F. Cities Excursions

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are n cities in Berland. Some pairs of them are connected with m directed roads. One can use only these roads to move from one city to another. There are no roads that connect a city to itself. For each pair of cities (x, y) there is at most one road from x to y.

A path from city s to city t is a sequence of cities $p_1, p_2, \ldots, p_k$, where $p_1 = s$, $p_k = t$, and there is a road from city p_i to city p_{i+1} for each i from 1 to $k - 1$. The path can pass multiple times through each city except t. It can't pass through t more than once.

A path p from s to t is *ideal* if it is the lexicographically minimal such path. In other words, p is *ideal* path from s to t if for any other path q from s to t $p_i < q_i$, where i is the minimum integer such that $p_i \neq q_i$.

There is a tourist agency in the country that offers q unusual excursions: the j-th excursion starts at city s_j and ends in city t_j.

For each pair s_j, t_j help the agency to study the ideal path from s_j to t_j. Note that it is possible that there is no ideal path from s_j to t_j. This is possible due to two reasons:

- there is no path from s_j to t_j;
- there are paths from s_j to t_j, but for every such path p there is another path q from s_j to t_j, such that $p_i > q_i$, where i is the minimum integer for which $p_i \neq q_i$.

The agency would like to know for the ideal path from s_j to t_j the k_j-th city in that path (on the way from s_j to t_j).

For each triple s_j, t_j, k_j $(1 \leq j \leq q)$ find if there is an ideal path from s_j to t_j and print the k_j-th city in that path, if there is any.

Input

The first line contains three integers n, m and q $(2 \leq n \leq 3000, 0 \leq m \leq 3000,\ 1 \leq q \leq 4 \cdot 10^5)$ — the number of cities, the number of roads and the number of excursions.

Each of the next m lines contains two integers x_i and y_i $(1 \leq x_i, y_i \leq n,\ x_i \neq y_i)$, denoting that the i-th road goes from city x_i to city y_i. All roads are one-directional. There can't be more than one road in each direction between two cities.

Each of the next q lines contains three integers s_j, t_j and k_j $(1 \leq s_j, t_j \leq n,\ s_j \neq t_j,\ 1 \leq k_j \leq 3000)$.

Output

In the j-th line print the city that is the k_j-th in the ideal path from s_j to t_j. If there is no ideal path from s_j to t_j, or the integer k_j is greater than the length of this path, print the string '`-1`' (without quotes) in the j-th line.

Example

input

```
7 7 5
1 2
2 3
1 3
3 4
4 5
5 3
4 6
1 4 2
2 6 1
```

```
1 7 3
1 3 2
1 3 5
```

output

```
2
-1
-1
2
-1
```