

# Xiangqi

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            1 second  
Memory limit:         1024 megabytes

Chinese chess (Xiangqi) is a popular game enjoyed by people of all ages. This was even noted at the 49th ICPC World Finals.



*Photo taken in the 49th ICPC World Finals*

Today, having just learned the rules of Xiangqi, you challenge the grandmaster Nanani. The grandmaster does not want to overwhelm you, so you will play a simplified version of the game with the following rules:

- The chessboard has 9 columns and 10 rows. Only two pieces are placed: a Knight and a Rook. If a piece is at column  $x$  and row  $y$ , we denote its position by  $(x, y)$ . Both the Rook and the Knight must stay within the chessboard and may not move outside it.
- The two players take turns moving their pieces. If, after a player's move, their piece moves to the exact position of the opponent's piece, it is said that they have "captured" the opponent's piece, and the game ends.
- The moving rules for the Knight are as follows:
  - It can move from  $(x_N, y_N)$  to  $(x_N + 2, y_N \pm 1)$ , if  $(x_N + 1, y_N)$  is empty.
  - It can move from  $(x_N, y_N)$  to  $(x_N - 2, y_N \pm 1)$ , if  $(x_N - 1, y_N)$  is empty.
  - It can move from  $(x_N, y_N)$  to  $(x_N \pm 1, y_N + 2)$ , if  $(x_N, y_N + 1)$  is empty.
  - It can move from  $(x_N, y_N)$  to  $(x_N \pm 1, y_N - 2)$ , if  $(x_N, y_N - 1)$  is empty.
- The Rook can move to any other position in the same row or column, but cannot jump over the other piece.

- Each player must move their piece on their turn. It can be proven that the current player always has a way to move their piece in any situation.

The grandmaster controls the Knight, and you control the Rook, with the grandmaster moving first. Does the Rook have a strategy such that, no matter how the Knight responds, the Rook can capture the Knight in finitely many moves without being captured by the Knight?

### Input

There are multiple test cases. The first line of the input contains an integer  $T$  ( $1 \leq T \leq 8010$ ) indicating the number of test cases. For each test case:

The first and only line contains four integers  $x_N, y_N, x_R$  and  $y_R$  ( $1 \leq x_N, x_R \leq 9, 1 \leq y_N, y_R \leq 10$ ), where  $(x_N, y_N)$  is the initial position of the Knight, and  $(x_R, y_R)$  is the initial position of the Rook. It is guaranteed that the initial positions of the two pieces are different.

### Output

For each test case output one line. If such a strategy exists for the Rook, output YES, or otherwise, output NO.

### Example

standard input	standard output
2	YES
1 5 2 5	NO
1 1 2 3	

### Note

The sample test cases are shown as follows. Red pieces represent Knight, and black pieces represent Rook.

