CHANDIGARH UNIVERSITY

UNIVERSITY INSTITUTE OF NGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Submitted By: Vivek Kumar(21BCS8129)	**Submitted To:** Mamta Punia(E12337)
Subject Name	Competitive Coding - I
Subject Code	20CSP-314
Branch	Computer Science and Engineering
Semester	5th

Experiment - 1

Student Name: Vivek Kumar **UID: 21BCS8129**
Branch: BE-CSE(LEET) **Section/Group: WM-20BCS-616/A**
Semester: 5th **Date of Performance: 12/08/2022**
Subject Name: Competitive coding - I **Subject Code: 20CSP-314**

1. Aim/Overview of the practical:
I. Given an array of integers, find the sum of its elements.
For example, if the array arr=[1,2,3], 1+2+3=6 , so return 6.

II. 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.

2. Task to be done/ Which logistics used:
Make the Changes in the present code and find the Actual output of the given question.

3. Algorithm/Flowchart (For programming-based labs)**:**

4. Steps for experiment/practical/Code:
I. Simple Array Sum:

```
import java.io.*;
import java.util.*;


public class Solution {
    public static int simpleArraySum(int n, int[] ar) {
```

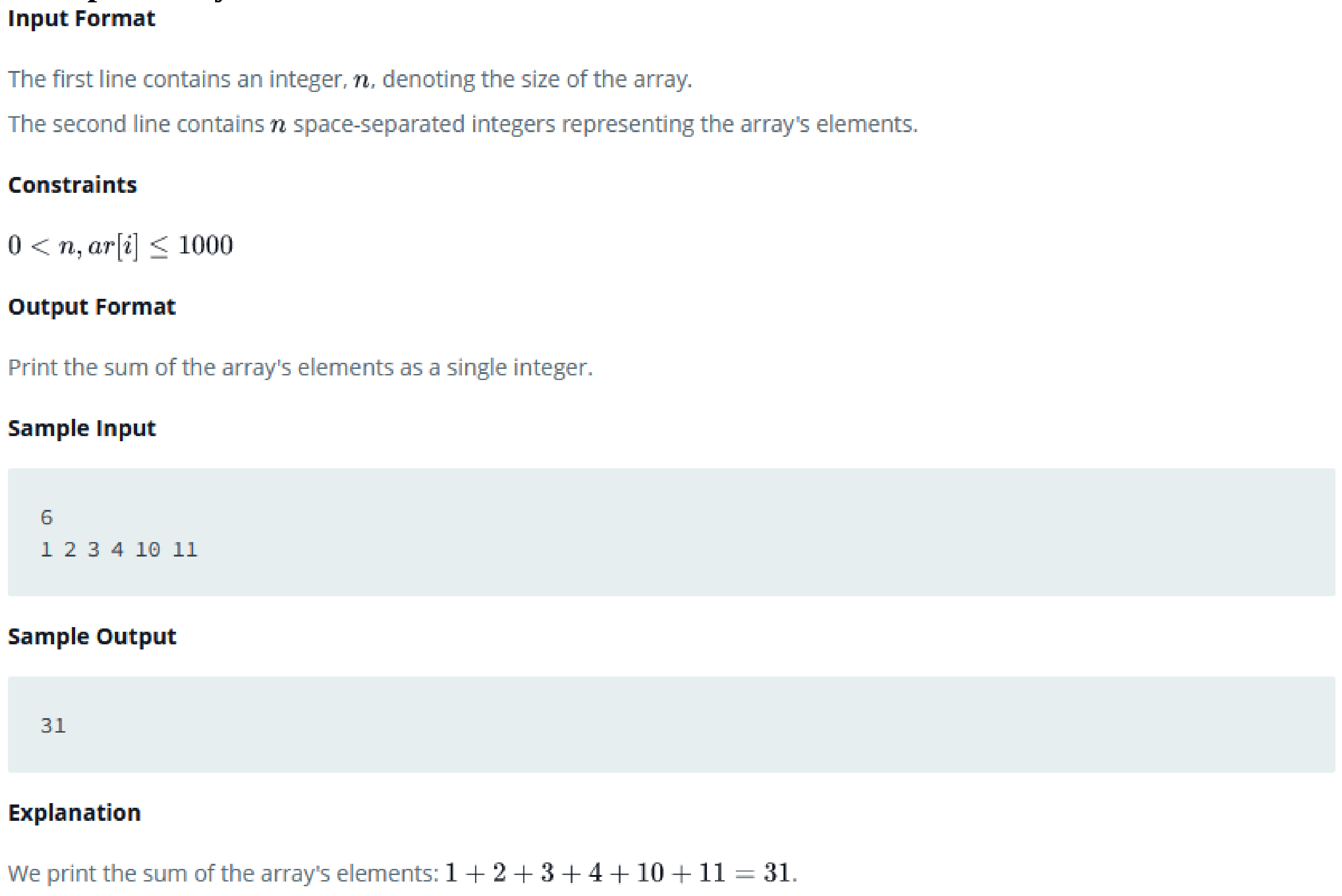
```
    // Write your code here
        int sum=0;
        for(int i=0;i<ar.length;i++){
            sum=sum+ar[i];
        }
        return sum;
    }

    public static void main(String[] args) throws IOException {
        Scanner in = new Scanner(System.in);
           int n = in.nextInt();
           int[] arr = new int[n];

           for(int i=0; i < n; i++){
               arr[i] = in.nextInt();
               }
               in.close();
             int sum=simpleArraySum(n, arr);
             System.out.print(sum);
               }
    }
```

II. Compare the Triplets:

```
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.regex.*;

class Result {

    /*
     * Complete the 'compareTriplets' function below.
     *
     * The function is expected to return an INTEGER_ARRAY.
     * The function accepts following parameters:
     *  1. INTEGER_ARRAY a
     *  2. INTEGER_ARRAY b
     */

    public static List<Integer> compareTriplets(List<Integer> a, List<Integer> b) {
    // Write your code here
        int alice = 0;
        int bob = 0;
        List<Integer> answer = new ArrayList<>();
        for(int i = 0; i < 3; i++) {
            if (a.get(i) > b.get(i)) alice++;
            if (a.get(i) < b.get(i)) bob++;
        }
        answer.add(0,alice);
        answer.add(1,bob);
        return answer;
    }

}

public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(System.in));
        BufferedWriter bufferedWriter = new BufferedWriter(new
FileWriter(System.getenv("OUTPUT_PATH")));

        String[] aTemp = bufferedReader.readLine().replaceAll("\\s+$", "").split(" ");

        List<Integer> a = new ArrayList<>();
        for (int i = 0; i < 3; i++) {
            int aItem = Integer.parseInt(aTemp[i]);
            a.add(aItem);
        }

        String[] bTemp = bufferedReader.readLine().replaceAll("\\s+$", "").split(" ");

        List<Integer> b = new ArrayList<>();

        for (int i = 0; i < 3; i++) {
            int bItem = Integer.parseInt(bTemp[i]);
            b.add(bItem);
        }

        List<Integer> result = Result.compareTriplets(a, b);

        for (int i = 0; i < result.size(); i++) {
            bufferedWriter.write(String.valueOf(result.get(i)));

            if (i != result.size() - 1) {
                bufferedWriter.write(" ");
            }
        }

        bufferedWriter.newLine();

        bufferedReader.close();
        bufferedWriter.close();
    }
}
```

5. Observations/Discussions/ Complexity Analysis:

I. Simple Array Sum:



Input Format

The first line contains an integer, n, denoting the size of the array.

The second line contains n space-separated integers representing the array's elements.

Constraints

$0 < n, ar[i] \leq 1000$

Output Format

Print the sum of the array's elements as a single integer.

Sample Input

```
6
1 2 3 4 10 11
```

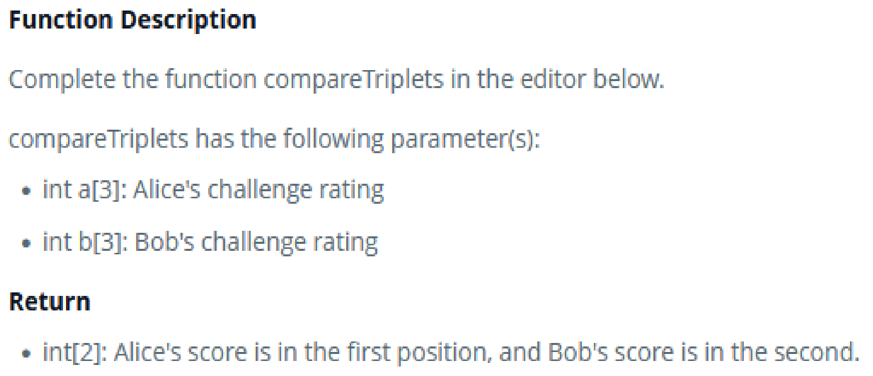
Sample Output

```
31
```

Explanation

We print the sum of the array's elements: $1 + 2 + 3 + 4 + 10 + 11 = 31$.

II. Compare the Triplets:



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:

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

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

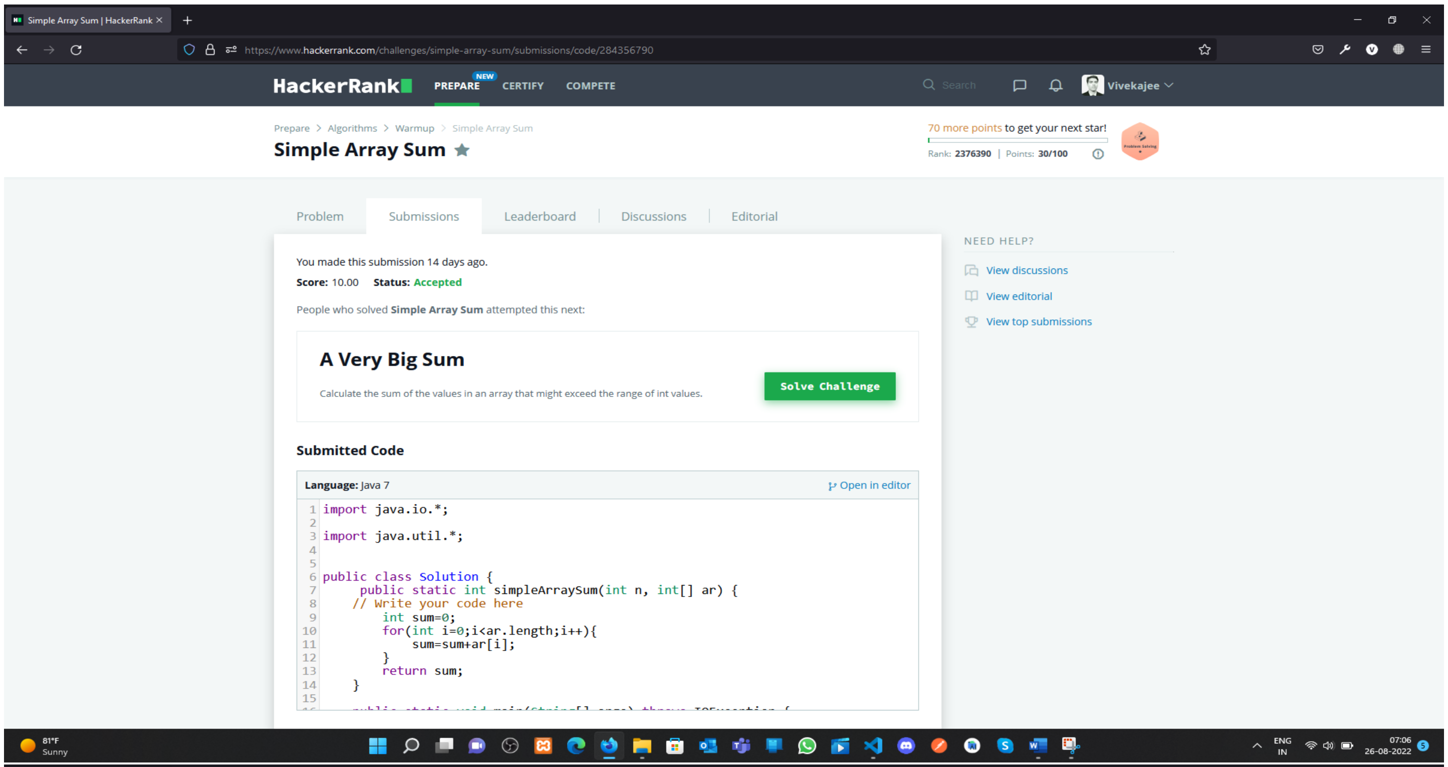
Comparing the 0^{th} elements, $17 < 99$ so Bob receives a point.

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

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

6. Result/Output/Writing Summary:

I. Simple Array Sum:


Simple Array Sum | HackerRank
https://www.hackerrank.com/challenges/simple-array-sum/submissions/code/284356790
HackerRank
PREPARE
NEW
CERTIFY
COMPETE
Search
Vivekajee
Prepare > Algorithms > Warmup > Simple Array Sum
70 more points to get your next star!
Rank: 2376390 | Points: 30/100
Problem
Submissions
Leaderboard
Discussions
Editorial
NEED HELP?
View discussions
View editorial
View top submissions
Solve Challenge
Open in editor
81°F
Sunny
ENG
IN
07:06
26-08-2022

Simple Array Sum

You made this submission 14 days ago.

Score: 10.00 Status: Accepted

People who solved Simple Array Sum attempted this next:

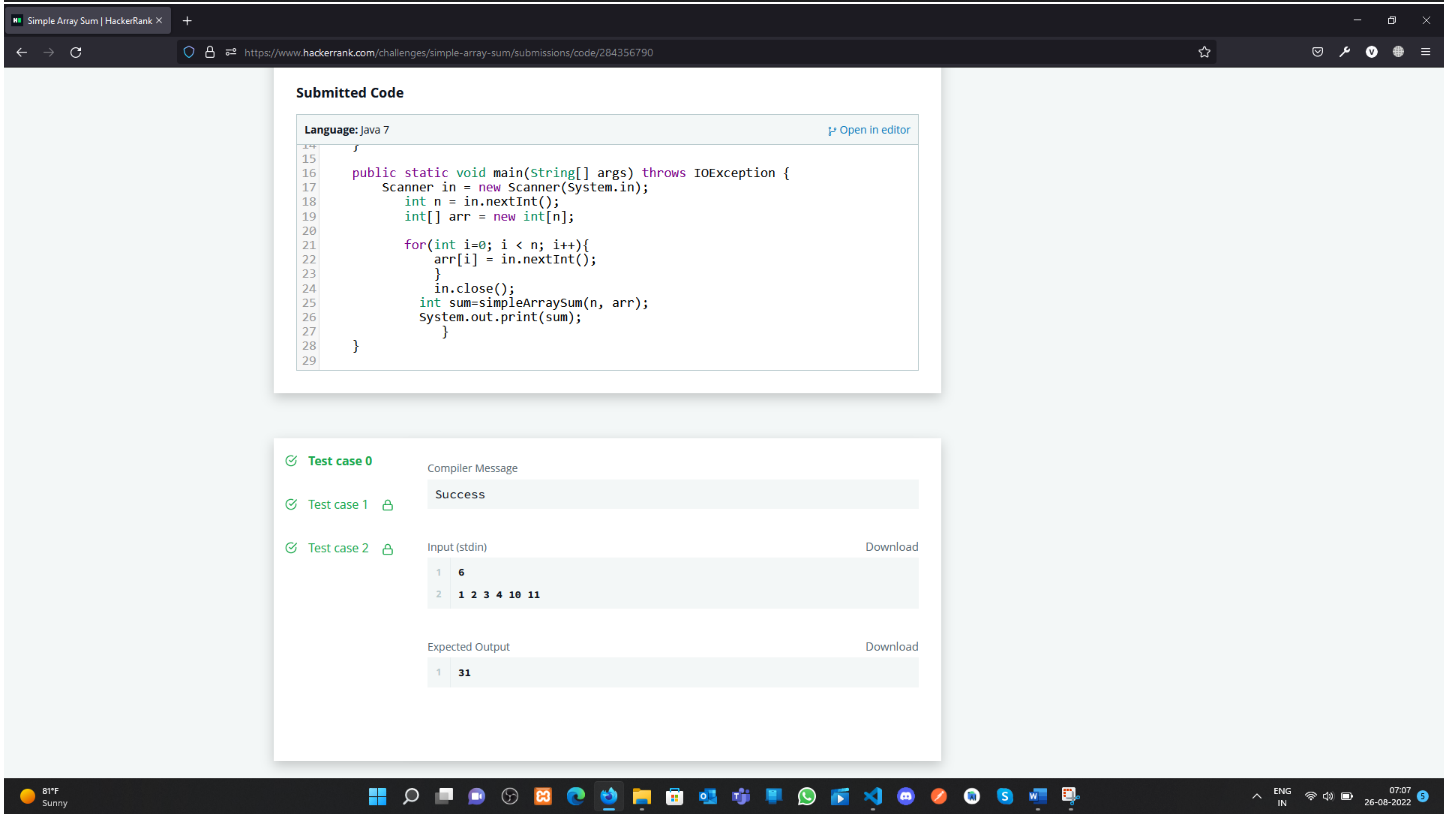
A Very Big Sum

Calculate the sum of the values in an array that might exceed the range of int values.

Submitted Code

Language: Java 7

```
import java.io.*;

import java.util.*;


public class Solution {
     public static int simpleArraySum(int n, int[] ar) {
    // Write your code here
        int sum=0;
        for(int i=0;i<ar.length;i++){
            sum=sum+ar[i];
        }
        return sum;
    }

```


Simple Array Sum | HackerRank
https://www.hackerrank.com/challenges/simple-array-sum/submissions/code/284356790
Open in editor
Download
Download
81°F
Sunny
ENG
IN
07:07
26-08-2022

Submitted Code

Language: Java 7

```

    public static void main(String[] args) throws IOException {
        Scanner in = new Scanner(System.in);
           int n = in.nextInt();
           int[] arr = new int[n];

           for(int i=0; i < n; i++){
               arr[i] = in.nextInt();
               }
               in.close();
             int sum=simpleArraySum(n, arr);
             System.out.print(sum);
                }
    }

```

Test case 0

Test case 1

Test case 2

Compiler Message

```
Success
```

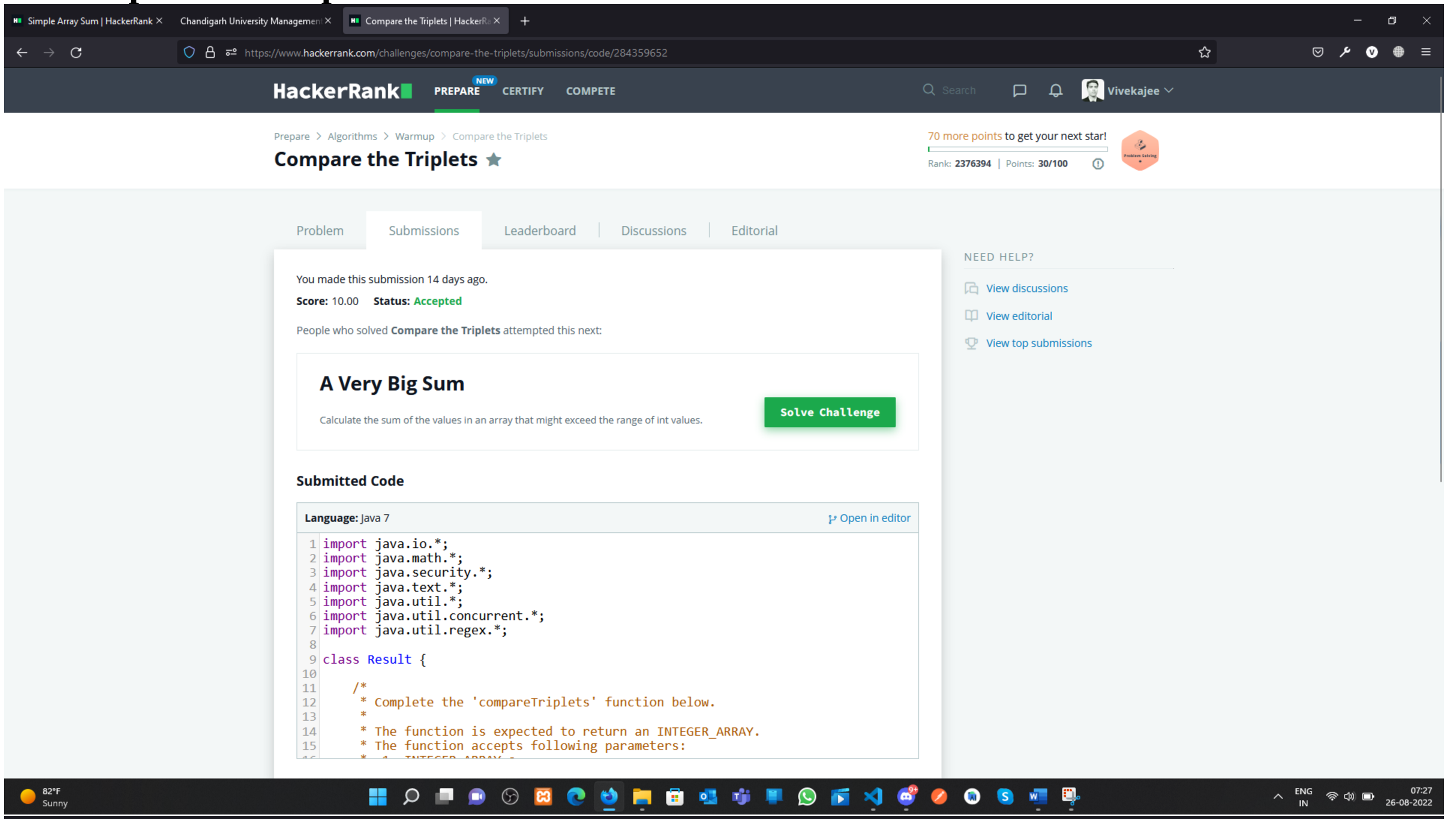
Input (stdin)

```
6
1 2 3 4 10 11
```

Expected Output

```
31
```

II. Compare the Triplets:



Prepare > Algorithms > Warmup > Compare the Triplets

Compare the Triplets

You made this submission 14 days ago.
Score: 10.00 **Status:** Accepted

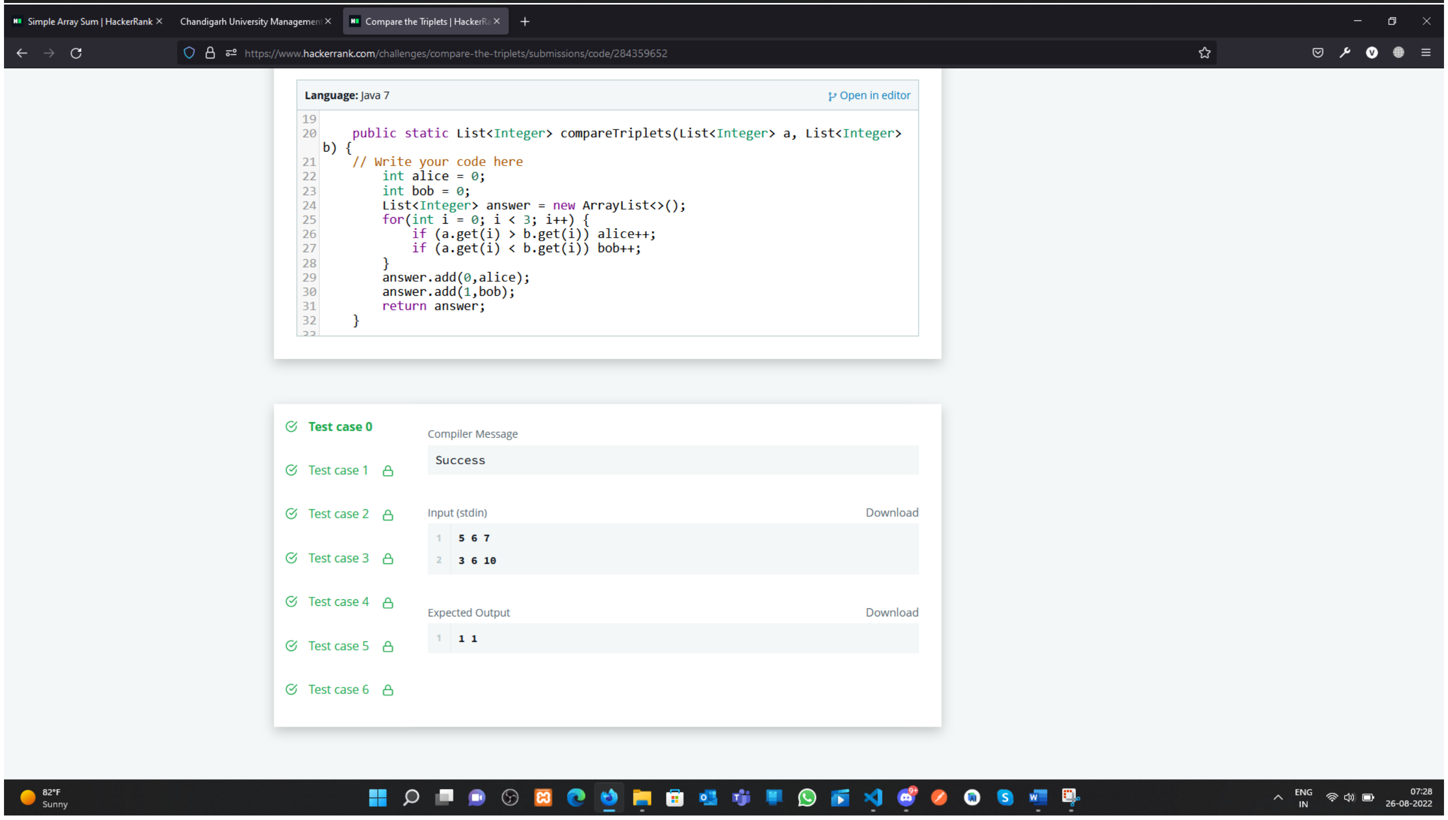
People who solved **Compare the Triplets** attempted this next:

A Very Big Sum
Calculate the sum of the values in an array that might exceed the range of int values.

Submitted Code

Language: Java 7

```java
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.regex.*;

class Result {

    /*
     * Complete the 'compareTriplets' function below.
     *
     * The function is expected to return an INTEGER_ARRAY.
     * The function accepts following parameters:
```



Language: Java 7

```java

    public static List<Integer> compareTriplets(List<Integer> a, List<Integer>
b) {
    // Write your code here
        int alice = 0;
        int bob = 0;
        List<Integer> answer = new ArrayList<>();
        for(int i = 0; i < 3; i++) {
            if (a.get(i) > b.get(i)) alice++;
            if (a.get(i) < b.get(i)) bob++;
        }
        answer.add(0,alice);
        answer.add(1,bob);
        return answer;
    }
```

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

Expected Output
```
1 1
```

Learning outcomes (What I have learnt):

1. Array concept in Java
2. Sum of the all-item present in an Array
3. Compare the triplets and show the results.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			