

Introduction

Chess is a well-known and well-studied game. Thousands of computer programs have been written about chess. Some play the game interactively, while others analyze and predict strategies. One thing is certain: if you don't know how the chess pieces move, your program will not be any good.



For this problem, we are going to take a look at a chess piece called the bishop. The bishop can move any number of squares, but it has to move diagonally in a straight line. Also, what fun is a standard chess board? We are going to be using boards of random size. One good thing though: your bishop will be the only piece on the board.

Program Input

The first line of the file will contain a positive integer T denoting the number of test cases that follow. Each test case will have the following input:

- The first line of each test case will contain two numbers, R and C , separated by a single comma that tell us how many rows and columns our chess board has. The top left square will be space $(1,1)$, and the bottom right will be (R, C) . Both R and C will be greater than or equal to 2, and less than or equal to 1,000.
- The second line of each test case will contain two numbers, $r1$ and $c1$, separated by a single comma denoting the starting row and column for our bishop.
- The last line of each test case will contain two numbers, $r2$ and $c2$, separated by a single comma denoting the ending row and column for our bishop.

Example Input:

```
3
8, 8
1, 1
8, 8
10, 20
5, 6
7, 8
100, 100
50, 50
20, 21
```

Program Output

Since we are just interested in you being able to make your bishop move, your program should simply output "Yes" if the bishop can move from the starting space to the ending space in any number of moves, and "No" if it can't.

Example Output:

Yes

Yes

No