Problem-2

Topics Involved : DFS.

After a revolution in Berland the new dictator faced an unexpected challenge: the country has to be somehow ruled. The dictator is a very efficient manager, yet he can't personally give orders to each and every citizen. That's why he decided to pick some set of leaders he would control. Those leaders will directly order the citizens. However, leadership efficiency turned out to vary from person to person (i.e. while person A makes an efficient leader, person B may not be that good at it). That's why the dictator asked world-famous berland scientists for help. The scientists suggested an innovatory technology — to make the leaders work in pairs.

A relationship graph is some undirected graph whose vertices correspond to people. A simple path is a path with no repeated vertices. Long and frighteningly expensive research showed that a pair of people has maximum leadership qualities if a graph of relationships has a simple path between them with an odd number of edges. The scientists decided to call such pairs of different people leader pairs. Secret services provided the scientists with the relationship graph so that the task is simple — we have to learn to tell the dictator whether the given pairs are leader pairs or not. Help the scientists cope with the task.

Input

The first line contains integers n and m ($1 \le n \le 10^5$, $0 \le m \le 10^5$) — the number of vertices and edges in the relationship graph correspondingly. Next m lines contain pairs of integers a and b which mean that there is an edge between the a-th and the b-th vertices (the vertices are numbered starting from 1, $1 \le a, b \le n$). It is guaranteed that the graph has no loops or multiple edges.

Next line contains number q ($1 \le q \le 10^5$) — the number of pairs the scientists are interested in. Next q lines contain these pairs (in the same format as the edges, the queries can be repeated, a query can contain a pair of the identical vertices).

Output

For each query print on a single line "Yes" if there's a simple odd path between the pair of people; otherwise, print "No".

Examples

Input

7 7
1 3
1 4
2 3
2 4
5 6
6 7
7 5
8
1 2
1 3
1 4

2 4
1 5
5 6
5 7
6 7

Output

No
Yes
Yes
Yes
No
Yes
Yes
Yes

Note

Notes to the samples:

1) Between vertices 1 and 2 there are 2 different simple paths in total: 1-3-2 and 1-4-2. Both of them consist of an even number of edges.

2) Vertices 1 and 3 are connected by an edge, that's why a simple odd path for them is 1-3.

5) Vertices 1 and 5 are located in different connected components, there's no path between them.