A. Spit Problem

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In a Berland's zoo there is an enclosure with camels. It is known that camels like to spit. Bob watched these interesting animals for the whole day and registered in his notepad where each animal spitted. Now he wants to know if in the zoo there are two camels, which spitted at each other. Help him to solve this task.

The trajectory of a camel's spit is an arc, i.e. if the camel in position x spits d meters right, he can hit only the camel in position $x + d$, if such a camel exists.

Input

The first line contains integer n $(1 \leq n \leq 100)$ — the amount of camels in the zoo. Each of the following n lines contains two integers x_i and d_i $(-10^4 \leq x_i \leq 10^4, 1 \leq |d_i| \leq 2 \cdot 10^4)$ — records in Bob's notepad. x_i is a position of the i-th camel, and d_i is a distance at which the i-th camel spitted. Positive values of d_i correspond to the spits right, negative values correspond to the spits left. No two camels may stand in the same position.

Output

If there are two camels, which spitted at each other, output `YES`. Otherwise, output `NO`.

Sample test(s)

input

```
2
0 1
1 -1
```

output

```
YES
```

input

```
3
0 1
1 1
2 -2
```

output

```
NO
```

input

```
5
2 -10
3 10
0 5
5 -5
10 1
```

output

```
YES
```

B. Traffic Lights

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A car moves from point A to point B at speed v meters per second. The action takes place on the X-axis. At the distance d meters from A there are traffic lights. Starting from time 0, for the first g seconds the green light is on, then for the following r seconds the red light is on, then again the green light is on for the g seconds, and so on.

The car can be instantly accelerated from 0 to v and vice versa, can instantly slow down from the v to 0. Consider that it passes the traffic lights at the green light instantly. If the car approaches the traffic lights at the moment when the red light has just turned on, it doesn't have time to pass it. But if it approaches the traffic lights at the moment when the green light has just turned on, it can move. The car leaves point A at the time 0.

What is the minimum time for the car to get from point A to point B without breaking the traffic rules?

Input

The first line contains integers l, d, v, g, r ($1 \le l, d, v, g, r \le 1000$, $d < l$) — the distance between A and B (in meters), the distance from A to the traffic lights, car's speed, the duration of green light and the duration of red light.

Output

Output a single number — the minimum time that the car needs to get from point A to point B. Your output must have relative or absolute error less than 10^{-6}.

Sample test(s)

input

```
2 1 3 4 5
```

output

```
0.66666667
```

input

```
5 4 3 1 1
```

output

```
2.33333333
```

C. Mail Stamps

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

One day Bob got a letter in an envelope. Bob knows that when Berland's post officers send a letter directly from city «A» to city «B», they stamp it with «A B», or «B A». Unfortunately, often it is impossible to send a letter directly from the city of the sender to the city of the receiver, that's why the letter is sent via some intermediate cities. Post officers never send a letter in such a way that the route of this letter contains some city more than once. Bob is sure that the post officers stamp the letters accurately.

There are n stamps on the envelope of Bob's letter. He understands that the possible routes of this letter are only two. But the stamps are numerous, and Bob can't determine himself none of these routes. That's why he asks you to help him. Find one of the possible routes of the letter.

Input

The first line contains integer n $(1 \le n \le 10^5)$ — amount of mail stamps on the envelope. Then there follow n lines with two integers each — description of the stamps. Each stamp is described with indexes of the cities between which a letter is sent. The indexes of cities are integers from 1 to 10^9. Indexes of all the cities are different. Every time the letter is sent from one city to another, exactly one stamp is put on the envelope. It is guaranteed that the given stamps correspond to some valid route from some city to some **other** city.

Output

Output $n+1$ numbers — indexes of cities in one of the two possible routes of the letter.

Sample test(s)

input

```
2
1 100
100 2
```

output

```
2 100 1
```

input

```
3
3 1
100 2
3 2
```

output

```
100 2 3 1
```

D. Ant on the Tree

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Connected undirected graph without cycles is called a tree. Trees is a class of graphs which is interesting not only for people, but for ants too.

An ant stands at the root of some tree. He sees that there are n vertexes in the tree, and they are connected by $n - 1$ edges so that there is a path between any pair of vertexes. A leaf is a distinct from root vertex, which is connected with exactly one other vertex.

The ant wants to visit every vertex in the tree and return to the root, passing every edge twice. In addition, he wants to visit the leaves in a specific order. You are to find some possible route of the ant.

Input

The first line contains integer n $(3 \le n \le 300)$ — amount of vertexes in the tree. Next $n - 1$ lines describe edges. Each edge is described with two integers — indexes of vertexes which it connects. Each edge can be passed in any direction. Vertexes are numbered starting from 1. The root of the tree has number 1. The last line contains k integers, where k is amount of leaves in the tree. These numbers describe the order in which the leaves should be visited. It is guaranteed that each leaf appears in this order exactly once.

Output

If the required route doesn't exist, output `-1`. Otherwise, output $2n - 1$ numbers, describing the route. Every time the ant comes to a vertex, output it's index.

Sample test(s)

input

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

output

```
1 2 3 2 1
```

input

```
6
1 2
1 3
2 4
4 5
4 6
5 6 3
```

output

```
1 2 4 5 4 6 4 2 1 3 1
```

input

```
6
1 2
1 3
2 4
4 5
4 6
5 3 6
```

output

```
-1
```

E. Quarrel

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Friends Alex and Bob live in Bertown. In this town there are n crossroads, some of them are connected by bidirectional roads of equal length. Bob lives in a house at the crossroads number 1, Alex — in a house at the crossroads number n.

One day Alex and Bob had a big quarrel, and they refused to see each other. It occurred that today Bob needs to get from his house to the crossroads n and Alex needs to get from his house to the crossroads 1. And they don't want to meet at any of the crossroads, but they can meet in the middle of the street, when passing it in opposite directions. Alex and Bob asked you, as their mutual friend, to help them with this difficult task.

Find for Alex and Bob such routes with equal number of streets that the guys can follow these routes and never appear at the same crossroads at the same time. They are allowed to meet in the middle of the street when moving toward each other (see Sample 1). Among all possible routes, select such that the number of streets in it is the least possible. Until both guys reach their destinations, none of them can stay without moving.

The guys are moving simultaneously with equal speeds, i.e. it is possible that when one of them reaches some of the crossroads, the other one leaves it. For example, Alex can move from crossroad 1 to crossroad 2, while Bob moves from crossroad 2 to crossroad 3.

If the required routes don't exist, your program should output `-1`.

Input

The first line contains two integers n and m ($2 \le n \le 500, 1 \le m \le 10000$) — the amount of crossroads and the amount of roads. Each of the following m lines contains two integers — the numbers of crossroads connected by the road. It is guaranteed that no road connects a crossroads with itself and no two crossroads are connected by more than one road.

Output

If the required routes don't exist, output `-1`. Otherwise, the first line should contain integer k — the length of shortest routes (the length of the route is the amount of roads in it). The next line should contain $k+1$ integers — Bob's route, i.e. the numbers of $k+1$ crossroads passed by Bob. The last line should contain Alex's route in the same format. If there are several optimal solutions, output any of them.

Sample test(s)

input
```
2 1
1 2
```
output
```
1
1 2
2 1
```

input
```
7 5
1 2
2 7
7 6
2 3
3 4
```
output
```
-1
```

input
```
7 6
1 2
2 7
7 6
2 3
3 4
1 5
```
output
```
6
1 2 3 4 3 2 7
7 6 7 2 1 5 1
```