Triple sum ☆

Your Triple sum submission got 40.00 points. Share Tweet

Try the next challenge

Problem Submissions Leaderboard Editorial

Given 3 arrays $a,\ b,\ c$ of different sizes, find the number of distinct triplets $(p,\ q,\ r)$ where p is an element of a, written as $p \in a,\ \ q \in b$, and $\ r \in c$, satisfying the criteria: $p \leq q\ and\ q \geq r$.

For example, given $a = [3,5,7], b = [3,6]$, and $c = [4,6,9]$, we find four distinct triplets: $(3,6,4),(3,6,6),(5,6,4),(5,6,6)$.

Function Description

Complete the triplets function in the editor below. It must return the number of distinct triplets that can be formed from the given arrays.

triplets has the following parameter(s):

- a, b, c: three arrays of integers .

Input Format

The first line contains 3 integers $lena,\ lenb,\ and\ lenc$, the sizes of the three arrays.
The next 3 lines contain space-separated integers numbering $lena,\ lenb,\ and\ lenc$ respectively.

Constraints

$1 \leq lena, lenb, lenc \leq 10^5$

$1 \leq$ all elements in $a, b, c \leq 10^8$

Output Format

Print an integer representing the number of distinct triplets.

Sample Input 0

```
3 2 3
1 3 5
2 3
1 2 3
```

Sample Output 0

```
8
```

Explanation 0

The special triplets are $(1,2,1),(1,2,2),(1,3,1),(1,3,2),(1,3,3),(3,3,1),(3,3,2),(3,3,3)$.

Sample Input 1

```
3 3 3
1 4 5
2 3 3
1 2 3
```

Sample Output 1

```
5
```

Explanation 1

The special triplets are $(1,2,1),(1,2,2),(1,3,1),(1,3,2),(1,3,3)$

Sample Input 2

```
4 3 4
1 3 5 7
5 7 9
7 9 11 13
```

Sample Output 2

```
12
```

Explanation 2

The special triplets are $(1,7,7),(1,9,7),(1,9,9),(3,7,7),(3,9,7),(3,9,9),(5,7,7),(5,9,7),(5,9,9),(7,7,7),(7,9,7),(7,9,9)$.

Change Theme C++

```
// Complete the triplets function below.
long triplets(vector<int> a, vector<int> b, vector<int> c) {
    sort(a.begin(), a.end());
    sort(b.begin(), b.end());
    sort(c.begin(), c.end());

    a.erase(unique(a.begin(),a.end()),a.end()); //removing duplicates
    b.erase(unique(b.begin(),b.end()),b.end());
    c.erase(unique(c.begin(),c.end()),c.end());

    int i = 0, j = 0, k = 0;
    long count=0;

    while(i < b.size()){
        while(a[j]<=b[i] && j<a.size())   // lopping through a and checking number<= b[i]
            j++;
        while(c[k]<=b[i] && k<c.size()) // lopping through c and checking number<= b[i]
            k++;
        count+= (long)j*k;
        i++;
    }
```

Line: 21 Col: 90

Upload Code as File ☐ Test against custom input

Run Code Submit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

✓ Test case 0

✓ Test case 1

✓ Test case 2 🔒

✓ Test case 3 🔒

✓ Test case 4 🔒

✓ Test case 5 🔒

✓ Test case 6 🔒

Compiler Message

```
Success
```

Input (stdin) Download

```
3 2 3
1 3 5
2 3
1 2 3
```

Expected Output Download

```
8
```

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature