A Very Big Sum ★

169 more points to get your next star!

Rank: **634398** | Points: **306/475**

You have successfully solved A Very Big Sum Share Tweet

You are now 169 points away from the 4th star for your problem solving badge.

Try the next challenge

Problem Submissions Leaderboard Editorial

RATE THIS CHALLENGE

★ ★ ★ ★ ★

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

$1 \leq n \leq 10$

$0 \leq ar[i] \leq 10^{10}$

Sample Input

```
5
1000000001 1000000002 1000000003 1000000004 1000000005
```

Output

```
5000000015
```

Note:

The range of the 32-bit integer is (-2^{31}) *to* $(2^{31}-1)$ *or* $[-2147483648, 2147483647]$.

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 Language Python 3

```python
#!/bin/python3

import math
import os
import random
import re
import sys

# Complete the aVeryBigSum function below.
def aVeryBigSum(ar):
    return sum(ar)

if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')

    ar_count = int(input())

    ar = list(map(int, input().rstrip().split()))

    result = aVeryBigSum(ar)

    fptr.write(str(result) + '\n')

    fptr.close()

```

Line: 22 Col: 5

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

Congratulations

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

Next Challenge

Test case 0

Test case 1

Compiler Message

```
Success
```

Input (stdin) Download

```
5
1000000001 1000000002 1000000003 1000000004 1000000005
```

Expected Output Download

```
5000000015
```