Mini-Max Sum ☆

9 more points to get your first star!

Rank: **1489833** | Points: **21/30**

Your Mini-Max Sum submission got 10.00 points. Share Tweet

You are now 9 points away from the 1st star for your problem solving badge.

Try the next challenge | **Try a Random Challenge**

Problem Submissions Leaderboard Editorial

Given five positive integers, find the minimum and maximum values that can be calculated by summing exactly four of the five integers. Then print the respective minimum and maximum values as a single line of two space-separated long integers.

For example, $arr = [1, 3, 5, 7, 9]$. Our minimum sum is $1 + 3 + 5 + 7 = 16$ and our maximum sum is $3 + 5 + 7 + 9 = 24$. We would print

```
16 24
```

Function Description

Complete the miniMaxSum function in the editor below. It should print two space-separated integers on one line: the minimum sum and the maximum sum of 4 of 5 elements.

miniMaxSum has the following parameter(s):

- arr: an array of 5 integers

Input Format

A single line of five space-separated integers.

Constraints

$1 \leq arr[i] \leq 10^9$

Output Format

Print two space-separated long integers denoting the respective minimum and maximum values that can be calculated by summing exactly four of the five integers. (The output can be greater than a 32 bit integer.)

Sample Input

```
1 2 3 4 5
```

Sample Output

```
10 14
```

Explanation

Our initial numbers are 1, 2, 3, 4, and 5. We can calculate the following sums using four of the five integers:

1. If we sum everything except 1, our sum is $2+3+4+5=14$.
2. If we sum everything except 2, our sum is $1+3+4+5=13$.
3. If we sum everything except 3, our sum is $1+2+4+5=12$.
4. If we sum everything except 4, our sum is $1+2+3+5=11$.
5. If we sum everything except 5, our sum is $1+2+3+4=10$.

Hints: Beware of integer overflow! Use 64-bit Integer.

Need help to get started? Try the Solve Me First problem

Change Theme Java 8

```
1  import java.util.*;
2
3  public class Solution {
4
5      public static void main(String[] args) {
6          Scanner scan = new Scanner(System.in);
7          long sum = 0;
8          long min = 1000000000;
9          long max = 0;
10         while(scan.hasNext()) {
11             long n = scan.nextLong();
12             sum = sum + n;
13             if (min > n) {
14                 min = n;
15             }
16             if (max < n) {
17                 max = n;
18             }
19         }
20         scan.close();
21
22         System.out.println((sum - max) + " " + (sum - min));
23     }
24 }
25
```

Line: 24 Col: 2

Upload Code as File Test against custom input Run Code Submit Code

You have earned 10.00 points!

You are now 9 points away from the 1st star for your problem solving badge.

70% 21/30

Problem Solving

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

- 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) Download

```
1 1 2 3 4 5
```

Expected Output Download

```
1 10 14
```