

Compare the Triplets

Alice and Bob each created one problem for HackerRank. A reviewer rates the two challenges, awarding points on a scale from 1 to 100 for three categories: *problem clarity*, *originality*, and *difficulty*.

The rating for Alice's challenge is the triplet $a = (a[0], a[1], a[2])$, and the rating for Bob's challenge is the triplet $b = (b[0], b[1], b[2])$.

The task is to calculate their comparison points by comparing each category:

- If $a[i] > b[i]$, then Alice is awarded 1 point.
- If $a[i] < b[i]$, then Bob is awarded 1 point.
- If $a[i] = b[i]$, then neither person receives a point.

Example

$a = [1, 2, 3]$

$b = [3, 2, 1]$

- For elements $*0*$, Bob is awarded a point because $a[0] < b[0]$.
- For the equal elements $a[1]$ and $b[1]$, no points are earned.
- Finally, for elements 2 , $a[2] > b[2]$ so Alice receives a point.

The return array is $[1, 1]$ with Alice's score first and Bob's second.

Function Description

Complete the function `compareTriplets` with the following parameter(s):

- `int a[3]`: Alice's challenge rating
- `int b[3]`: Bob's challenge rating

Returns

- `int[2]`: the first element is Alice's score and the second is Bob's score

Input Format

The first line contains 3 space-separated integers, $a[0]$, $a[1]$, and $a[2]$, the respective values in triplet a . The second line contains 3 space-separated integers, $b[0]$, $b[1]$, and $b[2]$, the respective values in triplet b .

Constraints

- $1 \leq a[i] \leq 100$
- $1 \leq b[i] \leq 100$

Sample Input 0

```
5 6 7
3 6 10
```

Sample Output 0

```
1 1
```

Explanation 0

In this example:

- $a = (a[0], a[1], a[2]) = (5, 6, 7)$
- $b = (b[0], b[1], b[2]) = (3, 6, 10)$

Now, let's compare each individual score:

- $a[0] > b[0]$, so Alice receives **1** point.
- $a[1] = b[1]$, so nobody receives a point.
- $a[2] < b[2]$, so Bob receives **1** point.

Alice's comparison score is **1**, and Bob's comparison score is **1**. Thus, we return the array **[1, 1]**.

Sample Input 1

```
17 28 30
99 16 8
```

Sample Output 1

```
2 1
```

Explanation 1

Comparing the **0th** elements, **17 < 99** so Bob receives a point.

Comparing the **1st** and **2nd** elements, **28 > 16** and **30 > 8** so Alice receives two points.

The return array is **[2, 1]**.