

## D. Aztec Catacombs

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

Indiana Jones found ancient Aztec catacombs containing a golden idol. The catacombs consists of  $n$  caves. Each pair of caves is connected with a two-way corridor that can be opened or closed. The entrance to the catacombs is in the cave 1, the idol and the exit are in the cave  $n$ .

When Indiana goes from a cave  $x$  to a cave  $y$  using an open corridor, all corridors connected to the cave  $x$  change their state: all open corridors become closed, all closed corridors become open. Indiana wants to go from cave 1 to cave  $n$  going through as small number of corridors as possible. Help him find the optimal path, or determine that it is impossible to get out of catacombs.

### Input

The first line contains two integers  $n$  and  $m$  ( $2 \leq n \leq 3 \cdot 10^5$ ,  $0 \leq m \leq 3 \cdot 10^5$ ) — the number of caves and the number of open corridors at the initial moment.

The next  $m$  lines describe the open corridors. The  $i$ -th of these lines contains two integers  $u_i$  and  $v_i$  ( $1 \leq u_i, v_i \leq n$ ,  $u_i \neq v_i$ ) — the caves connected by the  $i$ -th open corridor. It is guaranteed that each unordered pair of caves is presented at most once.

### Output

If there is a path to exit, in the first line print a single integer  $k$  — the minimum number of corridors Indians should pass through ( $1 \leq k \leq 10^6$ ). In the second line print  $k + 1$  integers  $x_0, \dots, x_k$  — the number of caves in the order Indiana should visit them. The sequence  $x_0, \dots, x_k$  should satisfy the following:

- $x_0 = 1$ ,  $x_k = n$ ;
- for each  $i$  from 1 to  $k$  the corridor from  $x_{i-1}$  to  $x_i$  should be open at the moment Indiana walks along this corridor.

If there is no path, print a single integer  $-1$ .

We can show that if there is a path, there is a path consisting of no more than  $10^6$  corridors.

### Examples

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

input
4 2 1 2 2 3
output
-1

4

1 2 3 1 4