Task: FLA
Flappy Bird

Olimpiada Informatyczna

XXIV OI, Stage 1. Source file `fla.*` Available memory: 128 MB. *17.10–14.11.2016*

Ever since Byteasar installed the game *Flappy Bird* on his smartphone, he simply cannot stop playing it. He has become such an avid player that even the most difficult level is no challenge to him. Hence, to keep the game interesting, Byteasar has decided to complete each level with the least effort possible. Adopting this philosophy immediately, he has asked you to kindly write a program that will aid him in his effort.

To explain what he needs, Byteasar has formalized the game play as follows. At every moment, the bird avatar, controlled by the player, may reside in any integer point of a rectangular (i.e., 2D Cartesian) coordinate system. Initially, the bird is located in the origin $(0, 0)$, and the player wins if they move the bird to any point with the first coordinate X.

The bird moves every second: if it is currently located at the point (x, y), then in the next second it will move to one of two possible points. Namely, if the player taps the screen, the bird will move to the point $(x + 1, y + 1)$. If, however, the player does nothing, the bird will move to the point $(x + 1, y - 1)$.

Furthermore, there are n obstacles the bird has to avoid. Every obstacle consists of two *forbidden rays*. Should the bird ever touch a forbidden ray, the player loses immediately. The i-th obstacle is described by a triple of numbers (x_i, a_i, b_i), which indicate that the points (x_i, y) for $y \leq a_i$ and the points (x_i, y) for $y \geq b_i$ form the obstacle's forbidden rays, depicted below as thin rectangles.

Given a set of obstacles, Byteasar wants to know the minimum number of screen taps that suffice to win.

Input

The first line of the standard input contains two integers n and X ($0 \leq n \leq 500\,000$; $1 \leq X \leq 10^9$), which specify the number of obstacles and the finish line. The n lines that follow specify the obstacles: the i-th line contains three integers x_i, a_i, and b_i ($0 < x_i < X$; $a_i < b_i$; $a_i, b_i \in [-10^9, 10^9]$), describing the i-th obstacle in the aforementioned format. The obstacles are ordered from left to right, i.e., $x_1 < x_2 < \ldots < x_n$.

Output

If winning is impossible for the given set of obstacles, the word `NIE` (Polish for *no*) should be printed to the first and only line of the standard output. Otherwise, a single nonnegative integer equal to the minimum number of screen taps sufficient for winning should be printed.

Example

For the following input data:

```
4 11
4 1 4
7 -1 2
8 -1 3
9 0 2
```

the correct result is:

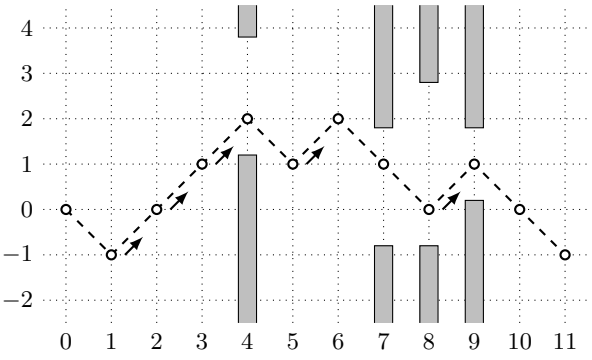
```
5
```

whereas for the following input data:

```
1 2
1 -1 1
```

the correct result is:

```
NIE
```



The figure depicts the first sample test. Arrows mark the positions in which the player should tap the phone screen.

Sample grading tests:

`1ocen`: $n = 5$, $X = 6$;

`2ocen`: $n = 0$, $X = 10^9$, answer: 0;

`3ocen`: $n = 500\,000 - 1$, $X = 10^9$, $x_i = 1000 \cdot i$, $a_i = -1$, $b_i = 2$, answer: 249 999 500.

Grading

The set of tests consists of the following subsets. Within each subset, there may be several test groups. Each subset conforms to the rules specified in the "Input" section.

Subset	**Property**	**Score**
1	$n \leq 1000$, $X \leq 1000$, $a_i, b_i \in [-1000, 1000]$	28
2	$n \leq 1000$, $a_i, b_i \in [-1000, 1000]$	23
3	$b_i = 10^9$	21
4	no additional properties	28