Task: DRO
Vari-directional Streets

Olimpiada Informatyczna

XXIII, Stage II, Day one. Source file `dro.*` Available memory: 512 MB. *10.02.2016*

Byteasar ponders moving to Bytown and renting an apartment there. Bytown is a beautiful city with many advantages, but it is a driver's nightmare. There are n intersections in the city, interconnected by a more or less erratic network of m streets. The streets ar extremely narrow, necessitating unidirectional traffic. Surprisingly, the urban planners recently came up with a work-around that allows moving in both directions along any street without widening it. Namely, they realized that traffic need not flow both ways concurrently. And so, on odd days the traffic flows as it did since time immemorial, whereas on even days all the street directions are reversed.

Byteasar wants to rent an apartment in a well-connected location. Namely, he is interested in an apartment at such an intersection that every other intersection can be reached from it *in one day* – this may be an odd day for some destinations and even for others. The way back can be ignored, as in the worst case Byteasar can backtrack on the very next day.

Given the Bytown's road network, determine all the intersections that satisfy Byteasar's requirements.

Input

In the first line of the standard input, there are two integers n and m ($n \geq 2$, $m \geq 1$), separated by a single space, specifying the number of intersections and of streets in Bytown respectively. The intersections are numbered from 1 to n. The m lines that follow describe the streets: The i-th such line contains two integers a_i and b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$), separated by a single space, which indicate that there is a one-way street originally oriented from the intersection no. a_i to the intersection no. b_i (i.e., on odd days the street can be traversed from a_i to b_i, whereas on even days it can be traversed from b_i to a_i). Every ordered pair (a_i, b_i) will appear on input at most once.

Output

In the first line of the standard input a single integer k, equal to the number of intersections satisfying Byteasar's requirements, should be printed. In the second line, an increasing sequence (of length k) of those intersections' numbers, separated by single spaces, should be printed. If $k = 0$, the second line should be empty; your program may either print an empty line or not print it at all.

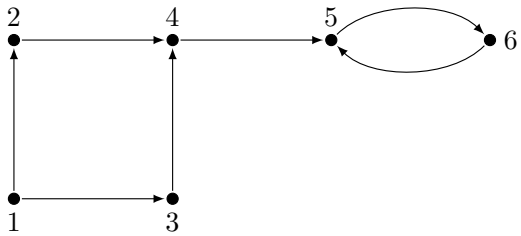
Example

For the following input data:

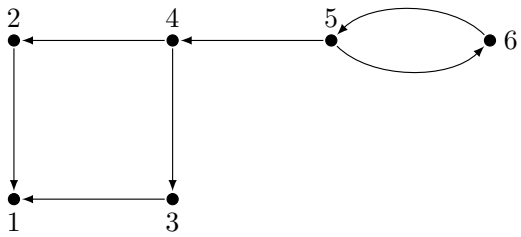
```
6 7
1 2
1 3
2 4
3 4
4 5
5 6
6 5
```

the correct result is:

```
4
1 4 5 6
```



The road network on odd days.



The road network on even days.

Explanation of the example: From the intersection no. 1, all other intersection can be reached on odd days. From each of the intersection no. 5 and 6, all other intersections can be reached on even days. From the

intersection no. 4, the intersections no. 5 and 6 can be reached on odd days, whereas the intersections no. 1, 2, and 3 on even days.

Sample Grading Tests:

`1ocen:` $n = 10$, $m = 9$, an "alternating path" whose edges are directed left or right depending on their parity. No intersection satisfies Byteasar's requirements.

`2ocen:` $n = 100\,000$, $m = 100\,000$, on odd days, every intersection can be reached directly from intersection no. 1; moreover, on odd days, the intersection no. 1 can be reached directly from the intersection no. n. The intersections no. 1 and n are the only ones that satisfy Byteasar's requirements.

`3ocen:` $n = 500\,000$, $m = 499\,999$, a "path"; all intersections satisfy Byteasar's requirements.

Grading

The set of tests consists of the following subsets. Within each subset, there may be several test groups.

Subset	Property	Score
1	$n, m \leq 5000$	28
2	$n \leq 300\,000$, $m \leq 1\,000\,000$; from each intersection satisfying Byteasa'r requirements, all other intersections can be reached on odd days	29
3	$n \leq 500\,000$, $m \leq 1\,000\,000$	43