

Ejercicio 1 (30 pts)

Se desea analizar la información obtenida de un dispositivo electrónico usado en medicina que almacena los datos obtenidos en un archivo ASCII. Se dispone de uno de los archivos que genera este dispositivo: *data20211104.txt* que contiene valores enteros (uno por línea).

- a) Escriba un programa C++ que lea el archivo y organice los datos en algún contenedor STL.
- b) Diseñe una función C++ llamada *procesa_secuencia(...)* que debe ser invocada desde el programa y haga lo siguiente:
 - i) si el tamaño de la secuencia es impar eliminar el elemento del medio;
 - ii) luego debe ordenar de menor a mayor la primera mitad de la secuencia y de mayor a menor la segunda mitad
 - iii) obtener la suma de los elementos de las 2 mitades; estos 2 valores deben ser devueltos por la función.
- c) Luego de invocar la función, el programa debe mostrar los 2 promedios y la secuencia modificada; d) Reemplazar los datos del archivo por la nueva secuencia modificada.

Siempre que sea posible, resuelva la consigna usando métodos y algoritmos de la STL en lugar de hacer sus propios algoritmos.

Ejercicio 2 (35 pts)

Un archivo binario "*productos.dat*" contiene información sobre los productos que vende una empresa. Por cada producto se guarda mediante un *struct Producto* el código (un número entero), el precio, la descripción, y el stock (cuántas unidades quedan disponibles).

Otro archivo binario "*ventas.dat*" contiene un registro de las ventas. Por cada venta se guarda uno o varios *structs Detalle* con la fecha de la venta, el código de un cliente, el código de un producto y la cantidad de unidades de ese producto que compró el cliente. Notar que si el cliente compra más de un producto, se registra más de un *Detalle* para la misma venta.

- a) Proponga posibles definiciones para los *structs Producto* y *Detalle*.
- b) Escriba una función para actualizar el archivo de ventas. Debe recibir el dni de un cliente, la fecha de una venta, y dos vectores de enteros: uno con los códigos de los productos de esa venta, y otro con las cantidades vendidas de cada uno. El método debe registrar la venta en el archivo "*ventas.dat*" (agregar un *Detalle* por cada producto), y actualizar también el archivo "*productos.dat*" (descontar del stock las cantidades vendidas de cada producto en esa venta). Evite leer toda la información del archivo "*ventas.dat*" (ya que no es necesario usar información de ventas anteriores en esta función).

Ejercicio 3 (30 pts)

- a) Diseñe una función genérica *mas_repetido(...)* que reciba una lista (*std::list*) con elementos cualquier tipo y retorne el elemento de la lista que más veces se repite.
- b) Escriba un programa C++ cliente de la función *mas_repetido(...)* que invoque a la función 3 veces:
 - i) con una secuencia de strings leída por consola,
 - ii) con un listado de enteros entre 100 y 999 generados al azar
 - iii) con un listado de datos de tipo *struct Ficha { int dni; string apellido, nombres; };* que se leen de un archivo de texto como el que sigue:

```
35938475 Prince, Martin
34245353 Van Houten, Milhouse
36547389 Gorgory, Rafa
...
```