

Práctica 1: Suma

Descripción del problema

En este problema de práctica, deberás implementar una función que reciba **N** números en un arreglo, y compute su suma.

Se ejercitan específicamente:

- Tipos de 64 bits vs 32 bits
- Cuidado adecuado para prevención de “overflow”
- Recibir entrada en un arreglo: iterar, saber su longitud
- Retornar resultado de la función
- Entender las subtareas, y usarlas para crear intencionalmente programas que apuntan a menos de 100 puntos

Recuerda que los tipos de 32 bits (`int` en nuestro C++ o en Java) almacenan números entre -2^{31} y $2^{31} - 1$, inclusive.

En cambio los tipos de 64 bits (`long` en nuestro C++ y `long` en Java) almacenan números entre -2^{63} y $2^{63} - 1$, inclusive.

Recuerda también que si sumas dos variables `int`, habrá error en los cálculos si su resultado no entra en `int`, por más que luego guardes ese resultado (erróneo) en un `long long`.

Descripción de la función

Debes implementar la función

`suma(v : ARREGLO[N] de ENTEROS)`
: ENTERO LARGO

Donde `v` será el arreglo con los **N** números que se debe sumar. La función debe retornar la suma calculada.

Evaluador

El evaluador local lee la entrada de `stdin` con el siguiente formato:

- Una línea con un número *N*, que indica la cantidad de números.
- *N* líneas, cada una con un número del arreglo.

El evaluador devuelve una única línea con el resultado devuelto por la función `suma`.

Cotas

$$1 \leq N \leq 1.000$$

$$1 \leq v_i \leq 10^9$$

Ejemplo

Si se invoca al evaluador con la siguiente entrada:

```
5
4
5
2
6
9
```

Realiza la llamada `suma({4,5,2,6,9})`, que con una implementación correcta le devolvería el entero largo 26, que es la suma de los elementos del arreglo `v` en este caso.

Subtareas

Los puntajes de cada subtarea están dados en la siguiente tabla:

$N = 1$	5 puntos
$N = 2, v_i \leq 1.000$	10 puntos
$N = 5$	10 puntos
$v_i \leq 1.000$	30 puntos
Sin más restricciones	45 puntos