

## Problem H. Removing Vertices

Input file:            removing.in  
Output file:           removing.out  
Time limit:            2 seconds  
Memory limit:         256 mebybytes

You are given an bidirectional graph with  $(n + 1)$  vertices numbered from 0 to  $n$ . There are no loops and no multiple edges in this graph. Additionally, all cycles pass through vertex 0. Your task is to remove the minimum possible number of vertices so that there are no cycles in the graph. You can not remove the vertex number 0.

### Input

On the first line, there are two integers  $n$  and  $m$ : the number of vertices excluding vertex 0 and the number of edges ( $1 \leq n \leq 10^5$ ,  $1 \leq m \leq 10^6$ ).

Each of next  $m$  lines contains two integers  $a$  and  $b$ : the numbers of vertices connected by an edge ( $0 \leq a, b \leq n$ ).

It is guaranteed that all cycles pass through vertex 0.

### Output

On the first line, print one integer  $k$ : the minimum possible number of vertices to remove. On the second line, print  $k$  integers: the numbers of the vertices to remove in any order. If there are several optimal answers, print any one of them.

### Examples

| removing.in                                   | removing.out |
|---|--------------|
| 4 6<br>0 1<br>1 2<br>2 0<br>0 3<br>3 4<br>4 0 | 2<br>1 3     |
| 3 5<br>0 1<br>1 2<br>2 0<br>2 3<br>3 0        | 1<br>2       |