Work Stealing

Time Limit: 3.0s **Memory Limit:** 256M

Work stealing is a scheduling strategy for multithreaded computer programs. It solves the problem of executing a dynamically multithreaded computation. In a work stealing scheduler, each processor has a queue of work items to perform. Each work item consists of a series of instructions to be executed sequentially and those instructions are prerequisite of each other.

Let's say you want to implement your own scheduler. You are given n prerequisite pairs i, j which means that you must execute instruction j before instruction i. You are asked to find a way to execute every instruction in the prerequisite list.

Input

The first line contains one integer, n — the number of prerequisites.

Each of the next n lines contain two integers i and j, meaning that instruction j must be executed before instruction i.

- $1 \leq i, j \leq 1000$
- $i \neq j$

All the pairs i, j are distinct.

There is no cyclic dependency between instructions.

Output

First, print the number of instructions to execute, m.

Afterwards, print m integers representing the indices of the instructions to execute from first to last.

Example

Input 1:

```
1
2 7
```

Output 1:

```
2
7 2
```

Input 2:

```
4
2 1
3 1
4 2
4 3
```

Output 2:

```
4
1 2 3 4
```

Explanation

Input 1: There are 2 instructions to execute, and to execute instruction 2 scheduler should execute instruction 7. So the correct course order is `7 2`.

Input 2: There are 4 instructions to execute, and to execute instruction 4, scheduler should execute instruction 2 and instruction 3. Both instruction 2 and instruction 3 should also be executed after instruction 1. So, one correct order is `1 2 3 4`. `1 3 2 4` is also acceptable.