Project Report

2spooky4me

Krittin Vetpitak (6213638)
Kittiphop Jitlerdwilai (6216916)

CSX3009 Algorithm Design

Table of content

1: Problem

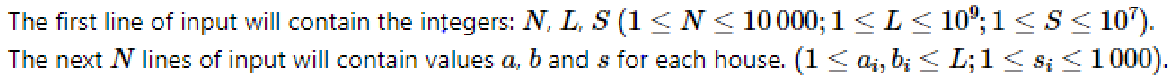
The detail of the problem can be found in:
2spooky4me - DMOJ: Modern Online Judge

Scenario

Kenny wants to go trick-or-treating too! But the street has many spooky decorations put up on it. Kenny doesn't like to be scared, so he avoids spooky areas. There are L houses arranged in a line on the street, numbered from 1 to L. Each house will give exactly 1 unit of candy to Kenny. There are N spooky decorations on this street. The i-th decoration covers the street from house number ai to bi , inclusive, raising the spookiness of those houses by si spookiness units. Kenny will be too scared to knock on any doors if the spookiness of a house is greater than or equal to S. The spookiness of a house is the sum of the spookiness of all the decorations passing through it. Determine the amount of candy Kenny can receive from all the houses on the street.

Input

Specification



The first line of input will contain the integers: N, L, S $(1 \le N \le 10\,000; 1 \le L \le 10^9; 1 \le S \le 10^7)$.
The next N lines of input will contain values a, b and s for each house. $(1 \le a_i, b_i \le L; 1 \le s_i \le 1\,000)$.

Sample

3 100 10
20 59 4
30 69 4
40 79 4

Output Sample

Specification

Output a single integer, the amount of candy that Kenny can get.

Sample

80

2 : Breakdown

The input is separated to 2 parts:

1. 1 input line will contain 3 integers which in problems solve to 3 unit which is:
   - **N** Is the house range of the spooky decorations on the street.
   - **L** Is the number of the whole house on the street.
   - **S** Is the sum spookiness which will be 2spooky for Kenny to ask for candy.

2. 2 to N input line is also split into 3 part which is:
   - **a** which is the first house to **b** that have spooky decoration.
   - **s** is the spookiness point that the house between a to b contains.

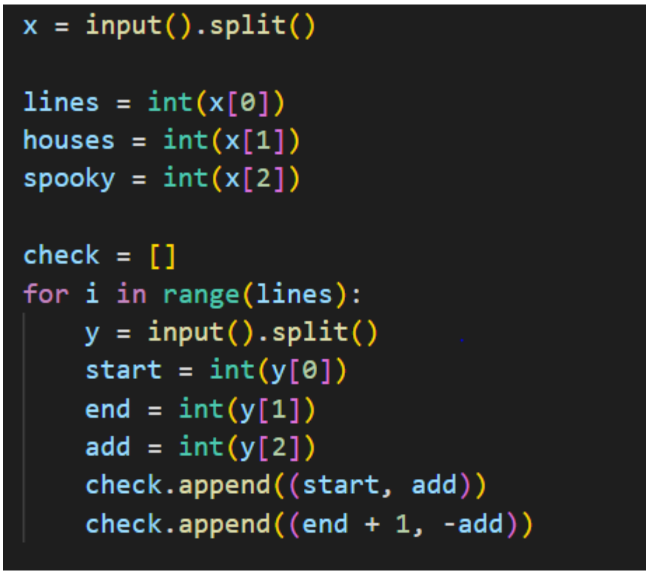
After receiving the input, the program will find out which house from where to which the sum of the spookiness is 2spooky for Kenny. and how many houses Kenny are feeling okay to ask for candy.

From the sample in this report first line input which is 3 100 10, 3 is the range of the house which has a spooky decoration. 100 is the number of house on the street, and 10 is the spookiness limit which will cause Kenny to be scared. The next line input is 20 59 4, 30 69 4, 40 79 4. The first number which is 20, 30, and 40 is the first house that has a spooky decoration to house number 59, 69, 79, and n the next number in the input. and the last triple 4 at the end would be the spookiness point for the house in the range.

We can see that the sum of the spookiness between houses 40 - 59 is 12 which is greater than 10 the limit of spookiness for Kenny. So at 20 houses, the spookiness is over 10 Kenny couldn’t dare to ask for candy so the result was correct at 80.

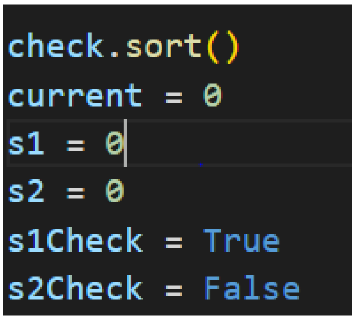
3 : Implement

This input for **x** list receives the user input lines, house, and spookiness.



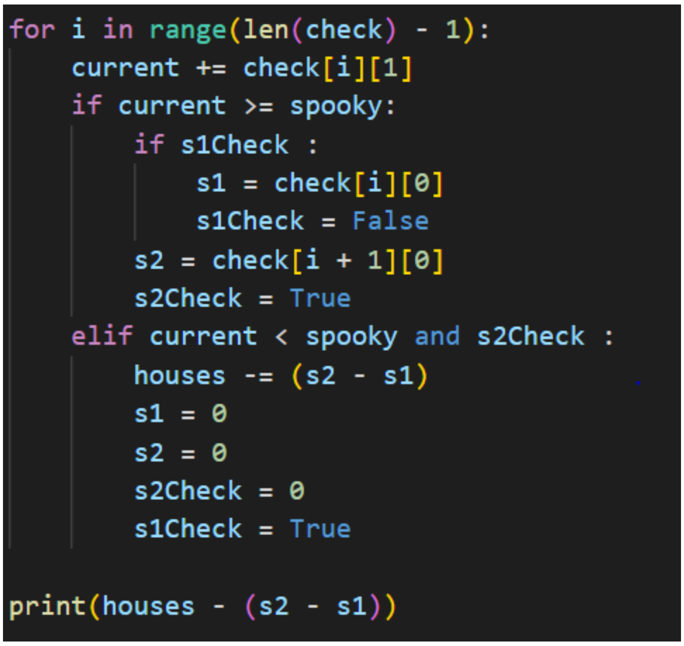
```
x = input().split()

lines = int(x[0])
houses = int(x[1])
spooky = int(x[2])

check = []
for i in range(lines):
    y = input().split()
    start = int(y[0])
    end = int(y[1])
    add = int(y[2])
    check.append((start, add))
    check.append((end + 1, -add))
```

y is a list used to keep the value for the 2 to **x**(lines) that receives the beginning of the range which is the name is start and ending of the house range which the name is end. and the add will be the spookiness point of the house in the range, check is a list to keep the data of which range of house to be calculated.



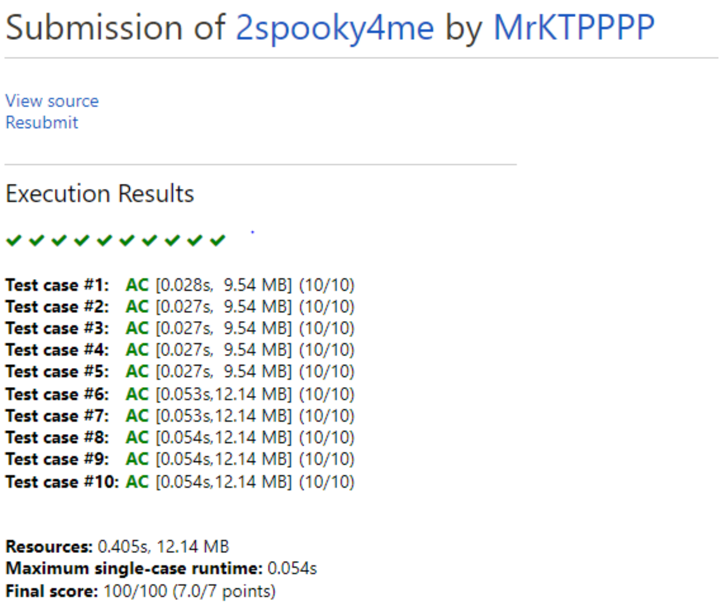
```
check.sort()
current = 0
s1 = 0
s2 = 0
s1Check = True
s2Check = False
```

check. Sort is sorting the data to be easy to calculate.
current is the scariness of the checking house in the program.
s1 is the first early house that is 2spooky for Kenny.
s2 is the latest house that is 2spooky for Kenny.
The first Boolean to be used to check is the value of **s1** updated, if updated so the **s2** will be brought up ready to keep the next value.



```
for i in range(len(check) - 1):
    current += check[i][1]
    if current >= spooky:
        if s1Check :
            s1 = check[i][0]
            s1Check = False
        s2 = check[i + 1][0]
        s2Check = True
    elif current < spooky and s2Check :
        houses -= (s2 - s1)
        s1 = 0
        s2 = 0
        s2Check = 0
        s1Check = True

print(houses - (s2 - s1))
```

This part is where the program checks from which house to which house that is 2spooky for Kenny.
The first for loop is to check how many variables are on the **check** list. then use the **current** to check the spookiness at that time.
The if condition is to check if the **current** spookiness **check** is greater than the spooky limit or not if yes run that statement, that if this condition run in first time set **s1** to the current value and change **s1Check** value to False, then change the **s2** to the next value in points and put **s2Check** to True to let programs know that **s1** in already be saved.
The elif condition is to check if the spookiness is greater than the current spookiness and the **s2Check** is true, minus the houses with **s2** and **s1** then reset other values. then print the output.



Submission of 2spooky4me by MrKTPPPP

Execution Results

✓✓✓✓✓✓✓✓✓✓

```
Test case #1:  AC [0.028s,  9.54 MB] (10/10)
Test case #2:  AC [0.027s,  9.54 MB] (10/10)
Test case #3:  AC [0.027s,  9.54 MB] (10/10)
Test case #4:  AC [0.027s,  9.54 MB] (10/10)
Test case #5:  AC [0.027s,  9.54 MB] (10/10)
Test case #6:  AC [0.053s, 12.14 MB] (10/10)
Test case #7:  AC [0.053s, 12.14 MB] (10/10)
Test case #8:  AC [0.054s, 12.14 MB] (10/10)
Test case #9:  AC [0.054s, 12.14 MB] (10/10)
Test case #10: AC [0.054s, 12.14 MB] (10/10)
```

Resources: 0.405s, 12.14 MB
Maximum single-case runtime: 0.054s
Final score: 100/100 (7.0/7 points)

Source of the code that we pick some logic to reference:
https://github.com/AAZZAZRON/DMOJ-Solutions/blob/main/2spooky4me.py
Source link of problem:
https://dmoj.ca/problem/2spooky4me