# 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 find their comparison points by comparing a[0] with b[0], a[1] with b[1], and a[2] with b[2].

* 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.

Comparison points is the total points a person earned.

Given a and b, determine their respective comparison points.

**Example**

a = [1, 2, 3]  
b = [3, 2, 1]

* For elements \*0\*, Bob is awarded a point because a[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*in the editor below.*

*compareTriplets has the following parameter(s):*

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

***Return***

* int[2]: Alice's score is in the first position, and Bob's score is in the second.

**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 ≤ a[i] ≤ 100
* 1 ≤ b[i] ≤ 100

**Sample Input 0**

5 6 7

3 6 10

**Sample Output 0**

1 1

**Explanation 0**

In this example:

Now, let's compare each individual score:

* , so Alice receives  point.
* , so nobody receives a point.
* , so Bob receives  point.

Alice's comparison score is , and Bob's comparison score is . Thus, we return the array .

**Sample Input 1**

17 28 30

99 16 8

**Sample Output 1**

2 1

**Explanation 1**

Comparing the  elements,  so Bob receives a point.  
Comparing the  and  elements,  and  so Alice receives two points.  
The return array is .

# A Very Big Sum

In this challenge, you are required to calculate and print the sum of the elements in an array, keeping in mind that some of those integers may be quite large.

In this challenge, you are required to calculate and print the sum of the elements in an array, keeping in mind that some of those integers may be quite large.

**Function Description**

Complete the aVeryBigSum function in the editor below. It must return the sum of all array elements.

aVeryBigSum has the following parameter(s):

* int ar[n]: an array of integers .

**Return**

* int: the sum of all array elements

**Input Format**

The first line of the input consists of an integer .  
The next line contains  space-separated integers contained in the array.

**Output Format**

Return the integer sum of the elements in the array.

**Constraints**

**Sample Input**

5

1000000001 1000000002 1000000003 1000000004 1000000005

**Output**

5000000015

**Note:**

The range of the 32-bit integer is .

When we add several integer values, the resulting sum might exceed the above range. You might need to use long int C/C++/Java to store such sums.

Change Theme

# Diagonal Difference

Given a square matrix, calculate the absolute difference between the sums of its diagonals.

For example, the square matrix  is shown below:

1 2 3

4 5 6

9 8 9

The left-to-right diagonal = . The right to left diagonal = . Their absolute difference is .

1 2 3

4 5 6

9 8 9

The left-to-right diagonal = . The right to left diagonal = . Their absolute difference is .

**Function description**

Complete the  function in the editor below.

diagonalDifference takes the following parameter:

* int arr[n][m]: an array of integers

**Return**

* int: the absolute diagonal difference

**Input Format**

The first line contains a single integer, , the number of rows and columns in the square matrix .  
Each of the next  lines describes a row, , and consists of  space-separated integers .

**Constraints**

**Output Format**

Return the absolute difference between the sums of the matrix's two diagonals as a single integer.

**Sample Input**

3

11 2 4

4 5 6

10 8 -12

**Sample Output**

15

**Explanation**

The primary diagonal is:

11

5

-12

Sum across the primary diagonal: 11 + 5 - 12 = 4

The secondary diagonal is:

4

5

10

Sum across the secondary diagonal: 4 + 5 + 10 = 19  
Difference: |4 - 19| = 15

**Note:** |x| is the [absolute value](https://www.mathsisfun.com/numbers/absolute-value.html) of x