

Edge Removals

2 sec

256000KB

100

Difficulty

Time Limit

Memory

Score

80/80 XP

30/30

Description

You have given an undirected graph G with N nodes, indexed from 1 to N and M edges, indexed from 1 to M .

There are two types of operations:

- 1 X: Remove the edge numbered X .
- 2: Print the number of connected components in the graph.

Input Format

The first line of contains three space-separated numbers N, M, Q ($1 \leq N, M, Q \leq 100000$).

The next M lines contain 2 space-separated integers u and v which depicts an edge between nodes u and v ($1 \leq u, v \leq N$). i th line denotes the i th edge.

This is followed by Q lines, each describing an operation in the following format:

- 1 X: Remove the edge numbered X .
- 2: Print the number of connected components in the graph.

There are no self-loops or multiple edges in the graph.

Note: If there are multiple queries for removal of the same edge, then the last such query should be considered. Also, the index of the edges does not change after the removal of any of the edges between the nodes.

Output Format

The output should consist of the answer to each of the operations of the 2nd type printed on a new line.

Constraints

```
89     for (auto e : edges,
90         {
91             pair<int, int> p = edge_ordering[e - 1];
92             add(p.first, p.second);
93         }
94     for (auto q : reverse_query)
95     {
96         if (q.first == 2)
97         {
98             ans.push_back(components);
99         }
100     else
101     {
102         pair<int, int> edge_to_add = edge_ordering[q.second - 1];
103         add(edge_to_add.first, edge_to_add.second);
104     }
105 }
```

Sample Tests

Manual Tests

Test Case 1

ACCEPTED

Input

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

Output