E. Shortest Path

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In Ancient Berland there were n cities and m two-way roads of equal length. The cities are numbered with integers from 1 to n inclusively. According to an ancient superstition, if a traveller visits three cities a_i, b_i, c_i in row, without visiting other cities between them, a great disaster awaits him. Overall there are k such city triplets. Each triplet is ordered, which means that, for example, you are allowed to visit the cities in the following order: a_i, c_i, b_i. Vasya wants to get from the city 1 to the city n and not fulfil the superstition. Find out which minimal number of roads he should take. Also you are required to find one of his possible path routes.

Input

The first line contains three integers n, m, k ($2 \le n \le 3000, 1 \le m \le 20000, 0 \le k \le 10^5$) which are the number of cities, the number of roads and the number of the forbidden triplets correspondingly.

Then follow m lines each containing two integers x_i, y_i ($1 \le x_i, y_i \le n$) which are the road descriptions. The road is described by the numbers of the cities it joins. No road joins a city with itself, there cannot be more than one road between a pair of cities.

Then follow k lines each containing three integers a_i, b_i, c_i ($1 \le a_i, b_i, c_i \le n$) which are the forbidden triplets. Each ordered triplet is listed mo more than one time. All three cities in each triplet are distinct.

City n can be unreachable from city 1 by roads.

Output

If there are no path from 1 to n print -1. Otherwise on the first line print the number of roads d along the shortest path from the city 1 to the city n. On the second line print $d + 1$ numbers — any of the possible shortest paths for Vasya. The path should start in the city 1 and end in the city n.

Examples

input
4 4 1 1 2 2 3 3 4 1 3 1 4 3
output
2 1 3 4

input
3 1 0 1 2
output
-1

input
4 4 2 1 2 2 3 3 4

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

output

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