

Tramo par más largo

El problema

Dado vector **int** $v[n]$ de enteros ($n \geq 0$), se dice que una secuencia de valores consecutivos de v es un *tramo par* cuando todos los valores son números pares.

Debe especificarse, diseñarse e implementarse un algoritmo iterativo **eficiente** que, dado el vector **int** $v[n]$ determine la **longitud** (es decir, el número de valores) del **tramo par más largo** en v (Nota: En caso de que v no contenga tramos pares, el resultado deberá ser 0). Debe determinarse, asimismo, justificadamente el orden de complejidad del algoritmo.

Trabajo a realizar

Para realizar el control se proporciona un archivo `tramospares.cpp` que contiene un programa que lee por la entrada estándar vectores, invoca a la función `long_tramo_par_mas_largo` sobre dichos vectores, e imprime el valor devuelto por dicha función.

Los vectores se introducen indicando, como primer valor, el número de elementos, y, a continuación, enumerando los distintos elementos; el final se indica introduciendo -1 como tamaño del vector. El tamaño máximo de los vectores soportados por este programa de prueba es de un millón de elementos.

A continuación, se muestra un ejemplo de entrada procesable por este programa, y de salida producida (suponiendo una implementación adecuada de `long_tramo_par_mas_largo`):

Entrada	Salida
10 2 4 5 7 9 6 8 10 12 14	5
5 2 4 5 8 10	2
3 1 3 5	0
-1	

Tu trabajo consiste en:

- Especificar y diseñar el algoritmo pedido, siguiendo el método explicado en clase.
- Determinar su complejidad.
- Implementar el algoritmo como cuerpo de la función `long_tramo_par_mas_largo`.
- Entregar `tramospares.cpp` a través del juez en línea de la asignatura.

Importante:

Deben indicarse los nombres y apellidos de los autores del control en el comentario que aparece al principio del archivo `tramospares.cpp` (**de lo contrario, el control no puntuará**). En caso de que alguno de los miembros del grupo no haya realizado el control, se deberá indicar en dicho comentario.

Únicamente puntuarán las soluciones que superen satisfactoriamente los casos de prueba del juez, en el tiempo máximo asignado para ello (veredicto de CORRECT). El ejercicio deberá resolverse **presencialmente**, en el laboratorio. Finalizada la hora del laboratorio, el juez permanecerá abierto dos horas más. No obstante, la puntuación de los envíos correctos enviados en ese plazo de tiempo se penalizará, no pudiéndose superar los 6.0 puntos sobre 10.0.

Importante: aparte de la implementación, el algoritmo debe especificarse, diseñarse y analizarse correctamente, utilizando el método explicado en clase, utilizando los comentarios habilitados al respecto. Llevar a cabo una especificación, un diseño (invariante, cota, justificación de la corrección parcial y de la terminación ...) y un análisis de la complejidad correctos será esencial para superar esta prueba.

Únicamente se tendrá en cuenta el último envío CORRECT enviado dentro de plazo.

No modificar el código proporcionado. Únicamente debe responderse a los distintos apartados, en el interior de los comentarios, e implementar la función `long_tramo_par_mas_largo` (sí pueden incluirse funciones auxiliares, si se considera necesario).