/ Rooks

On a chessboard of size $n \times n$ we place n rooks. The arrangement of the rooks should satisfy the following rules:

→ For each $i = 1,...,n$, the i-th rook can be placed within a rectangle specified by two pairs of coordinates: (a_i, b_i), (c_i, d_i), where (a_i, b_i) are the coordinates of the square in the upper left corner of the rectangle (row, column), (c_i, d_i) are the coordinates of the square in the lower right corner of the rectangle, $1 \le a_i \le c_i \le n$ and $1 \le b_i \le d_i \le n$. The square in the upper left corner has coordinates $(1, 1)$; the square in the lower right corner has coordinates (n, n).

→ No two rooks can attack each other, i.e. they cannot appear in the same row or the same column.

Task

Write a program that:

→ reads from the input the size of the chessboard n and for each $i = 1,...,n$ the coordinates of the rectangle in which the i-th rook can be placed,

→ verifies whether the rooks may be placed in appropriate rectangles such that no two of them may attack each other and, if so, finds one such arrangement,

→ writes to the output one arrangement of rooks that satisfies the given conditions, or the single word NIE (Polish for *no*) if no such arrangement exists.

Input

In the first line of the input there is a positive integer n ($1 \le n \le 3000$). In each of the following n lines appear four positive integers not greater than n, separated by single spaces. The numbers in the i-th line are the coordinates of the rectangle in which the i-th rook may be placed (a_i, b_i, c_i and d_i respectively).

Output

The output should comprise either one word, NIE, or a series of *n* lines on each of which are written two integers separated by a single space. The numbers in the *i*-th line should specify the position of the *i*-th rook (row, column). Each such position should be within the rectangle specified by the coordinates in the $(i + 1)$-th line of the input. Pay attention to the fact that the positions of the rooks should be written in the same order as the coordinates of the rectangles are specified in the input.

Example

For the input data:

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

the correct result is:

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