Korn

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

Consider a connected graph $G = (V, E)$ without loops and parallel edges. Let's define a *complete walk* starting in vertex v as a sequence of visited vertices $v = p_0, p_1, p_2, \ldots, p_k = u$ such that the following conditions are satisfied:

1. For each $i = 1, 2, \ldots, k$, the unordered pair $(p_i, p_{i-1}) \in E$, that is, each two consecutive vertices are connected by an edge.
2. Each edge (a, b) appears at most once among all (p_i, p_{i-1}), that is, the walk p doesn't pass through the same edge twice.
3. There exists no vertex p_{k+1} that can be appended at the end of the walk such that the previous two conditions are still satisfied.

The vertex u is called the *terminal* vertex.

The vertex v is called *unavoidable* if any complete walk starting in v visits all edges in the graph (that is, $k = |E|$), and its terminal vertex is also v.

Your task is to find all *unavoidable* vertices in a given graph.

Input

The first line of input contains two integers n and m ($3 \le n \le 2 \cdot 10^5$, $n - 1 \le m \le 5 \cdot 10^5$), the number of vertices and the number of edges in the graph respectively.

The following m lines contain pairs of integers a_i, b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$), denoting endpoints of i-th edge.

It is guaranteed that the graph contains no loops and no parallel edges, and also that it is connected.

Output

Print the number of *unavoidable* vertices on the first line of output, and 1-based indices of all *unavoidable* vertices on the second line in ascending order.

Example

standard input	standard output
6 8 3 5 5 1 3 4 4 1 6 3 1 6 2 3 1 2	2 1 3

Note

In the sample, for example, vertex 4 is not unavoidable because there exists a complete walk 4, 5, 2, 1, 4 that terminates in 4 but that doesn't visit all edges in the graph.