C. Woodcutters

time limit per test

 1 second

memory limit per test

 256 megabytes

input

 standard input

output

 standard output

Little Susie listens to fairy tales before bed every day. Today's fairy tale was about wood cutters and the little girl immediately started imagining the choppers cutting wood. She imagined the situation that is described below.

There are *n* trees located along the road at points with coordinates *x*1, *x*2, ..., *xn*. Each tree has its height *hi*. Woodcutters can cut down a tree and fell it to the left or to the right. After that it occupies one of the segments [*xi* - *hi*, *xi*] or [*xi*;*xi* + *hi*]. The tree that is not cut down occupies a single point with coordinate *xi*. Woodcutters can fell a tree if the segment to be occupied by the fallen tree doesn't contain any occupied point. The woodcutters want to process as many trees as possible, so Susie wonders, what is the maximum number of trees to fell.

**Input**

The first line contains integer *n* (1 ≤ *n* ≤ 105) — the number of trees.

Next *n* lines contain pairs of integers *xi*, *hi* (1 ≤ *xi*, *hi* ≤ 109) — the coordinate and the height of the *і*-th tree.

The pairs are given in the order of ascending *xi*. No two trees are located at the point with the same coordinate.

**Output**

Print a single number — the maximum number of trees that you can cut down by the given rules.

**Examples**

**input**

5

1 2

2 1

5 10

10 9

19 1

**output**

3

**input**

5

1 2

2 1

5 10

10 9

20 1

**output**

4

**Note**

In the first sample you can fell the trees like that:

* fell the 1-st tree to the left — now it occupies segment [ - 1;1]
* fell the 2-nd tree to the right — now it occupies segment [2;3]
* leave the 3-rd tree — it occupies point 5
* leave the 4-th tree — it occupies point 10
* fell the 5-th tree to the right — now it occupies segment [19;20]

In the second sample you can also fell 4-th tree to the right, after that it will occupy segment[10;19].

一开始看到这道题的tag是dp和greedy……然后我就开始用dp的方法想……

那么对于一颗树来说，其能不能倒下，首先得左边和右边倒下以后都不能碰到别的树……

然后还得注意之前如果有树向右倒下来的话那就不能冲突……然后就觉得好像dp的想法不大适用于这道题……这个的确是可以把它分解成子问题，但是子问题对于母问题的影响似乎不是很大，唯一的影响就只有树倒下来不能冲突……

然后大概觉得这道题可以贪心，条件是如果一棵树可以倒向左边，或者可以倒向右边，则这棵树就可以砍倒了，但是这里还有一点要注意的就是和之前倒下来的树不能冲突……这个一开始没注意到

然后发现了第二个错误……就是一开始为了防止越界我是把最后一棵树直接在输出里+1的，因为肯定向右倒……右边肯定没有树……同理第一棵树肯定向左倒……

然后就A题了……不愧是有5000多人A的C题……

#include<iostream>

#include<string>

#include<algorithm>

#include<queue>

#include<vector>

#include<sstream>

#include<stack>

#include<map>

#include<cstring>

#include<climits>

#define MAX 20000

using namespace std;

vector<pair<int,int>>trees;

int main()

{

          int num;

          cin >> num;

          trees.resize(num);

          for (int i =0; i < num; i++)

          {

                    cin >> trees[i].first;

                    cin >> trees[i].second;

          }

          int cnt =1;

          intformer = 1;

          if (num== 1)

                    cout << cnt;

          else

          {

                    for (int i =1; i < trees.size() -1; i++)

                    {

                               if(trees[i].first -trees[i].second > trees[i- 1].first&&trees[i].first -trees[i].second >former)

                                         cnt++;

                               else if (trees[i].first+ trees[i].second < trees[i + 1].first)

                               {

                                         former =trees[i].first + trees[i].second;

                                         cnt++;

                               }

                    }

                    cout << cnt + 1;

          }

}