A Very Big Sum

by vatsalchanana

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
$1 \leq N \leq 10$
$0 \leq A[i] \leq 10^{10}$

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

Submissions: 375317
Max Score: 10
Difficulty: Easy

Rate This Challenge:
☆☆☆☆☆

More

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

Current Buffer (saved locally, editable) Go

```go
package main

import (
    "fmt"
)


func main() {

    var index int
    fmt.Scanf("%d", &index)

    leftright := 0
    rightleft := 0
    leftrightT := 0
    rightleftT := 0
    for y := 0; y < index; y++ {

        for x := 0; x < index; x++ {
            dontcare := 0
            if x == y {
                fmt.Scanf("%d", &leftright)
                leftrightT += leftright
            } else if x == index-y {
                fmt.Scanf("%d", &rightleft)
                rightleftT += rightleft
            } else {
                fmt.Scanf("%d", &dontcare)
            }
            fmt.Println(rightleft)

        }

    }
    fmt.Print(leftrightT - rightleftT)

}

/*
func Getindex() int {

    return index
}

func CreateGrid(numRowsClmns int) int {
    grid := make([][]int, numRowsClmns)

}
*/

/*
x numRowsClmns-x

func AddGrid(numRowsClmns int, grid [][]int) int {
    leftRight := 0
    for x, y := 0, 0; x < numRowsClmns; x, y = x+1, y+1 {
        leftRight += grid[y][x]
    }
    rightLeft := 0
    numRowsClmns = numRowsClmns - 1
    for x, y := numRowsClmns, 0; x >= 0; x, y = x-1, y+1 {
        rightLeft += grid[y][x]
    }
    finalNum := 0
    finalNum = leftRight-rightLeft
    return finalNum

}

*/

```

Line: 1 Col: 1

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