

# Road Trip

Five friends set off on a road trip between Athens and Thessaloniki. Greece's road network consists of  $N$  cities, numbered from 1 to  $N$ , and  $M$  two-way roads, each connecting a pair of these cities. Athens is city number 1 and Thessaloniki is city number  $N$  and, of course, it is possible to travel between these two cities, via the road network.

All five friends love to drive, so they decide to take turns in driving. To avoid quarrelling, they devise a not so simple yet fair plan that will take them to their destination in the most enjoyable (and certainly not the fastest!) way. The plan is the following:

- Every day they start travelling in the morning. Each one of the five friends drives the car along one road of the network, between two directly connected cities. That is, every day they will follow a path consisting of exactly five roads.
- They agree not to visit the same city twice during the same day (but they can visit the same city any number of times on different days). That is, every day they will be at exactly six different cities.

You are asked to write a program to help the five friends determine the least possible number of days that their road trip will take, adhering to this plan, and the list of the cities where they will be in the morning of each day.

## Input

Your program must read from the standard input.

The first line will contain two space-separated integers  $N$  and  $M$ : the number of cities and the number of roads.

Each of the following  $M$  lines will contain two space-separated integers  $u$  and  $v$ , denoting that there exists a road directly connecting city  $u$  and city  $v$ . It will always be  $u \neq v$ .

## Output

If it is not possible for the five friends to complete their road trip according to their plan, your program must print just one line to the standard output, containing the number  $-1$ .

Otherwise, your program must print two lines to the standard output.

The first line will contain a single integer  $D$ : the least possible number of days in which the five friends will complete their road trip.

The second line will contain a sequence of  $D + 1$  space-separated integers: the numbers of the cities where the friends will be in the morning of each day. The first number of this sequence will always be 1 and the last number will always be  $N$ . If there are more than one different possible such sequences, your program should print the **lexicographically smallest** one.

## Constraints

- $2 \leq N \leq 100000$
- $1 \leq M \leq 500000$
- $1 \leq u \leq N$  and  $1 \leq v \leq N$
- Time and memory limit: See the CMS.

## Subtasks

- Subtask 1 (7 points):  $2 \leq N \leq 20$
- Subtask 2 (8 points):  $2 \leq N \leq 60$
- Subtask 3 (9 points):  $2 \leq N \leq 200$  and  $1 \leq M \leq 2500$
- Subtask 4 (13 points):  $2 \leq N \leq 200$
- Subtask 5 (16 points):  $2 \leq N \leq 2000$  and  $1 \leq M \leq 5000$
- Subtask 6 (21 points):  $2 \leq N \leq 2000$
- Subtask 7 (26 points): No further constraints.

## Example 1

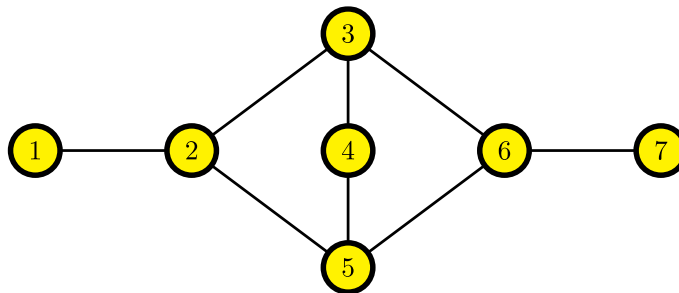
Input

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

Output

2  
1 4 7

## Explanation



One way to complete the road trip in two days is by travelling through cities  $1 \rightarrow 2 \rightarrow 3 \rightarrow 6 \rightarrow 5 \rightarrow 4$  on the first day, and then through cities  $4 \rightarrow 3 \rightarrow 2 \rightarrow 5 \rightarrow 6 \rightarrow 7$ . Another way to achieve the same is by travelling through cities  $1 \rightarrow 2 \rightarrow 5 \rightarrow 6 \rightarrow 3 \rightarrow 4$  on the first day, and then through cities  $4 \rightarrow 5 \rightarrow 2 \rightarrow 3 \rightarrow 6 \rightarrow 7$ . In both cases, the five friends are in city 4 in the morning of day 2.

Notice that, according to the friends' plan, it is **not legal** that they travel through cities  $1 \rightarrow 2 \rightarrow 5 \rightarrow 6 \rightarrow 3 \rightarrow 2$  on the first day, and then through cities  $2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7$  on the second day, because on the first day they would visit city 2 twice. If this had been allowed, the friends would have been in city 2 in the morning of day 2 and the resulting sequence 1 2 7 would have been preferred, as it would have been lexicographically smaller than 1 4 7.

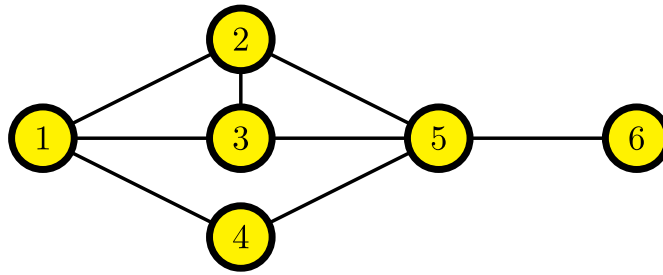
## Example 2

### Input

```
6 8
1 2
1 3
1 4
2 3
2 5
3 5
4 5
5 6
```

### Output

## Explanation



It is not possible to travel from city 1 to city 6 according to the friends' plan.