Problem D: Decorative Dominoes

Marie likes Dominoes. She is too young to fully understand the game, so she just creates arrangements based on the following simple rule: Each of the two ends of a domino must be adjacent to an end of another domino with the same number on it.

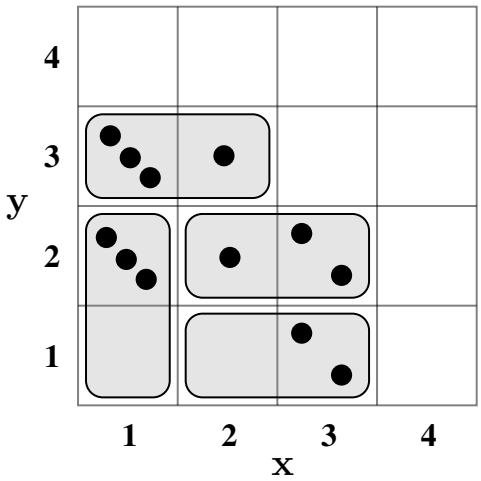


Figure D.1: Visualization of the first sample test case.

Today, Marie found a large box of blank dominoes. This is very exciting for her because now she can show her full creativity by first creating an unrestricted arrangement and then, in a second step, painting numbers on both ends of all dominoes so that her simple rule is fulfilled.

She already decided that putting the same number on each end of every domino is not satisfying enough for her. She only wants to use each number at most twice. However, she does not restrict herself to numbers between 0 and 6, and she also does not care if two dominoes have the same pair of numbers on them.

Marie positions the dominoes along an integer grid, so that each domino occupies exactly two neighbouring grid squares. Note that Marie's arrangement does not necessarily have to be connected.

After Marie decided on an arrangement, she notices that choosing suitable numbers is harder than initially anticipated. Help her to find a valid numbering for her given arrangement or state that this is impossible.

Input

The input consists of:

- One line with an integer n ($2 \leq n \leq 5\,000$), the number of dominoes in Marie's arrangement.
- n lines, each with four integers x_1, y_1, x_2, y_2 ($1 \leq x_1, y_1, x_2, y_2 \leq 10\,000$), where (x_1, y_1) and (x_2, y_2) are the grid positions of the two ends of one of the dominoes.

It is guaranteed that all dominoes occupy two neighbouring positions in the integer grid and no two dominoes overlap.

Output

If a valid numbering exists, print n lines, the ith of which contains two numbers, the integers that Marie should write on the two ends of the ith domino, respectively. Output the numbers in

the same order as the dominoes (including their two ends) appear in the input. All numbers in the output should be integers between 0 and 10^6 inclusive. In case multiple valid numberings exist, you may output any one of them. If there does not exist a valid numbering, output `impossible` instead.

Sample Input 1

```
4
1 1 1 2
2 2 3 2
2 1 3 1
1 3 2 3
```

Sample Output 1

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

Sample Input 2

```
4
1 1 2 1
1 2 2 2
4 2 4 3
4 4 3 4
```

Sample Output 2

```
impossible
```