** connected by rooms and flip the direction of **every corridor** along that path. So, if he chooses k corridors, then for each $1 \leq i \leq k$ the i-th corridor has to lead from z_{i-1} to z_i and after the rearrangement it will lead instead from z_i to z_{i-1}. There are no other restrictions, in particular the rooms z_i do not have to be different and z_0 is not necessarily the same as z_k. He can do this once per race, before he starts zooming. After this operation, all corridor directions are fixed.

Dexter, as always, requires your help:

1. Determine how many rooms are loopie-friendly in the original layout.
2. Then you are given q additional directed corridors. These do not exist yet, but Dexter sometimes finds a way where there is none. For each corridor, considered independently, compute how many rooms would be loopie-friendly if it – and *only* it – was added to the layout.

Input

The first line of input contains two integers n, m, the number of rooms and the number of corridors, respectively.

The next m lines each contain two integers $1 \leq u, v \leq n$, describing a directed corridor from room u to room v.

The next line contains a single integer q.

The next q lines each contain two integers $1 \leq x, y \leq n$, describing a proposed directed corridor to add.

Note that there might be more than one corridor connecting two given rooms, or a corridor from a room to itself. This includes the corridors in the proposed additions.

Output

Your program should output exactly $q + 1$ lines.

The first line should contain the number of loopie-friendly rooms in the original apartment.

Then, the i-th next line should contain the number of loopie-friendly rooms after adding the i-th proposed corridor.

Example

For the input:

```
5 5
1 2
2 3
3 4
2 4
4 5
3
1 3
4 5
1 5
```

the correct output is:

```
3
4
4
5
```

Explanation: In the original layout rooms $2, 3, 4$ are loopie-friendly – if Dexter decides to flip the direction of the corridor between 2 and 4 he could start and end a loopie in either of the three.

If a corridor between 1 and 3 existed, 1 would be loopie-friendly, as that corridor could be flipped. In a similar manner, the corridor connecting 4 and 5 makes 5 loopie-friendly.

Finally, if the corridor connecting 1 and 5 existed, then all rooms would be loopie-friendly, no rearrangement needed.

Additional examples

The following initial tests are also available:

- `0b` – $n = 3$, $m = 9$, $q = 1$, exactly one corridor between each pair of rooms in each direction. The query is $x_1 = 2, y_1 = 2$, the results are 3 and 3.
- `0c` – $n = 5\,000, m = n, q = 0, u_i = i, v_i = (i \bmod n) + 1$. The result is 5000.
- `0d` – $n = 500\,000, m = 2 \cdot (n-1), q = 0, u_i = u_{n-1+i} = 1, v_i = v_{n-1+i} = i + 1$ for each $1 \leq i \leq n - 1$. Result is $500\,000$.
- `0e` – $n = 1\,000\,000, m = n - 1, q = n - 3, a_i = \left\lfloor \frac{i+1}{2} \right\rfloor, b_i = i + 1$ for each $1 \leq i \leq n - 1$. Queries are $x_i = \left\lfloor \frac{i+3}{4} \right\rfloor, y_i = i + 3$ for each $1 \leq i \leq n - 3$. Results are, in order, $0, 3, 3, ..., 3$.

Limits

Your solution will be evaluated on a number of hidden test cases divided into groups. Points for a group are awarded if and only if the submission returns the correct answer for each of the tests in the group within the allotted time limit. These groups are organised into subtasks with the following limits and points awarded.

Subtask	Limits	Points
1.	$1 \leq n, m \leq 5000$, $q = 0$	2
2.	$1 \leq n, m \leq 1\,000\,000$, $q = 0$	2
3.	$1 \leq n, m, q \leq 1\,000\,000$	6