



Task 5: Network

Rui Yuan the Duck is building a new network to run all of his new applications. This network is comprised of n servers numbered from 1 to n . There are also $n - 1$ network links, numbered from 1 to $n - 1$ connecting these servers.

The i -th of these links connects servers $u[i]$ and $v[i]$ **bidirectionally**. It is guaranteed that it is possible to travel from any server to any other server using only the servers and network links.

Initially, all n servers are active. Rui Yuan has m applications running, which have distinct IDs from 1 to m . Each application has 2 databases. Application j has databases in servers $a[j]$ and $b[j]$ (which may be the same server). It is possible for two different applications to have a database on the same server.

In order for application j to work, both of its databases must be connected through network links and **active** servers. To be precise, application j is **working** if **both** of the following conditions are met:

- Servers $a[j]$ and $b[j]$ are both active.
- One can travel from server $a[j]$ to server $b[j]$ using only network links and active servers.

Rui Yuan has asked Benson the Rabbit to test his network. Using his hacking skills, Benson can select any set of servers and deactivate all of them simultaneously. The robustness of the network is the minimum number of servers that must be deactivated to ensure **all** m applications are not working.

Benson is very busy, so he wants you to help him compute the robustness of the network and the selection of servers he should deactivate so that he doesn't waste time deactivating more servers than needed.

Input format

Your program must read from standard input.

The first line of input contains two integers, n and m .

$n - 1$ lines will follow. The i -th line contains two integers, $u[i]$ and $v[i]$ respectively, which are the servers connected by the i -th network link.

Another m lines will follow. The j -th line contains two integers, $a[j]$ and $b[j]$ respectively, which are the servers storing the databases for application j .



Output format

Your program must print to standard output.

Output two lines. The first line should contain a single integer, representing the robustness of the network, which we will denote as x .

The second line should contain x integers, representing the servers that should be deactivated. All x integers must be pairwise distinct. You may output them in any order.

Subtasks

For all testcases, the input will satisfy the following bounds:

- $1 \leq n \leq 200\,000$
- $1 \leq m \leq 200\,000$
- $1 \leq u[i], v[i] \leq n, u[i] \neq v[i]$ (for all $1 \leq i \leq n$).
- $1 \leq a[j], b[j] \leq n$ (for all $1 \leq j \leq m$)
- It is possible to travel from any server to any other server using only the servers and network links

Your program will be tested on input instances that satisfy the following restrictions:

Subtask	Marks	Additional Constraints
1	4	$a[i] = b[i]$
2	17	$u[i] = i, v[i] = i + 1$
3	14	$n, m \leq 15$
4	29	$n, m \leq 2000$
5	36	No additional restrictions

Sample Testcase 1

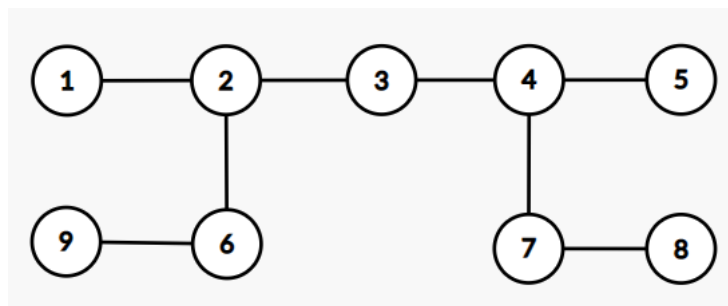
This testcase is valid for subtasks 3, 4 and 5.



Input	Output
9 4 1 2 2 3 2 6 3 4 4 5 4 7 6 9 7 8 6 2 5 3 4 8 5 9	2 4 2

Sample Testcase 1 Explanation

The network described in the input may be represented by the diagram below.



Suppose servers 4 and 2 are deactivated, then all applications will not be working as shown below.

- $b[1] = 2$, so application 1 is not working.
- A path from server $a[2] = 5$ to server $b[2] = 3$ must pass through server 4, so application 2 is not working.
- $a[3] = 4$, so application 3 is not working.
- A path from server $a[4] = 5$ to server $b[4] = 9$ must pass through both servers 2 and 4, so application 4 is not working.



It can be shown that it is not possible to ensure all applications are not working by deactivating a single server. Hence, the robustness of the network is 2.

Note that you may output the chosen servers in any order, so the following output is also valid.

```
2
2 4
```

Also note that there may exist other selections of 2 servers that cause all applications to not be working.

Sample Testcase 2

This testcase is valid for subtasks 1, 3, 4 and 5.

Input	Output
6 5 1 2 2 3 4 1 3 5 4 6 1 1 2 2 3 3 2 2 4 4	4 3 2 4 1

Sample Testcase 2 Explanation

Note that you may output the chosen servers in any order. For example, the following output is also valid.

```
4
4 3 2 1
```



Sample Testcase 3

This testcase is valid for subtasks 2, 3, 4 and 5.

Input	Output
8 3 1 2 2 3 3 4 4 5 5 6 6 7 7 8 3 5 8 5 4 4	2 4 6

Sample Testcase 4

This testcase is valid for subtasks 3, 4 and 5.

Input	Output
6 2 1 2 1 3 1 4 1 5 1 6 2 2 5 6	2 2 1