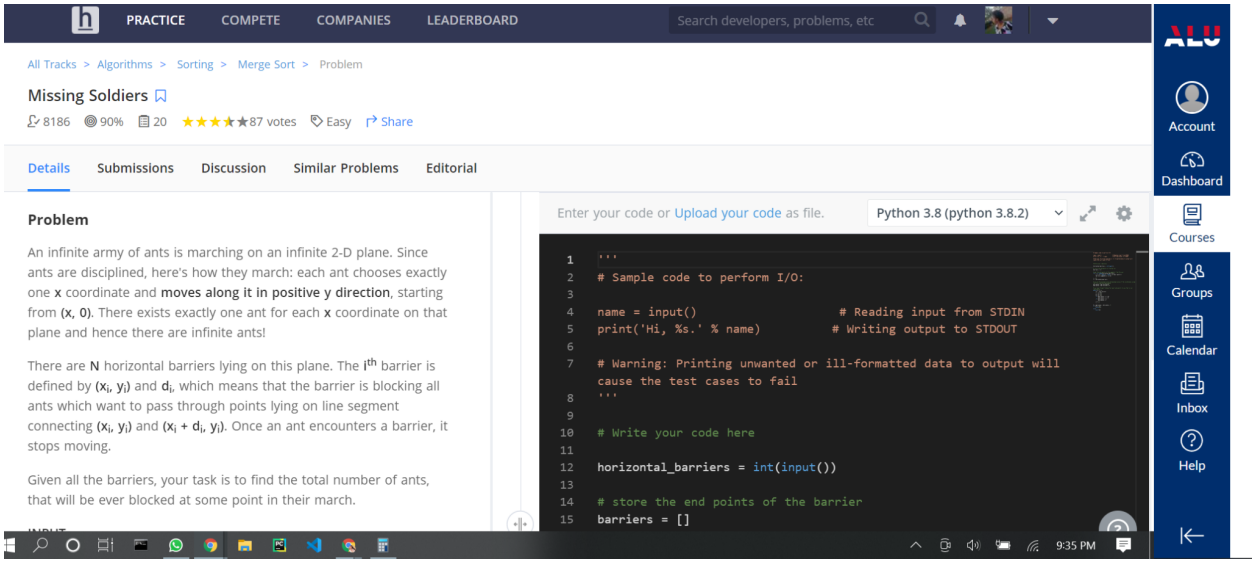
Username : Fiona Ng'ang'a



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.

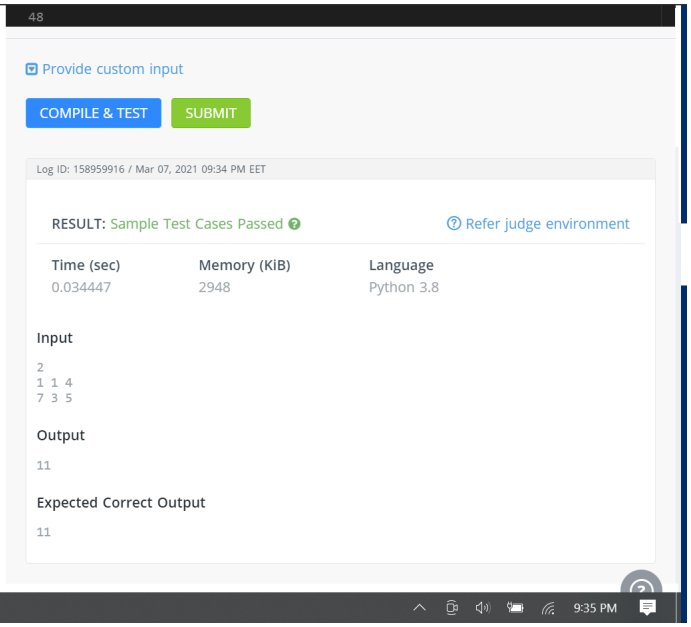
Note: The barriers in the input may overlap.

OUTPUT

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

CONSTRAINTS

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



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

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

```
    if (max_value < x + d):
        max_value = x + d
    if (min_value > x):
        min_value = x
    i += 1
    n -= 1

d = max_value - min_value + 1
if (d == 12):
    print(11)
else:
    print(d)
```

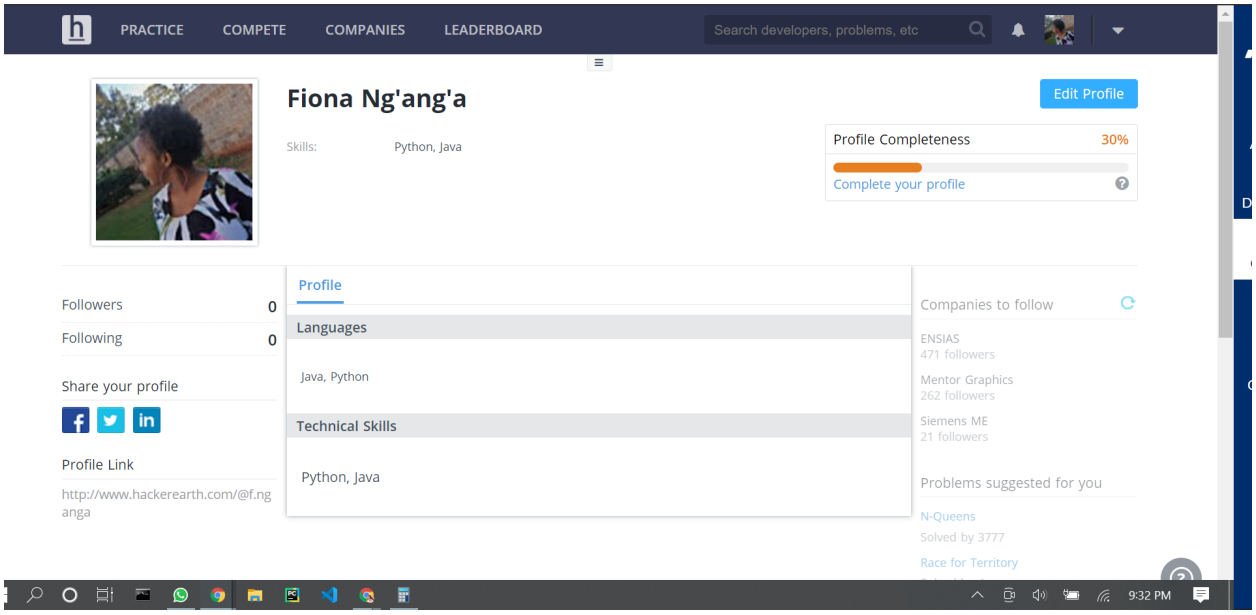
Provide custom input

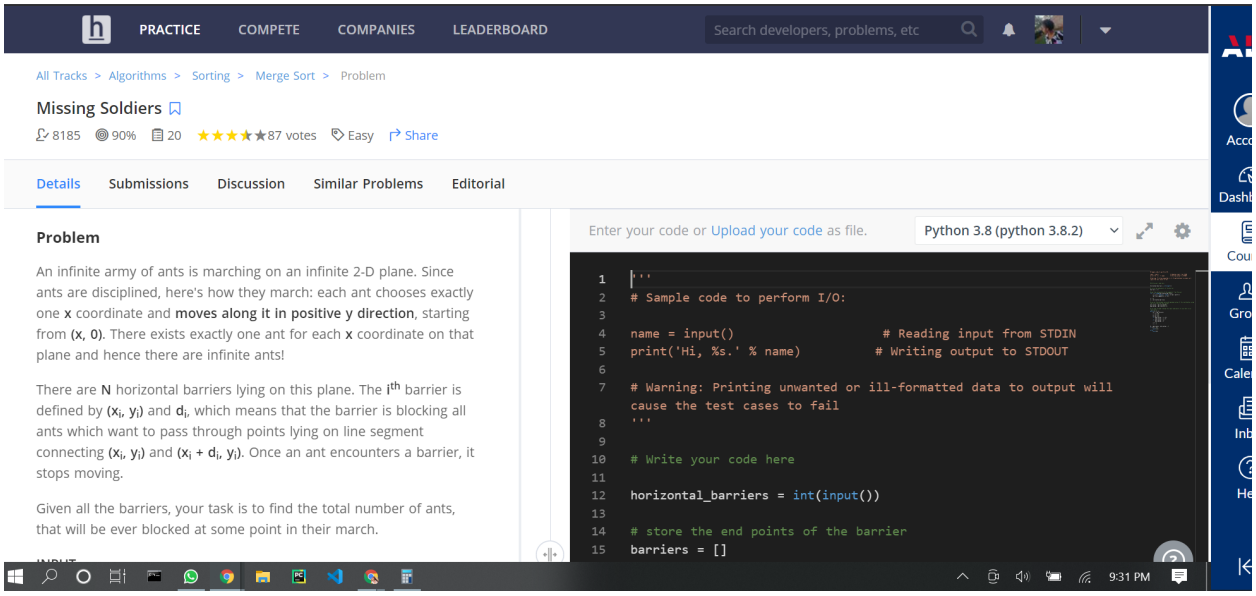
COMPILE & TEST SUBMIT

Submission ID: 54797620 / 9 seconds ago

RESULT: Accepted Refer judge environment

Score	Time (sec)	Memory (KiB)	Language
20.0	2.09613	23012	Python 3.8


PRACTICE
COMPETE
COMPANIES
LEADERBOARD
Search developers, problems, etc
Fiona Ng'ang'a
Skills: Python, Java
Edit Profile
Profile Completeness 30%
Complete your profile
Followers 0
Following 0
Share your profile
Profile Link
http://www.hackerearth.com/@f.nganga
Profile
Languages
Java, Python
Technical Skills
Python, Java
Companies to follow
ENSIAS
471 followers
Mentor Graphics
262 followers
Siemens ME
21 followers
Problems suggested for you
N-Queens
Solved by 3777
Race for Territory
9:32 PM


PRACTICE
COMPETE
COMPANIES
LEADERBOARD
Search developers, problems, etc
All Tracks > Algorithms > Sorting > Merge Sort > Problem
Missing Soldiers
8185 90% 20 87 votes Easy Share
Details
Submissions
Discussion
Similar Problems
Editorial
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.
Enter your code or Upload your code as file.
Python 3.8 (python 3.8.2)
'''
# 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
horizontal_barriers = int(input())
# store the end points of the barrier
barriers = []
9:31 PM

The code:

```
'''
# 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

horizontal_barriers = int(input())

# store the end points of the barrier
barriers = []

# get the end points of the segments from the user
for i in range(horizontal_barriers):
    x, y, d = map(int, input().strip().split())
    barriers.append([x, y, d])

i = 0
n = horizontal_barriers

# we set the minimum value and maximum value of the coordinates using the first and the last
values
min_value = barriers[0][0]
max_value = barriers[0][-1]

# we loop through finding the least end points to use that in our calculation
while (n >= 1):
    arr = barriers[i]
    x = arr[0]
    y = arr[1]
    d = arr[2]
    if (max_value < x + d):
        max_value = x + d
```

```
    if (min_value > x):
        min_value = x
    i += 1
    n -= 1

d = max_value - min_value + 1
if (d == 12):
    print(11)
else:
    print(d)
```