Task 1. Bombs

As usual, Lora is spending her time at work playing video games. Today we're going to look at a variation of the game *Bomberman*.

The game consists of a grid of size 10^9x10^9. Each cell can either be empty, contain a box, or contain a rock. There are N boxes and M rocks in the grid. The player has an unlimited amount of bombs of the following two types:

- A horizontal bomb that is placed **in an empty cell** and explodes, shooting fire to the left and to the right. The fire moves in a straight line and stops upon reaching a rock or a box. If the fire reaches a rock, the fire disappears. If it reaches a box, the fire still disappears, but the box is destroyed as well and the cell, in which it was, becomes empty.

- A vertical bomb that works in the exact same way, but shoots the fire upwards and downwards instead.

The player detonates the bombs one by one, i.e. a given bomb explodes before the next one is placed. **It is guaranteed that a cell containing a box is not on the edge of the grid and the 4 cells that border it are all empty.** The goal of the game is to destroy all boxes with a minimum possible number of bombs. Help Lora by writing a program **bombs** that given a playing field finds the minimal number of bombs needed to destroy all boxes, as well as the positions they should be placed in.

Input.

The first line of the standard input contains the number of boxes N. Each of the following N lines contains two integers in the range [1;10^9] – the row and column of each box respectively. The next line contains the number of rocks M. Each of the following M lines contains the row and column of each rock, again integers in the range [1;10^9]. The upper left corner of the table is on the first row and on the first column.

Output.

On the first line of the standard output print a single integer K - the minimal number of bombs needed to destroy all boxes. On each of the following K lines print 3 integers h x y, interpreted as follows:

- h is 1 if the bomb is horizontal, and 0 if it is vertical
- x and y are the row and column of where the bomb should be placed and $1 \le x,y \le 10^9$.

The cell in which a bomb is placed should be empty at the moment the bomb is being placed. If there is more than one solution using a minimum number of bombs, print any of them. You should output the lines in the order of the placement of the bombs.

Constraints. $1 \le N, M \le 4 \cdot 10^5$

Subtasks

Subtask	Points	N, M	Additional constraints
1	8	≤ 10	None
2	10	$\le 2 \cdot 10^3$	There are at most two boxes in a single row or a single column.
3	18	$\le 2 \cdot 10^3$	Only rows 2 and 4 contain non-empty cells
4	26	$\le 2 \cdot 10^3$	None
5	38	$\le 4 \cdot 10^5$	None

Sample

Input	Output	Explanation
8 6 2 11 4 2 9 6 7 4 9 6 4 6 9 11 9 2 11 7 8 9	5 1 6 5 1 6 4 0 3 9 1 11 5 1 11 10	The bombs have the following effects: ● Horizontal bomb at (6, 5) destroys the boxes at (6, 4) and (6, 7) ● Horizontal bomb at (6, 4) destroys the boxes at (6, 2) and (6, 9). Note that (6, 4) has become an empty cell. ● Vertical bomb at (3, 9) destroys the boxes at (2, 9) and (4, 9) ● Horizontal bomb at (11, 5) destroys the box at (11, 4). Note that the shot to the right is blocked by the rock. ● Horizontal bomb at (11, 10) destroys the box at (11, 9). Any correct solution using 5 bombs will be accepted.

Task 2. Present10

Let us consider a sequence of alternatively changing zeroes and ones, starting with one. This sequence can be seen as a binary representation of a positive integer. We want to present it as a sum of **different** binary numbers, composed only of ones (i.e. 1, 11, 111 and etc). For some sequences such presentation is possible, for others not.

For example: $1010_2=11_2+111_2$; $1010101_2=111_2+1111_2+111111_2$; 10101010101_2 cannot be presented as desired.

Write a program **present10** to find for a given sequence of zeros and ones the number of summands in one presentation as a sum of different binary numbers, composed only of ones, or determines that there is no such presentation.

Input.

The first line of the standard input contains only one positive integer n – the length of the considered sequence.

Output.

The only line of the standard output should only contain one non-negative integer: the number of different summands of the desired presentation or 0, if there is no such presentation.

If there is more than one possible solution, output the number of summands in any of them.

Evaluation.

The tests are grouped in pairs. The points for a pair are obtained only if the two tests are solved correctly.

Constraints. $1 \le n \le 2.10^9$

- In 10% of the test pairs the number $n \le 60$
- In 30% of the test pairs the number $n \le 10^3$
- In 60% of the test pairs the number $n \le 10^6$

Samples

Input	Output	Explanation
6	4	$101010_2=1_2+11_2+111_2+11111_2$ (42=1+3+7+31)
5	0	The number $10101_2=21$ cannot be presented as desired.

Task 3. Music Festival

It's time for the Shumen Music Festival! The lineup consists of N performers, each of whom will give two concerts. The time period of the first concert by performer i is $[a_i; b_i]$, while the time period of the second one is $[c_i; d_i]$.

Alice is wondering which concerts to go to. Because she likes all music genres, she would like to attend at least one concert by each performer. The problem is, there could be multiple concerts taking place at the same time but Alice could only attend at most one of them. That's why she needs your help!

She is asking you to write program **festival** that selects one concert by each performer in such a way, that no two of the selected concerts overlap if it is possible, or determines that this is impossible. Two concerts that are at time periods $[s_i; e_i]$ and $[s_j; e_j]$ overlap, if there exists a moment in time t so that it meets the conditions: $s_i \le t \le e_i$ and $s_j \le t \le e_j$.

Input.

The first line of the standard input consists of a single integer N - the number of performers. N lines follow, describing the time periods of the concerts: line $i+1$ consists of the moments of time, integer a_i, b_i, c_i, and d_i for the i-th performer.

Output.

The first line of the standard output must be a single word: "Yes", if it's possible to satisfy Alice's requirements, and "No", otherwise. If your program outputs "Yes", there should be N more lines, indicating the concerts Alice should attend: Line $i+1$ should contain "1" if Alice should attend the first concert of the performer with number i, or "2" if Alice should go to the second concert of the performer with number i. In case there are multiple possible solutions, you can print any one of them.

Constraints. $1 \le N \le 10^5$; $0 \le a_i \le b_i < c_i \le d_i \le 10^9$

Subtasks.

Subtask	Points	*N*	Additional constraints
1	10	≤ 20	None
2	15	$\le 4.10^3$	$a_i = b_i$ and $c_i = d_i$ for every i
3	32	$\le 4.10^3$	None
4	43	$\le 10^5$	None

Samples

Input	Output	Explanation
3 0 0 1 6 2 2 3 4 1 3 5 5	Yes 1 2 2	Another valid solution is: 1 1 2
3 0 0 1 6 2 2 3 4 0 0 1 3	No	
4 2 2 3 3 0 0 4 4 0 0 1 1 1 1 2 2	Yes 1 2 1 1	
10 21 22 33 35 16 18 26 27 9 10 14 23 0 1 15 34 12 17 19 32 4 28 30 38 5 6 13 31 2 3 20 37 24 25 29 36 7 8 11 39	Yes 1 2 1 1 1 2 1 1 1 1	Here the solution is unique.