

Control 1

Pares suman, impares restan

El objetivo de este control es especificar un algoritmo para resolver un problema sencillo, y diseñar sistemáticamente una implementación a partir de dicha especificación. Como objetivo secundario se persigue que el alumno se familiarice con la dinámica de resolución de problemas en *DomJudge*.

El problema

Dado un vector `int a[n]`, con $n \geq 0$, debe encontrarse el resultado de sumar todos sus elementos pares, y restar todos sus elementos impares.

Por ejemplo, si el vector contiene los siguientes valores:

1, 4, 8, 5, 6

el resultado será $-1 + 4 + 8 - 5 + 6 = 12$

Trabajo a realizar

Para realizar el control se proporciona un archivo `control1.cpp` que contiene un programa que lee por la entrada estándar vectores, y escribe por salida el resultado pedido invocando a la función `pares_menos_impares`.

A continuación, se muestra un ejemplo de entrada procesable por este programa, y de salida producida (suponiendo una implementación adecuada de `pares_menos_impares`) (los vectores se introducen indicando, como primer valor, el número de elementos, y, a continuación, enumerando los distintos elementos; el final se indica indicando -1 como tamaño del vector):

Entrada	Salida
5 1 4 8 5 6	12
1 1	-1
0	0
5 -1 -4 -8 -5 -6	-12
-1	

Tu trabajo consiste en:

- Especificar formalmente el algoritmo que calcula el resultado pedido, rellenando los huecos correspondientes en los comentarios que rodean a `pares_menos_impares`
- Diseñar **sistemáticamente** una implementación C++ para dicho algoritmo. Debes describir la derivación en el comentario habilitado a tal fin en el archivo proporcionado, siguiendo el método de diseño presentado en clase.
- Completar la función `pares_menos_impares` a partir del diseño realizado.
- Entregar `control3.cpp` a través del juez en línea de la asignatura.

Importante:

- Únicamente se evaluarán aquellas entregas que superen satisfactoriamente los casos de prueba del juez.
- No modificar el código proporcionado. Únicamente deben responderse a los distintos apartados, en el interior de los comentarios, e implementar la función `pares_menos_impares`. Para escribir las fórmulas de la especificación en texto plano, pueden utilizarse los siguientes convenios:

Símbolo	Representación en texto plano
\forall	PARA_TODO
\exists	EXISTE
Σ	SUMA
Π	PROD

Por su parte, las conectivas lógicas y operadores de relación no expresables directamente en texto plano, pueden escribirse utilizando la notación de C++.

Ejemplo:

$\forall i: 0 \leq i < n: (\exists j: 0 \leq j < n: (((\Pi k: 0 \leq k < j: a[k]) = (\Sigma r: j \leq r < i: a[r])) \vee a[j] = 0))$

Escrita en texto plano:

PARATODO i: 0<=i<n: (EXISTE j: 0<=j<n:
(((PROD k:0<=k<j:a[k]) = (SUMA r:j<=r<i:a[r])) v a[j]=0))