Triple sum ★

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 JavaScript (Node.js)

```
21  });
22
23  function readLine() {
24      return inputString[currentLine++];
25  }
26
27  // Complete the triplets function below.
28  function triplets(a, b, c) {
29      const sortedUnique = (value, index, self) => !(index>0 && value == self[index-1])
30
31      let cnt = 0
32      const as = a.sort((a, b) => a - b).filter(sortedUnique)
33      const bs = b.sort((a, b) => a - b).filter(sortedUnique)
34      const cs = c.sort((a, b) => a - b).filter(sortedUnique)
35      let ai=0, bi=0, ci=0;
36      while(bi<bs.length) {
37          while(ai < as.length && as[ai] <= bs[bi]) { ai++ }
38          while(ci < cs.length && cs[ci] <= bs[bi]) { ci++ }
39          cnt = cnt + ci * ai
40          bi++
41      }
42      return cnt
43  }
44
```

Line: 42 Col: 15

Upload Code as File Test against custom input Run Code Submit Code

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Sample Test case 1

Sample Test case 2

Input (stdin) Download

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

Your Output (stdout)

```
8
```

Expected Output Download

```
1 8
```