**HACKERRANK PROBLEM SOLUTION DATE:13/10/2022:--**

Q:1)[**https://www.hackerrank.com/challenges/simple-array-sum/problem**](https://www.hackerrank.com/challenges/simple-array-sum/problem) **?**

**Given an array of integers, find the sum of its elements**

**For example, f the array ar =|1,2,3]14 2+3=6, so return 6.**

**Function Description**

**Complete the simpleArraySum function in the editor below. It must return the sum of the array elements a an integer.**

**simpleArraySum has the following parameter(s**

**ar an array of integers**

**Input Format**

**The first line contains an integer, 7n, denoting the ize of the array**

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

**Constraints**

**0n,arlis 1000**

**Output Format**

**Print the sum of the array's elements asa single integer.**

**Sample input**

**234 1e 11**

**Sample Output**

**Ans:1)import java.io.\*;**

**import java.math.\*;**

**import java.security.\*;**

**import java.text.\*;**

**import java.util.\*;**

**import java.util.concurrent.\*;**

**import java.util.function.\*;**

**import java.util.regex.\*;**

**import java.util.stream.\*;**

**import static java.util.stream.Collectors.joining;**

**import static java.util.stream.Collectors.toList;**

**class Result {**

**/\***

**\* Complete the 'simpleArraySum' function below.**

**\***

**\* The function is expected to return an INTEGER.**

**\* The function accepts INTEGER\_ARRAY ar as parameter.**

**\*/**

**public static int simpleArraySum(List<Integer> ar) {**

**// Write your code here**

**int sum=0;**

**for(Integer value:ar){**

**sum=sum+value;**

**}**

**return sum;**

**}**

**}**

**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")));**

**int arCount = Integer.parseInt(bufferedReader.readLine().trim());**

**List<Integer> ar = Stream.of(bufferedReader.readLine().replaceAll("\\s+$", "").split(" "))**

**.map(Integer::parseInt)**

**.collect(toList());**

**int result = Result.simpleArraySum(ar);**

**bufferedWriter.write(String.valueOf(result));**

**bufferedWriter.newLine();**

**bufferedReader.close();**

**bufferedWriter.close();**

**}**

**}**

**Q:2)**[**https://www.hackerrank.com/challenges/a-very-big-sum/problem**](https://www.hackerrank.com/challenges/a-very-big-sum/problem) **?**

**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**

**long: the sum of all array elements**

**Input Format**

**The first line of the input consists of an integer n.**

**The next line contains n space-separated integers contained in the array.**

**Output Format**

**Return the integer sum of the elements in the array.**

**Constraints**

**1<n< 10**

**0 < arli]s 1040**

**Sample Input**

**10090000e 1 1000000002 1000000093 1090000004 100000ee05**

**Ans:2) import java.io.\*;**

**import java.math.\*;**

**import java.security.\*;**

**import java.text.\*;**

**import java.util.\*;**

**import java.util.concurrent.\*;**

**import java.util.function.\*;**

**import java.util.regex.\*;**

**import java.util.stream.\*;**

**import static java.util.stream.Collectors.joining;**

**import static java.util.stream.Collectors.toList;**

**class Result {**

**/\***

**\* Complete the 'aVeryBigSum' function below.**

**\***

**\* The function is expected to return a LONG\_INTEGER.**

**\* The function accepts LONG\_INTEGER\_ARRAY ar as parameter.**

**\*/**

**public static long aVeryBigSum(List<Long> ar) {**

**// Write your code here**

**long sum=0;**

**for(Long value:ar)**

**{**

**sum=sum+value;**

**}**

**return sum;**

**}**

**}**

**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")));**

**int arCount = Integer.parseInt(bufferedReader.readLine().trim());**

**List<Long> ar = Stream.of(bufferedReader.readLine().replaceAll("\\s+$", "").split(" "))**

**.map(Long::parseLong)**

**.collect(toList());**

**long result = Result.aVeryBigSum(ar);**

**bufferedWriter.write(String.valueOf(result));**

**bufferedWriter.newLine();**

**bufferedReader.close();**

**bufferedWriter.close();**

**}**

**}**

**Q:3)**[**https://www.hackerrank.com/challenges/compare-the-triplets/problem**](https://www.hackerrank.com/challenges/compare-the-triplets/problem) **?**

**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[01, a[1], a[2]), and the rating for Bob's challenge is the triplet b = (b[O], 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 all>b[i), then Alice is awarded 1 point.**

**If all< b[i), then Bob is awarded 1 point.**

**If atl=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= [B, 2,1]**

**For elements \*0\*, Bob is awarded a point because a[0].**

**Ifal <b[D then Bob is awarded 1 point**

**Ifal-bl 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[] 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 compare Triplets in the editor below.**

**compareTriplers has the following parameters**

**int a3 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], a1], and a[2], the respective values in triplet a**

**The second line contains 3 space-separated integers, b[O], b[1]. and b[2], the respective values in triplet b.**

**Constraints**

**1sals 100**

**1s Ds 100**

**Ans:3) import java.io.\*;**

**import java.math.\*;**

**import java.security.\*;**

**import java.text.\*;**

**import java.util.\*;**

**import java.util.concurrent.\*;**

**import java.util.function.\*;**

**import java.util.regex.\*;**

**import java.util.stream.\*;**

**import static java.util.stream.Collectors.joining;**

**import static java.util.stream.Collectors.toList;**

**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) {**

**int alice=0;**

**int bob=0;**

**List<Integer>category=new ArrayList<Integer>();**

**for(int i=0;i<3;i++){**

**if(a.get(i)>b.get(i))**

**alice++;**

**if(a.get(i)<b.get(i))**

**bob++;**

**}**

**category.add(alice);**

**category.add(bob);**

**return category;**

**}**

**}**

**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")));**

**List<Integer> a = Stream.of(bufferedReader.readLine().replaceAll("\\s+$", "").split(" "))**

**.map(Integer::parseInt)**

**.collect(toList());**

**List<Integer> b = Stream.of(bufferedReader.readLine().replaceAll("\\s+$", "").split(" "))**

**.map(Integer::parseInt)**

**.collect(toList());**

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

**bufferedWriter.write(**

**result.stream()**

**.map(Object::toString)**

**.collect(joining(" "))**

**+ "\n"**

**);**

**bufferedReader.close();**

**bufferedWriter.close();**

**}**

**}**