

Screenshots:

**Note:** The barriers in the input may overlap.

#### OUTPUT

Output a single integer, the number of ants that will be ever blocked at some point in their march.

#### CONSTRAINTS

$1 \leq N \leq 10^5$   
 $1 \leq x_i, y_i, d_i \leq 10^9$

Sample Input	Sample Output
2 1 1 4 7 3 5	11

Time Limit: 1  
Memory Limit: 256  
Source Limit:

#### Explanation

Here 5 ants will be blocked on points (1,1) , (2, 1) , (3, 1), (4, 1) and (5, 1).

6 ants will be blocked on (7, 3), (8, 3), (9, 3), (10, 3), (11, 3), (12, 3).

In total, 11 ants are blocked in their journey.

#### Contributors:



Aditya Shah



Paweł Kacprzak

COMPILE & TEST

SUBMIT

Submission ID: 54798033 / 3 seconds ago

RESULT: Accepted

[Refer judge environment](#)

Score	Time (sec)	Memory (KiB)	Language
20.0	2.3911	17524	Python 3

Input	Result	Time (sec)	Memory (KiB)	Score	Your Output	Correct Output	Diff
Input #1	Accepted	0.034717	3772	5			
Input #2	Accepted	0.051081	3760	5			
Input #3	Accepted	0.034099	3772	10			
Input #4	Accepted	0.034458	3692	10			
Input #5	Accepted	0.043862	3760	10			
Input #6	Accepted	0.066984	4784	10			
Input #7	Accepted	0.426014	17524	10			
Input #8	Accepted	0.425555	17516	10			
Input #9	Accepted	0.416164	17504	10			
Input #10	Accepted	0.415785	17508	10			

```
#Inputting number of barriers
```

```
N = int(input())
```

```
#the coordinates of the barriers
```

```
barriers_coordinates = []
```

```
#getting inputd of the coordinates from the user
```

```
for i in range(N):  
    x, y, d = map(int, input().strip().split())  
    barriers_coordinates.append((x, x + d))
```

```
barriers_coordinates.sort()
```

```
#Counting the ants that were blocked and the points at which they were  
blocked on
```

```
the_blocked_ants = 0
```

```
the_blocked_points = 0
```

```
#the blocked ants in coordinates

for the_barrier in barriers_coordinates:
    if the_barrier[0] >= the_blocked_points:
        the_blocked_points = the_barrier[0]
        if the_blocked_points < the_barrier[1]:
            the_blocked_ants += (the_barrier[1] - the_blocked_points) + 1
            the_blocked_points = the_barrier[1] + 1

    elif the_blocked_points <= the_barrier[1]:
        the_blocked_ants += (the_barrier[1] - the_blocked_points) + 1
        the_blocked_points = the_barrier[1] + 1

print(the_blocked_ants)
```