C. Circle of Numbers

One day Vasya came up to the blackboard and wrote out n distinct integers from 1 to n in some order in a circle. Then he drew arcs to join the pairs of integers (a, b) $(a \neq b)$, that are either each other's immediate neighbors in the circle, or there is number c, such that a and c are immediate neighbors, and b and c are immediate neighbors. As you can easily deduce, in the end Vasya drew $2 \cdot n$ arcs.

For example, if the numbers are written in the circle in the order $1, 2, 3, 4, 5$ (in the clockwise direction), then the arcs will join pairs of integers $(1, 2)$, $(2, 3)$, $(3, 4)$, $(4, 5)$, $(5, 1)$, $(1, 3)$, $(2, 4)$, $(3, 5)$, $(4, 1)$ and $(5, 2)$.

Much time has passed ever since, the numbers we wiped off the blackboard long ago, but recently Vasya has found a piece of paper with $2 \cdot n$ written pairs of integers that were joined with the arcs on the board. Vasya asks you to find the order of numbers in the circle by these pairs.

Input

The first line of the input contains a single integer n ($5 \leq n \leq 10^5$) that shows, how many numbers were written on the board. Next $2 \cdot n$ lines contain pairs of integers a_i, b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$) — the numbers that were connected by the arcs.

It is guaranteed that no pair of integers, connected by a arc, occurs in the input more than once. The pairs of numbers and the numbers in the pairs are given in the arbitrary order.

Output

If Vasya made a mistake somewhere and there isn't any way to place numbers from 1 to n on the circle according to the statement, then print a single number "-1" (without the quotes). Otherwise, print any suitable sequence of n distinct integers from 1 to n.

If there are multiple solutions, you are allowed to print any of them. Specifically, it doesn't matter which number you write first to describe the sequence of the order. It also doesn't matter whether you write out the numbers in the clockwise or counter-clockwise direction.

Examples

input
5 1 2 2 3 3 4 4 5 5 1 1 3 2 4 3 5 4 1 5 2

output
1 2 3 4 5

input
6 5 6 4 3 5 3

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

output

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