**TaskB???: RACE**

**Author: Pano Panov**

Figure 1 gives an example of a course for a street race. You see N+1 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 paths, in the direction of the arrows only. At each point, a participant may choose any outgoing arrow.

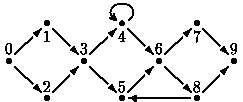


Figure 1: A path course with 10 points

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.

**Input**

There are *N*+1 lines in the system 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**

The only line to the system output should contain the number of unavoidable points in the input course, followed by the labels of these points, in ascending order.

**Constrains**

Number of Points ≤50

Number of Arrows ≤100

**Example**

**Input Output**

1 2 -2 2 3 6

3 -2

3 -2

5 4 -2

6 4 -2

6 -2

7 8 -2

9 -2

5 9 -2

-1