

## Jobs

You have a successful business where you make money by completing jobs for your clients. Currently, you can choose from  $N$  one-time jobs, numbered from 1 to  $N$ .

Completing job  $i$  will make you a profit of  $x_i$  euros. The profit may also be negative ( $x_i < 0$ ).

Some jobs depend on another job. That is, there may be a job numbered  $p_i$  that must be completed before the  $i$ -th job can be started. Hence, a job with a large profit may be less attractive than it seems if it depends on a job with a negative profit. If  $p_i = 0$ , the  $i$ -th job has no dependency.

You currently have  $s$  euros and can decide which jobs to do and in which order to do them, as long as the dependencies are respected. Moreover, the amount of money you own may not become negative at any point.

## Task

Calculate the maximum profit you can make by choosing to complete some (possibly none) of the  $N$  jobs in a selected order.

## Input

The first line contains two integers  $N$  and  $s$  – the number of jobs and the amount of money you initially own respectively.

Then,  $N$  lines follow. The  $i$ -th of them contains two integers  $x_i$  and  $p_i$  – the profit and the number of the prerequisite job for the  $i$ -th job, respectively. If  $p_i = 0$ , the  $i$ -th job does not have a job dependency.

## Output

Your program should output a single integer – the maximum profit that you can make.

## Examples

Input	Output	Explanation
6 1 3 0 -3 1 -5 0 2 1 6 3 -4 5	6	<p>To maximize profit, you should pick jobs 1, 4, 3 and 5 in the following order:</p> <ul style="list-style-type: none"> <li>• Job 1: money 1 <math>\rightarrow</math> 4,</li> <li>• Job 4 (prerequisite 1 complete): money 4 <math>\rightarrow</math> 6,</li> <li>• Job 3: money 6 <math>\rightarrow</math> 1,</li> <li>• Job 5 (prerequisite 3 complete): money 1 <math>\rightarrow</math> 7.</li> </ul> <p>Overall, the total profit is 7 - 1 = 6 (current money minus starting money).</p>

## Constraints

- $1 \leq N \leq 3 \cdot 10^5$
- $0 \leq s \leq 10^{18}$
- $-10^9 \leq x_i \leq 10^9$  (for all  $1 \leq i \leq N$ )
- $0 \leq p_i < i$  (for all  $1 \leq i \leq N$ )

## Subtasks

No.	Points	Additional constraints
1	11	$s = 10^{18}$ .
2	14	$N \leq 2000$ and for all jobs, either $p_i = 0$ , or $p_i = i - 1$ .
3	15	For all jobs, either $p_i = 0$ , or $p_i = i - 1$ .
4	29	$N \leq 2000$ .
5	31	No additional constraints.

# Portal

You think it would be funny to prank your best friend by placing them on cell  $(0, 0)$  of an infinite grid of coloured cells. The friend then moves around the grid indefinitely, one step at a time, always moving to one of the four adjacent cells.

$N$  of the cells on the grid contain a portal. Once your friend steps on a portal, they instantly teleport to a random portal (which might be the one they just stepped on, or might be a different one). If there is a portal at the cell  $(0, 0)$ , your friend is also teleported at the start when they get placed on the grid.

As part of the prank, you want to trick your friend into not noticing that there are portals at all. The only thing your friend sees is the colour of the cell they are currently at, so you should make sure that from your friend's perspective the colours of the tiles never change. In particular, if your friend thinks they have entered a cell more than once (for example by moving left and then immediately right), they should see the same color as the first time they think they entered the cell.

Note: when your friend steps on a portal, they will see both the colour of the cell they step on, and the one they are teleported to. You will therefore need to colour all portal cells the same colour to avoid teleportations being immediately obvious.

A simple solution would be to colour all cells the same colour. But colours are nice! So you would like to use as many colours as you can.

Let's consider an example where the portals are placed at cells  $(1, 1)$ ,  $(1, 3)$  and  $(3, 2)$ , and your friend makes the following sequence of moves: up, right, down, left.

### After 0 steps

(-1,4)	(0,4)	(1,4)	(2,4)	(3,4)	(4,4)
(-1,3)	(0,3)	(1,3)	(2,3)	(3,3)	(4,3)
(-1,2)	(0,2)	(1,2)	(2,2)	(3,2)	(4,2)
(-1,1)	(0,1)	(1,1)	(2,1)	(3,1)	(4,1)
(-1,0)	(0,0)	(1,0)	(2,0)	(3,0)	(4,0)
(-1,-1)	(0,-1)	(1,-1)	(2,-1)	(3,-1)	(4,-1)

Initial position. First time your friend sees colour of cell (0, 0)

### After 1 step

(-1,4)	(0,4)	(1,4)	(2,4)	(3,4)	(4,4)
(-1,3)	(0,3)	(1,3)	(2,3)	(3,3)	(4,3)
(-1,2)	(0,2)	(1,2)	(2,2)	(3,2)	(4,2)
(-1,1)	(0,1)	(1,1)	(2,1)	(3,1)	(4,1)
(-1,0)	(0,0)	(1,0)	(2,0)	(3,0)	(4,0)
(-1,-1)	(0,-1)	(1,-1)	(2,-1)	(3,-1)	(4,-1)

Go up to cell (0, 1)

### After 2 steps

(-1,4)	(0,4)	(1,4)	(2,4)	(3,4)	(4,4)
(-1,3)	(0,3)	(1,3)	(2,3)	(3,3)	(4,3)
(-1,2)	(0,2)	(1,2)	(2,2)	(3,2)	(4,2)
(-1,1)	(0,1)	(1,1)	(2,1)	(3,1)	(4,1)
(-1,0)	(0,0)	(1,0)	(2,0)	(3,0)	(4,0)
(-1,-1)	(0,-1)	(1,-1)	(2,-1)	(3,-1)	(4,-1)

Go right to cell (1, 1) and teleport to any of the three portals

### After 3 steps




(-1,4)	(0,4)	(1,4)	(2,4)	(3,4)	(4,4)
(-1,3)	(0,3)	(1,3)	(2,3)	(3,3)	(4,3)
(-1,2)	(0,2)	(1,2)	(2,2)	(3,2)	(4,2)
(-1,1)	(0,1)	(1,1)	(2,1)	(3,1)	(4,1)
(-1,0)	(0,0)	(1,0)	(2,0)	(3,0)	(4,0)
(-1,-1)	(0,-1)	(1,-1)	(2,-1)	(3,-1)	(4,-1)

Go down

### After 4 steps

(-1,4)	(0,4)	(1,4)	(2,4)	(3,4)	(4,4)
(-1,3)	(0,3)	(1,3)	(2,3)	(3,3)	(4,3)
(-1,2)	(0,2)	(1,2)	(2,2)	(3,2)	(4,2)
(-1,1)	(0,1)	(1,1)	(2,1)	(3,1)	(4,1)
(-1,0)	(0,0)	(1,0)	(2,0)	(3,0)	(4,0)
(-1,-1)	(0,-1)	(1,-1)	(2,-1)	(3,-1)	(4,-1)

Go left. Your friend thinks they're back to the start, but they might be at any of the coloured positions.

	Where your friend thinks they are
	Where your friend might be
	Cell contains portal

After the sequence of moves the friend thinks that they're back at the starting cell (0, 0), but in reality they might also end up in (0, 2) or (2, 1). They already saw the colour of cell (0, 0) at the beginning, so if they see a different colour now, they will realise there must be portals. We don't want that to happen, so we must choose the same colour for these 3 cells.

There is no sequence of moves where your friend would think that they end up on cell (0, 0) when they actually end up on (1, 0), so these cells can be safely coloured with different colours.

Below you can see a colouring with 4 colours for the example above. It is not possible to use more than 4 colours for this example.

(-1,4)	(0,4)	(1,4)	(2,4)	(3,4)	(4,4)
(-1,3)	(0,3)	(1,3)	(2,3)	(3,3)	(4,3)
(-1,2)	(0,2)	(1,2)	(2,2)	(3,2)	(4,2)
(-1,1)	(0,1)	(1,1)	(2,1)	(3,1)	(4,1)
(-1,0)	(0,0)	(1,0)	(2,0)	(3,0)	(4,0)
(-1,-1)	(0,-1)	(1,-1)	(2,-1)	(3,-1)	(4,-1)

Let's consider a different example with portals at cells  $(0,0)$ ,  $(0,1)$ ,  $(1,0)$ ,  $(0,-1)$  and  $(-1,0)$ . Say your friend tries to reach cell  $(1,3)$  by going right once and then up 3 times. One possibility is that they end up at cell  $(0,0)$  if they got teleported there at the start and after each step. If your friend now backtracks to what they think is cell  $(0,0)$  by going down 3 times and left once, and doesn't get teleported away from their current cell while doing so, they will end up at  $(-1,-3)$ . Your friend will think that they are at cell  $(0,0)$  for the second time, and will expect to see the same colour. So you need to colour  $(-1,-3)$  and  $(0,0)$  with the same colour.

Note that there was nothing special about our initial choice of cell  $(1,3)$ . You can similarly show that other cells have to share a colour with  $(0,0)$ .

## Task

Calculate the maximum number of colours you can use while making sure that your friend won't notice the existence of portals.

## Input

The first line contains the integer  $N$  - the number of portals.

The next  $N$  lines contain two integers each. The  $i$ -th of these lines contains  $x_i$  and  $y_i$ , indicating that there is a portal at cell  $(x_i, y_i)$ .

## Output

Print a single integer - the maximum number of colours that can be used without your friend noticing the portals, or  $-1$  if you can use an infinite number of colours.

## Examples

Input	Output	Explanation
3 1 1 1 3 3 2	4	First example discussed in the task description.
5 0 0 1 0 -1 0 0 1 0 -1	1	Second example discussed in the task description.
1 1 -1	-1	Your friend can only be "teleported" to the same cell that the portal is located in, so there is no way for them to notice the existence of portals even if every cell is coloured differently.

## Constraints

- $1 \leq N \leq 10^5$
- $-10^6 \leq x_i, y_i \leq 10^6$  (for all  $1 \leq i \leq N$ )
- No two portals share the same coordinates.

## Subtasks

No.	Points	Additional constraints
1	1	$N \leq 2$ .
2	10	$N \leq 3$ .
3	10	For all integers $x_1, x_2, y_1, y_2$ : if there are portals at the locations $(x_1, y_1)$ and $(x_2, y_2)$ , then there is also a portal at the location $(x_1, y_2)$ .
4	29	$N \leq 100$ and $-100 \leq x_i, y_i \leq 100$ for all $1 \leq i \leq N$ .
5	15	$N \leq 2000$ .
6	35	No additional constraints.

# Trains

You have arrived in Vilnius and want to visit various cities in Lithuania.

Cities in Lithuania are located on a straight line and numbered from 1 to  $N$  sequentially. Vilnius is assigned the number 1.

Each city has a train station with a single train route that operates from that station. You can only get on a train at the start of its route, though you can get off at any of its stops. Trains starting their route at the  $i$ -th city make a stop every  $d_i$  cities, and their route consists of  $x_i$  stops (not including the starting city). If  $d_i = 0$ , the trains operating from the  $i$ -th city are currently out of service, so you cannot get on them.

To be more precise, if you get on a train in the  $i$ -th city, you can get off the train in any city numbered  $i + t \cdot d_i$ , where  $1 \leq t \leq x_i$ . Note that since you only want to visit cities in Lithuania, you won't ride further than the  $N$ -th city, even if the train has more stops on its route.

## Task

You are going to visit some cities, using trains to travel between them. As you plan your trip, you start to wonder how many different options you have for your journey that starts at Vilnius. Two journeys are different if they make stops at different sequences of cities.

Calculate this number and print the answer modulo  $10^9 + 7$ .

## Input

The first line contains one integer  $N$  – the number of cities.

Then  $N$  lines follow. The  $i$ -th of them contains two integers  $d_i$  and  $x_i$  – the numbers describing the route starting in the  $i$ -th city.

## Output

Output a single integer – the number of ways you can visit some of the  $N$  cities taken modulo  $10^9 + 7$ .

## Examples

Input	Output	Explanation
5 1 3 2 1 1 3 0 10 3 5	7	There are 7 possible journeys you could take: <ul style="list-style-type: none"><li>• 1</li><li>• 1 → 2</li><li>• 1 → 2 → 4</li><li>• 1 → 3</li><li>• 1 → 3 → 4</li><li>• 1 → 3 → 5</li><li>• 1 → 4</li></ul>

## Constraints

- $1 \leq N \leq 10^5$
- $0 \leq d_i \leq 10^9$  (for each  $1 \leq i \leq N$ )
- $0 \leq x_i \leq 10^9$  (for each  $1 \leq i \leq N$ )

## Subtasks

No.	Points	Additional constraints
1	8	$n \leq 15$ .
2	13	$n \leq 10^4$ .
3	16	For all trains, $d_i = 1$ .
4	34	For all trains, $x_i = 10^9$ .
5	29	No additional constraints.