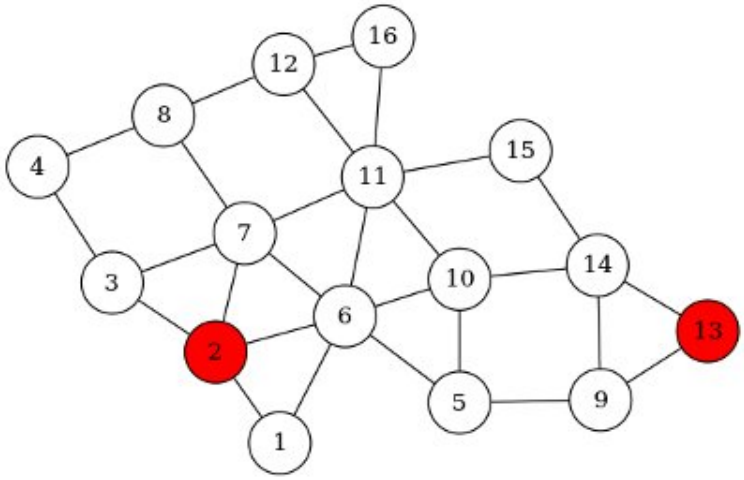
Problem C – Crimes in Whitechapel†



The notorious assassin "Jack the Ripper" terrorized the London district of Whitechapel in 1888. Despite extensive police investigations, Jack was never identified and caught. In this problem you get to help the police using computational methods that were not available in the 19th century.

Each position in the Whitechapel district is identified by an integer number from 1 to **N**, the number of different locations (see the figure above). In each night, Jack committed a crime in a different location (numbers 2 and 13 in the figure) and then returned to his hideout by a **shortest path**. The police has obtained some estimates about the distance he walked (but not exactly which streets he used): 3 streets from location 2 and 4 streets from location 13. However, this information is sufficient to deduce that Jack's hideout must be in either location 12 or 16.

The Task

Write a program which reads a description of the district map and distance clues and outputs a list of possible locations of Jack's hideout.

Input

The first line of input contains **N**, the number of district locations.

The following **N** lines describe the connections: the i-th of these lines consists of a list of locations connected to location i, separated by a space. Each list is terminated by a single zero (0).

†A few minor changes have been made to the original version, which appeared in the 5th Contest of TIUP 2012 (3rd of October), organized by Departamento de Ciência de Computadores da Faculdade de Ciências da Universidade do Porto.

The next line contains **M**, the number of distance clues.

The following **M** lines contain the clues, one per line. Each clue is a pair of numbers separated by a space specifying the crime location **c** and the estimated distance **d** to the hideout, respectively.

Constraints

$1 \leq \mathbf{M} \leq 20$	Number of clues
$\mathbf{M} < \mathbf{N} \leq 10\,000$	Number of locations
$1 \leq \mathbf{c} \leq \mathbf{N}$	Crime location
$1 \leq \mathbf{d} < \mathbf{N}$	Distance to the hideout

Output

The output should be the list of possible positions for Jack's hideout in ascending order and separated by a single space. If there is no location satisfying all given clues (that is, when the list is empty), the output should instead be "NO SOLUTION".

Example Input 1

```
16
6 2 0
1 7 3 6 0
2 4 7 0
3 8 0
10 6 9 0
5 2 1 11 7 10 0
6 3 2 8 11 0
7 4 12 0
5 14 13 0
6 5 11 14 0
10 7 6 16 12 15 0
11 8 16 0
9 14 0
13 10 9 15 0
14 11 0
12 11 0
2
2 3
13 4
```

Example Output 1

```
12 16
```

Example Input 2

```
16
6 2 0
1 7 3 6 0
2 4 7 0
3 8 0
10 6 9 0
5 2 1 11 7 10 0
6 3 2 8 11 0
7 4 12 0
5 14 13 0
6 5 11 14 0
10 7 6 16 12 15 0
11 8 16 0
9 14 0
13 10 9 15 0
14 11 0
12 11 0
2
2 1
13 2
```

Example Output 2

```
NO SOLUTION
```