

A Very Big Sum **■**



Problem Submissions Leaderboard Discussions Editorial

You are given an array of integers of size N. You need to print the sum of the elements in the array, keeping in mind that some of those integers may be quite large.

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

Print a single value equal to the sum of the elements in the array.

Constraints

 $\begin{aligned} &1 \leq N \leq 10 \\ &0 \leq A[i] \leq 10^{10} \end{aligned}$

Sample Input

5 100000001 100000002 100000003 100000004 100000005

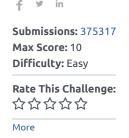
Output

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 long int in C/C++ or long data type in Java to store such sums.



Go

Need Help? Get advice from the discussion forum for this challenge. Or check out the environments page

Current Buffer (saved locally, editable) & 🗘

```
1
   package main
2
3
   import (
4
        "fmt"
5
6
7
8 v func main() {
10
        var index int
11
        fmt.Scanf("%d", &index)
12
13
        leftright := 0
14
        rightleft := 0
15
        leftrightT := 0
        rightleftT := 0
16
17 ▼
        for y := 0; y < index; y++ \{
18
19 ▼
            for x := 0; x < index; x++ \{
20
                dontcare := 0
21 ▼
                if x == y \{
                    fmt.Scanf("%d", &leftright)
22
                    leftrightT += leftright
23
24 ▼
                } else if x == index-y {
                     fmt.Scanf("%d", &rightleft)
25
26
                     rightleftT += rightleft
27 ▼
                } else {
28
                    fmt.Scanf("%d", &dontcare)
29
30
                fmt.Println(rightleft)
31
32
            }
33
34
35
        fmt.Print(leftrightT - rightleftT)
36
37
   }
38
39 ▼ /*
40
   func Getindex() int {
41
42
        return index
43
   }
44
45
    func CreateGrid(numRowsClmns int) int {
46
        grid := make([][]int, numRowsClmns)
47
48
   }
49
50
51 ▼ /*
   x numRowsClmns-x
52
53
54
    func AddGrid(numRowsClmns int, grid [][]int) int {
55
        leftRight := 0
56
        for x, y := 0, 0; x < numRowsClmns; x, y = x+1, y+1 {
57
            leftRight += grid[y][x]
58
59
        rightLeft := 0
60
        numRowsClmns = numRowsClmns - 1
61
        for x, y := numRowsClmns, 0; x \ge 0; x, y = x-1, y+1 {
62
            rightLeft += grid[y][x]
63
64
        finalNum := 0
        finalNum = leftRight-rightLeft
65
66
        return finalNum
67
   }
68
69
70
71
```

Line: 1 Col: 1

<u>Lupload Code as File</u> Test against custom input	Run Code Submit Code
Copyright © 2017 HackerRank. All Rights Reserved	

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature