IOI '95 P5 - Street Race

Time limit: 1.0s **Memory limit:** 8M

IOI '95 - Eindhoven, Netherlands

Figure 1 gives an example of a course for a street race. You see some points, labeled from 0 to N (here $N = 9$), and some arrows connecting them. Point 0 is the start of the race; point N is the finish. The arrows represent one-way streets. The participants of the race move from point to point via the streets, in the direction of the arrows only. At each point, a participant may choose any outgoing arrow.

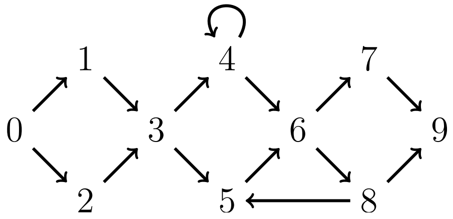

Figure 1: A street course with 10 points

A well-formed course has the following properties:

1. Every point in the course can be reached from the start.
2. The finish can be reached from each point in the course.
3. The finish has no outgoing arrows.

A participant does not have to visit every point of the course to reach the finish. Some points, however, are unavoidable. In the example, these are points 0, 3, 6, and 9. Given a well-formed course, your program has to determine the set of unavoidable points that all participants have to visit, excluding start and finish.

Suppose the race has to be held on two consecutive days. For that purpose the course has to be split into two courses, one for each day. On the first day, the start is at point 0, and the finish at some 'splitting point'. On the second day, the start is at this splitting point and the finish is at point N. Given a well-formed course, your program has to determine the set of splitting points.

A point S is a splitting point for the well-formed course C if S differs from the start and the finish of C, and the course can be split into two well-formed courses that have no common arrows and that have S as only common point. In the example, only point 3 is a splitting point.

Input Specification

The input describes a well-formed course with at most 50 points and at most 100 arrows. There are $N + 1$ lines in the input. The first N lines contain the endpoints of the arrows that leave from the points 0 through $N - 1$ respectively. Each of these lines ends with the number -2. The last line contains the number -1.

Output Specification

Your program should write two lines. The first line should contain the number of unavoidable points in the input course, followed by the labels of these points, in ascending order. The second line should contain the number of splitting points of the input course, followed by the labels of all these points, in ascending order.

Sample Input

```
1 2 -2
3 -2
3 -2
5 4 -2
6 4 -2
6 -2
7 8 -2
9 -2
5 9 -2
-1
```

Sample Output

```
2 3 6
1 3
```