

Block Bin

time limit per test: 1 second (3 seconds for Python)
memory limit per test: 256 megabytes
input: standard input
output: standard output
points: 100 · problem code: blockbin · author: de

It's a beautiful day and you are out at the arcade, and, after examining each game available, you have decided that the most optimal one to play is one named Block Bin, for you believe you have a strategy that can always win (and it has a high reward yield!). It's a pretty simple pixel game where pixels drop down and get cleared after filling the bottom row. The game is detailed as follows:

The screen starts with a grid of 10^{18} rows and Z columns. A square in the grid will be referred to as (x, y) , where x specifying the column, starting at 1 from the left and incrementing, and y specifying the row, starting at 0 from the bottom and incrementing. (Read again carefully.)

There are C blocks in total, each occupying one square of the grid. You will be given C .

You will be told their starting square at the **beginning** of time 0, each in the form x_i, y_i .

Time in this game moves in discrete intervals, moving from time a to time $a + 1$ abides by two simple rules:

1. If the entire bottom row is filled with blocks, then all blocks in the bottom row are cleared.
2. For each remaining block, in order from bottom to top, perform the following:
 - If the block is in the bottom row, or if there is a block in the cell immediately below it, do nothing.
 - Otherwise, move the block one cell downward.

Remember, the rules are always executed in that order!

The game will then turn the grid display off and then display queries, asking whether the i -th block exists after time t but before $t + 1$.

You will be given q , the number of queries.

You must answer this query with a simple "No", if the block hasn't been cleared, or "Yes" if it has. If the block never gets cleared at any point in time, answer instead with "Never".

These will be in the form $c_i t_i$, where c_i refers to the i -th block (1-indexed) given to you, and t_i is the time being queried.

Input

- The first line contains two integers C and Z , the number of blocks and the number of columns.
- The next C lines will contain two integers, $x_i y_i$, specifying the position of the i -th block.
- The following line contains integer q , the number of queries.
- The next q lines will contain two integers, $t_i i$, specifying that the i -th block is being queried at time t_i .

Output

- Either "Yes", "No", or "Never", one for each q queries given.

Constraints

- $1 \leq i \leq C \leq 2 \times 10^5$
- $1 \leq Z \leq C$
- $1 \leq X_i \leq Z$
- $1 \leq Y_i \leq 10^{18}$
- $1 \leq q \leq 10^5$
- $1 \leq t_i \leq 10^{18}$
- Two blocks cannot occupy the same square at any time

Example 1

Input

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

Output

```
No
Never
Yes
Never
Yes
Never
```

Example 2

Input

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

Output

```
Yes
Yes
Never
Never
```