



Apple and Orange ★

321.2 more points to get your gold badge!

Rank: 428849 | Points: 528.8/850



Problem

Submissions

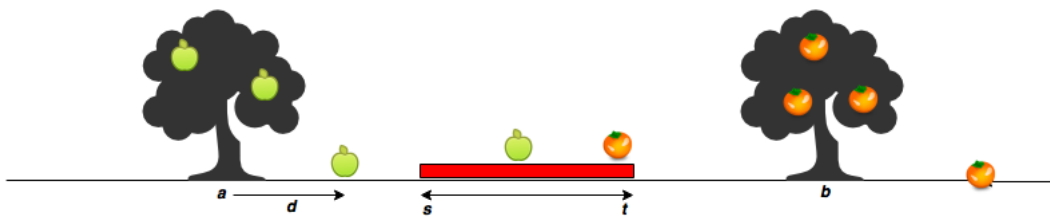
Leaderboard

Editorial

Sam's house has an apple tree and an orange tree that yield an abundance of fruit. Using the information given below, determine the number of apples and oranges that land on Sam's house.

In the diagram below:

- The red region denotes the house, where s is the start point, and t is the endpoint. The apple tree is to the left of the house, and the orange tree is to its right.
- Assume the trees are located on a single point, where the apple tree is at point a , and the orange tree is at point b .
- When a fruit falls from its tree, it lands d units of distance from its tree of origin along the x -axis. *A negative value of d means the fruit fell d units to the tree's left, and a positive value of d means it falls d units to the tree's right. *



Given the value of d for m apples and n oranges, determine how many apples and oranges will fall on Sam's house (i.e., in the inclusive range $[s, t]$)?

For example, Sam's house is between $s = 7$ and $t = 10$. The apple tree is located at $a = 4$ and the orange at $b = 12$. There are $m = 3$ apples and $n = 3$ oranges. Apples are thrown $apples = [2, 3, -4]$ units distance from a , and $oranges = [3, -2, -4]$ units distance. Adding each apple distance to the position of the tree, they land at $[4 + 2, 4 + 3, 4 + -4] = [6, 7, 0]$. Oranges land at $[12 + 3, 12 + -2, 12 + -4] = [15, 10, 8]$. One apple and two oranges land in the inclusive range $7 - 10$ so we print

1
2

Function Description

Complete the `countApplesAndOranges` function in the editor below. It should print the number of apples and oranges that land on Sam's house, each on a separate line.

`countApplesAndOranges` has the following parameter(s):

- s : integer, starting point of Sam's house location.
- t : integer, ending location of Sam's house location.
- a : integer, location of the Apple tree.
- b : integer, location of the Orange tree.
- $apples$: integer array, distances at which each apple falls from the tree.
- $oranges$: integer array, distances at which each orange falls from the tree.

Input Format

The first line contains two space-separated integers denoting the respective values of s and t .

The second line contains two space-separated integers denoting the respective values of a and b .

The third line contains two space-separated integers denoting the respective values of m and n .

The fourth line contains m space-separated integers denoting the respective distances that each apple falls from point a .

The fifth line contains n space-separated integers denoting the respective distances that each orange falls from point b .

Constraints

- $1 \leq s, t, a, b, m, n \leq 10^5$
- $-10^5 \leq d \leq 10^5$
- $a < s < t < b$

Output Format

Print two integers on two different lines:

1. The first integer: the number of apples that fall on Sam's house.
2. The second integer: the number of oranges that fall on Sam's house.

Sample Input 0

```
7 11
5 15
3 2
-2 2 1
5 -6
```

Sample Output 0

```
1
1
```

Explanation 0

The first apple falls at position $5 - 2 = 3$.

The second apple falls at position $5 + 2 = 7$.

The third apple falls at position $5 + 1 = 6$.

The first orange falls at position $15 + 5 = 20$.

The second orange falls at position $15 - 6 = 9$.

Only one fruit (the second apple) falls within the region between **7** and **11**, so we print **1** as our first line of output.

Only the second orange falls within the region between **7** and **11**, so we print **1** as our second line of output.

[Change Theme](#)

Language

Python 3



```
1  #!/bin/python3
2
3  import math
4  import os
5  import random
6  import re
7  import sys
8
9  #
10 # Complete the 'countApplesAndOranges' function below.
11 #
12 # The function accepts following parameters:
13 # 1. INTEGER s
14 # 2. INTEGER t
15 # 3. INTEGER a
16 # 4. INTEGER b
17 # 5. INTEGER_ARRAY apples
18 # 6. INTEGER_ARRAY oranges
19 #
20
21 def countApplesAndOranges(s, t, a, b, apples, oranges):
22     # Write your code here
23     apples_distance = [i + a for i in apples]
24     orange_distance = [i + b for i in oranges]
25     cnt = 0
26     for i in apples_distance:
```

```
27         if s <= i <= t:
28             cnt += 1
29     print(cnt)
30     cnt = 0
31     for i in orange_distance:
32         if s <= i <= t:
33             cnt += 1
34     print(cnt)
```

EMACS

Line: 60 Col: 1

[Upload Code as File](#)☐ Test against custom input[Run Code](#)[Submit Code](#)

Running Testcases

✔ Test case 0 ⚙ Test case 4 ⚙ Test case 8

✔ Test case 1 ✔ Test case 5 ✔ Test case 9

✔ Test case 2 ⚙ Test case 6 ⚙ Test case 10

⚙ Test case 3 ✔ Test case 7 ✔ Test case 11

[Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Helpdesk](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#)