

# Kingdom

Shin wants to be the “Great General of the Heaven”. It is his dream. But becoming a “Great General of the Heaven” is no ordinary task. For becoming a “Great General”, you have to be good in both fighting and tactical aspects of war.

Shin just received a new mission from headquarter. Enemies are planning to attack the Kingdom of Qin and he is ordered to stop them in any means necessary. The Kingdom of Qin can be considered as a connected graph of  $N$  nodes, where the different cities are nodes numbered from 1 to  $N$  and roads between them are the edges.

The enemies have already infiltrated the Qin. Their current position is not known yet, but soon the spies inside the enemy will alert Shin of their position. Meanwhile, the enemy is planning to attack headquarter of Qin’s Military Force. The position of headquarter is also unknown to Shin, but soon authorities of Qin will tell Shin of its position.



So basically, even though Shin is not aware of the position of enemy or headquarter now, within few hours he will know both. Once he knows their position, he will march his army to stop the enemy. Let’s call the node in which the enemy is residing node  $E$  and headquarter node  $H$ .

Shin can only battle enemy in the middle of a road. Fighting inside the city needs to be avoided in order to avoid damage to citizens. But Shin needs to be careful when choosing road for the battle. There could be multiple paths between node  $E$  to  $H$ . If Shin waits in a road with his army and there is a path from  $E$  to  $H$  without using the road that Shin is guarding, then there is a possibility that the enemy will bypass Shin using that path and attack headquarter. So in order to avoid such situation, Shin has decided to guard a road such that the enemy will be unable to reach  $H$  from  $E$  without using that road. There could be multiple such roads and he calls them “Great Roads of Destiny”.

The real location of  $E$  and  $H$  have not arrived yet, so Shin is getting bored. In order to pass the time, he decided to train his tactical skills. He assumed a pair of integer as  $\{E,H\}$  and then tried to find out how many “Great Roads of Destiny” exists.

He now wants you to do the same. Given the graph of Kingdom of Qin, Shin will provide you with  $Q$  queries, where query will be a pair of integer representing  $E$  and  $H$ . Please tell Shin how many “Great Roads of Destiny” exists.

## Input

The first line will contain an integer  $T$  ( $T \leq 5$ ) indicating number of test cases.

For each test case, first line will contain a pair of integer  $N$  ( $2 \leq N \leq 100000$ ) and  $R$  ( $1 \leq R \leq 200000$ ), where  $N$  is number of nodes and  $R$  is number of roads in the Kingdom of Qin. After that there are  $R$  lines each containing a pair of integer  $U, V$  ( $1 \leq U, V \leq N$ ) representing there

is a road between node U and V. After that there is a single integer Q ( $1 \leq Q \leq 100000$ ), which is number of queries Shin will ask you. Next Q lines will contain a pair of integer representing E and H, as explained in the problem.

You can safely assume that the graph will be connected. There will be no selfloops or multiple edges between nodes.

There could be blank lines, or extra spaces in input.

### Output

For each case print a line "Case X:", where X is the number of case. After that for each query print a line with an integer Y, where Y is the number of "Great Roads of Destiny". See the sample input/output for better understanding.

### Sample Input

3

3 2  
1 2  
2 3  
2  
1 2  
1 3

4 4  
1 2  
1 3  
2 4  
3 4  
1  
1 4

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

1 4

## Sample Output:

Case 1:

1

2

Case 2:

0

Case 3:

1

## Explanation

In the first test case, Query 1 is 1 since the only road Shin should be guarding is the road between node 1 and 2. Query 2 is 2 since he can guard either the road 1->2 or 2->3. In the second case, Query 1 is 0 since there is no "Great Road of Destiny". In the third case, for Query 1 the only road is 2->4.

**Problem-setter: Mohammad Samiul Islam**