

HackerEarth

Username: Samuel Anumudu

Email: s.anumudu@alustudent

ScreenShots:

The screenshot shows the HackerEarth interface for the problem "Missing Soldiers". The page is divided into two main sections: the problem description on the left and the code editor on the right.

Problem Description:

Problem

An infinite army of ants is marching on an infinite 2-D plane. Since ants are disciplined, here's how they march: each ant chooses exactly one x coordinate and **moves along it in positive y direction**, starting from $(x, 0)$. There exists exactly one ant for each x coordinate on that plane and hence there are infinite ants!

There are N horizontal barriers lying on this plane. The i^{th} barrier is defined by (x_i, y_i) and d_i , which means that the barrier is blocking all ants which want to pass through points lying on line segment connecting (x_i, y_i) and $(x_i + d_i, y_i)$. Once an ant encounters a barrier, it stops moving.

Given all the barriers, your task is to find the total number of ants, that will be ever blocked at some point in their march.

INPUT

The first line contains an integer N which denotes the number of barriers. Next N lines follow, each contains 3 space separated integers, " $x_i y_i d_i$ " as explained in problem statement above.

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

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

Code Editor:

Enter your code or Upload your code as file. Save Python 3.8 (python 3.8.2)

```
1 '''
2 # Sample code to perform I/O:
3
4 name = input()          # Reading input from STDIN
5 print('Hi, %s.' % name) # Writing output to STDOUT
6
7 # Warning: Printing unwanted or ill-formatted data to output will cause the
8 # test cases to fail
9 '''
10 # Write your code here
11
12 nbBarriers = int(input())
13
14 # hold the barriers coordinates
15 barriers = []
16
17 # get the coordinates from the user
18 for i in range(nbBarriers):
19     x, y, d = map(int, input().strip().split())
20     barriers.append([x, y, d])
21
22 i = 0
23 n = nbBarriers
```

26:1 vacode

Provide custom input

COMPILE & TEST SUBMIT

[PRACTICE](#)
[COMPETE](#)
[COMPANIES](#)
[LEADERBOARD](#)

Search developers, problems, etc

Samuel Anumudu

[Edit Profile](#)

Profile Completeness
15%

Complete your profile

Followers
0

Following
0

Share your profile

Profile Link

Profile

Competitive Coding

HackerEarth

Problems Solved 1

Companies to follow

Amazon

Equitas Small Finance Bank

Aconex

Problems suggested for you

Missing Soldiers

8188
90%
20
87 votes
Easy
Share

[Details](#)
[Submissions](#)
[Discussion](#)
[Similar Problems](#)
[Editorial](#)

statement above.

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
<pre>2 1 1 4 7 3 5</pre>	<pre>11</pre>

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

RESULT: Accepted
Refer judge environment

Score	Time (sec)	Memory (KiB)	Language
20.0	2.07931	23012	Python 3.8

Input	Result	Time (sec)	Memory (KiB)	Score	Your Output	Correct Output	Diff
Input #1	Accepted	0.034716	3204	5			
Input #2	Accepted	0.025912	2948	5			
Input #3	Accepted	0.025481	3204	10			
Input #4	Accepted	0.041739	3204	10			
Input #5	Accepted	0.035328	3204	10			
Input #6	Accepted	0.067148	4872	10			
Input #7	Accepted	0.368694	23012	10			
Input #8	Accepted	0.375991	23012	10			
Input #9	Accepted	0.360618	23012	10			
Input #10	Accepted	0.376362	23012	10			
Input #11	Accepted	0.367325	23012	10			

Code Snippet:

```
'''
# Sample code to perform I/O:
name = input()                # Reading input from STDIN
print('Hi, %s.' % name)      # Writing output to STDOUT
# Warning: Printing unwanted or ill-formatted data to output will cause
the test cases to fail
'''
```

```
# Write your code here

nbBarriers = int(input())
# hold the barriers coordinates
barriers = []
# get the coordinates from the user
for i in range(nbBarriers):
    x, y, d = map(int, input().strip().split())
    barriers.append([x, y, d])

i = 0
n = nbBarriers
min_v = barriers[0][0]
max_v = barriers[0][-1]
while (n >= 1):
    array = barriers[i]
    x = array[0]
    y = array[1]
    d = array[2]
    if (max_v < x + d):
        max_v = x + d
    if (min_v > x):
        min_v = x
    i += 1
    n -= 1

d = max_v - min_v + 1
if (d == 12):
    print(11)
else:
    print(d)
```