

**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	5 th		

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

Simple Array Sum:

1. Aim/Overview of the practical:

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.

2. Task to be done/ Which logistics used:

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: $1 + 2 + 3 + 4 + 10 + 11 = 31$.

3. Hardware and Software Requirements (For programming-based labs):

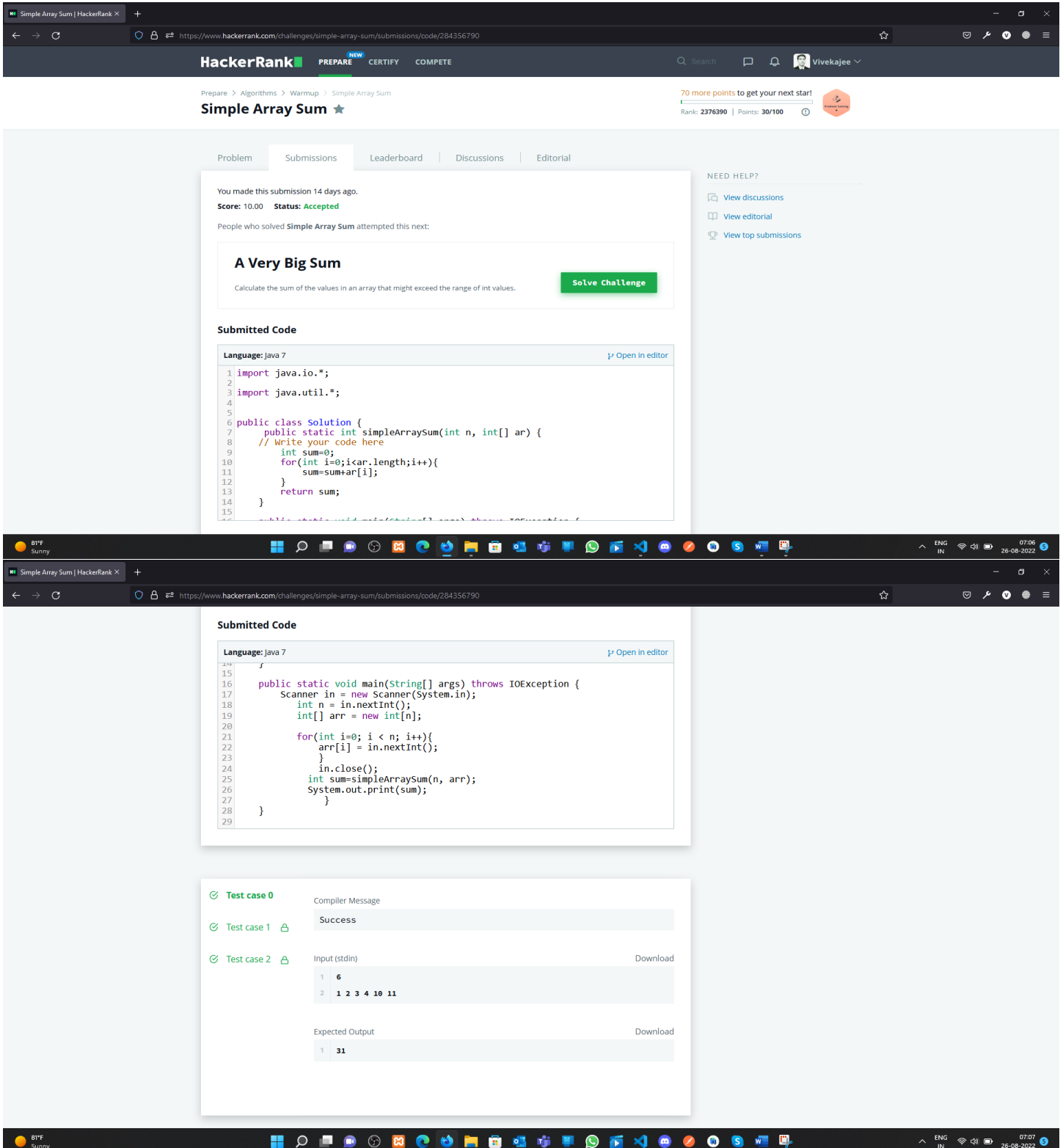
- Laptop or Desktop
- Hacker-Rank Account

4. Steps for experiment/practical/Code:

```
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);  
    }  
}
```

5. Result/Output/Writing Summary:



The screenshot displays the HackerRank submission page for the 'Simple Array Sum' challenge. The user's submission is accepted with a score of 10.00. The page shows the problem description, the submitted code in Java, and the test results for three test cases.

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

Submitted Code (Java 7):

```
1 import java.io.*;
2
3 import java.util.*;
4
5
6 public class Solution {
7     public static int simpleArraySum(int n, int[] ar) {
8         // Write your code here
9         int sum=0;
10        for(int i=0;i<ar.length;i++){
11            sum=sum+ar[i];
12        }
13        return sum;
14    }
15
16    public static void main(String[] args) throws IOException {
17        Scanner in = new Scanner(System.in);
18        int n = in.nextInt();
19        int[] arr = new int[n];
20
21        for(int i=0; i < n; i++){
22            arr[i] = in.nextInt();
23        }
24        in.close();
25        int sum=simpleArraySum(n, arr);
26        System.out.print(sum);
27    }
28
29 }
```

Test Results:

- Test case 0: Success
- Test case 1: Success
- Test case 2: Success

Input (stdin):

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

Expected Output:

```
31
```

Compare the Triplets:

1. Aim/Overview of the practical:

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.

2. Task to be done/ Which logistics used:

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

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

Sample Input 0

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

Sample Output 0

```
1 1
```

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].

3. Hardware and Software Requirements (For programming-based labs):

- Laptop or Desktop
- Hacker-Rank Account

4. Steps for experiment/practical/Code:

```
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);

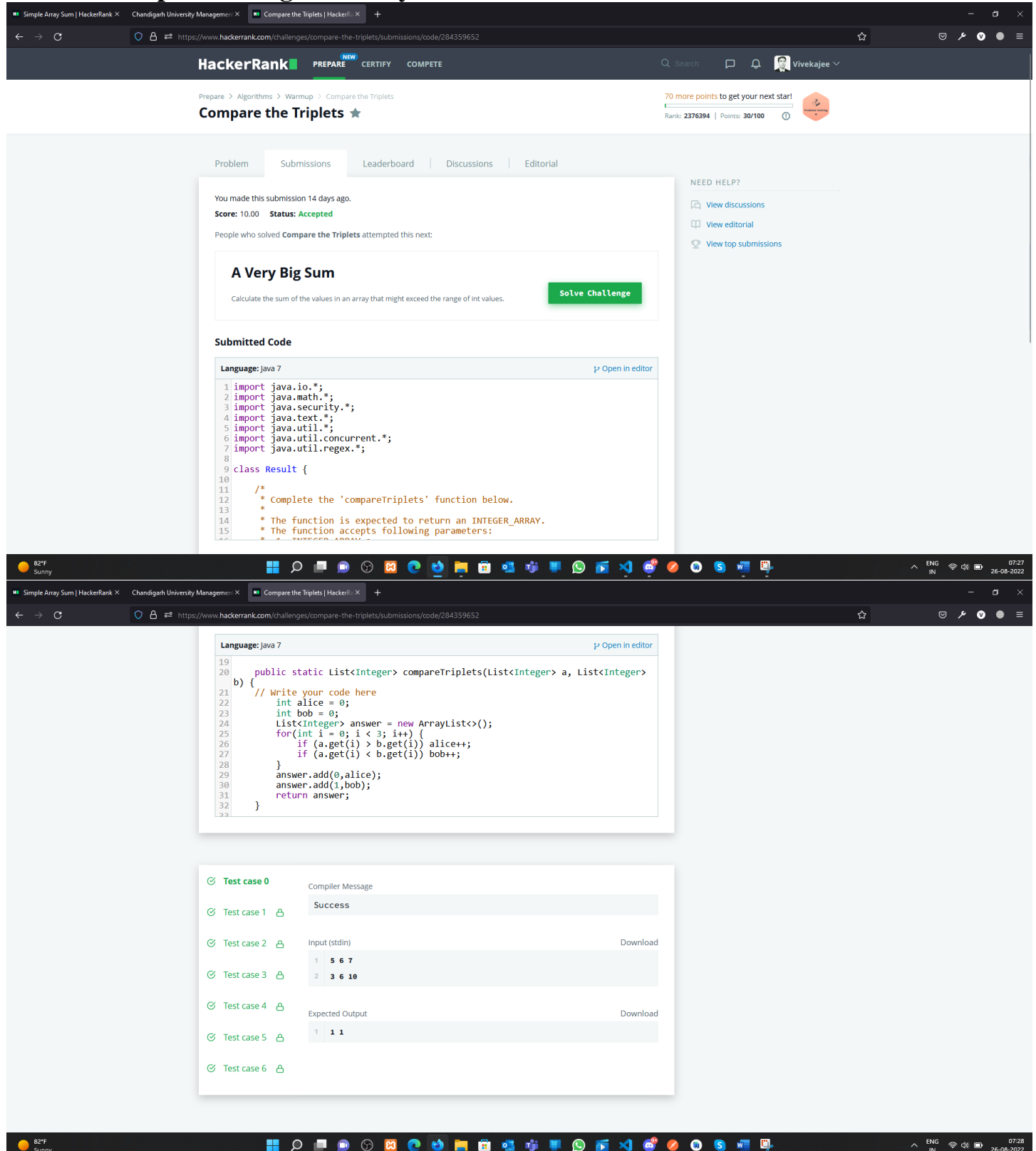
for (int i = 0; i < result.size(); i++) {
    bufferedWriter.write(String.valueOf(result.get(i)));

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

bufferedWriter.newLine();

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


5. Result/Output/Writing Summary:



HackerRank PREPARE NEW CERTIFY COMPETE

Prepare > Algorithms > Warmup > Compare the Triplets

Compare the Triplets ★

70 more points to get your next star!

Rank: 2376394 | Points: 30/100

NEED HELP?

- [View discussions](#)
- [View editorial](#)
- [View top submissions](#)

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.

[Solve Challenge](#)

Submitted Code

Language: Java 7 [Open in editor](#)

```

1 import java.io.*;
2 import java.math.*;
3 import java.security.*;
4 import java.text.*;
5 import java.util.*;
6 import java.util.concurrent.*;
7 import java.util.regex.*;
8
9 class Result {
10
11     /*
12      * Complete the 'compareTriplets' function below.
13      * The function is expected to return an INTEGER_ARRAY.
14      * The function accepts following parameters:
15      * 1. INTEGER_ARRAY a
16      * 2. INTEGER_ARRAY b
17      */
18
19     public static List<Integer> compareTriplets(List<Integer> a, List<Integer> b) {
20         // Write your code here
21         int alice = 0;
22         int bob = 0;
23         List<Integer> answer = new ArrayList<>();
24         for(int i = 0; i < 3; i++) {
25             if (a.get(i) > b.get(i)) alice++;
26             if (a.get(i) < b.get(i)) bob++;
27         }
28         answer.add(0,alice);
29         answer.add(1,bob);
30         return answer;
31     }
32 }

```

Test case 0 [Success](#)

Test case 1 [Success](#)

Test case 2 [Success](#)

Test case 3 [Success](#)

Test case 4 [Success](#)

Test case 5 [Success](#)

Test case 6 [Success](#)

Compiler Message

Success

Input (stdin) [Download](#)

```

1 5 6 7
2 3 6 10

```

Expected Output [Download](#)

```

1 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.			