

**Ejercicio 2 (25 pts)** Escriba una función genérica para buscar y reemplazar un dato dentro de un archivo binario. La función debe recibir el nombre del archivo y dos datos (A y B) del tipo genérico. Suponiendo que el archivo contiene elementos de dicho tipo, buscar todas las ocurrencias de A, y reemplazarlas por B. No utilice arreglos, vectores ni otros contenedores auxiliares (trabaje directamente sobre el archivo).

**Ejercicio 1 (25 pts)** Un archivo *“discurso.txt”* contiene la transcripción de un discurso de un político importante. El político en cuestión tiene la mala costumbre de usar palabrotas en su discurso. Sus asesores han decidido modificar la transcripción para no molestar a los votantes más susceptibles. Para ello han generado un archivo *“palabrotas.txt”*, con la lista de las que usa con más frecuencia. Escriba un programa c++ que modifique el archivo del discurso reemplazando todas las palabrotas por *“\*\*\*beep\*\*\*”*. No hay palabras cortadas en el texto y puede haber 2 o mas palabrotas en una línea..

**Ejercicio 2 (25pts)** a) Escriba una función genérica *mediana3* que reciba 3 valores de un mismo tipo genérico y retorne la mediana (la mediana es uno de los 3, el que no es ni el mínimo ni el máximo). b) Escriba un programa cliente que muestre la mediana entre 3 flotantes que ingrese el usuario; y entre 3 fracciones. Las fracciones deben ser del tipo: *struct Fraccion { int num, den; };*.

**Ejercicio 3 (30pts)** Un archivo binario guarda datos de usuarios de un sistema. Por cada usuario se guarda código (entero), nombre (cadena) y contraseña (cadena). Se quiere encontrar contraseñas comunes para sugerirle a esos usuarios que las modifiquen. Escriba un programa que: a) lea los datos y genere una lista de todas las contraseñas que estén repetidas más de 10 veces (que más de 10 usuarios tengan la misma). La lista debe estar ordenada por cantidad de repeticiones (primero las que más se repiten); y b) modifique el archivo estableciendo la cadena vacía como contraseña para todos los usuarios que tengan alguna de las contraseñas de la lista.

**Ejercicio 1 (25 pts)** Un archivo de textos INFORME.TXT contiene un reporte de funcionamiento de un paquete de software y se desea modificar el texto de forma de reemplazar la palabra “paquete” por la palabra “aplicación”. Es posible que en una línea aparezca más de una vez la palabra “paquete”. No hay palabras cortadas en el texto.

**Ejercicio 2 (25pts)** Diseñe una clase genérica llamada *Calculos* que tenga como atributos un entero n (tamaño del vector), un vector de n elementos genéricos y 2 datos de mismo tipo genérico que los elementos del vector. La clase debe tener un constructor, un método llamado *dosmax(..)* que obtenga los 2 datos mayores del vector y métodos que considere necesarios. En un programa cliente instancie la clase para una lista de flotantes y para una lista de strings que deben leerse como datos.

**Ejercicio 3 (30pts)** Un archivo binario contiene una base de datos de registros de tipo ficha como el que se indica en el recuadro generada por un programa C++. Cuando un usuario borra un registro desde el programa, marca el registro como borrado colocando *true* en el campo *borrado*, pero no lo quita del archivo. El archivo es ahora demasiado extenso. Escriba un programa C++ que - abra el archivo y lo ‘limpie’ eliminando definitivamente todos los registros con miembro *borrado* en *true*. Además, los datos deben ordenarse en forma decreciente por dni quedando así ordenados en el archivo. *struct ficha { bool borrado; int dni, char nombre[50], float h; }*

**Ejercicio 3)** Mediante un programa C++: a) defina un vector (STL) de strings, incorporando sus datos desde un archivo de textos LISTASTR.TXT.; b) elimine los strings repetidos; c) si un string tiene 2 o más blancos debe eliminar el 2do blanco y la subcadena que se encuentra entre el primer blanco y el segundo; d) ordene alfabéticamente la lista, pero a partir del 3er elemento; e) reescriba el archivo LISTASTR.TXT con la lista modificada.

**Ejercicio 2)** El siguiente código muestra un ejemplo de uso de una clase genérica MejoresPunt que permite administrar los N mejores puntajes obtenidos en un videojuego. Escriba el código de las clases Puntuacion y MejoresPunt para que el ejemplo funcione como se indica en sus comentarios.

```
// gestiona las N=10 mejores, lee los datos del archivo
"records.dat"
MejoresPunt<Puntuacion> mp(10,"records.dat");
// supongamos que un jugador llamado "Jug1" obtiene 1537
Puntuacion p_nueva("Jug1",1537); pts
// si está entre las mejores, agregarla en la lista
mp.AgregarPuntuacion(p_nueva);
// muestra la tabla resultante ya ordenada
for(int i=0;i<mp.VerCant();i++)
    cout<<"mp[i].VerNombre()<<" "<<mp[i].VerPuntos()<<endl;
mp.Guardar(); // actualiza el archivo binario "records.dat"
```

**Ejercicio 3 (25pts)** Escriba una función que reciba el nombre de un archivo binario que contenga un conjunto de registros de un tipo genérico, y elimine del archivo todos los registros repetidos. La función debe retornar la cantidad de elementos eliminados. Nota: la función podría alterar el orden de los registros si lo considera conveniente para su implementación.

**Ejercicio 1 (25pts)** Un archivo de texto "medallas.txt" contiene una línea por cada país que participa en los juegos olímpicos. Cada línea incluye, separados por espacios primero 3 valores enteros y luego el nombre completo del país. Los tres enteros se corresponden con las cantidades de medallas de oro, plata y bronce respectivamente que ha obtenido hasta el momento dicho país. Cada vez que una competencia finaliza, el archivo debe actualizarse. Escriba una función que reciba los nombre de los países que representan los tres ganadores de medallas de una competencia y actualice el archivo.

**Ejercicio 1 (30 pts)** Escriba una función genérica *eliminaObjetos()* que reciba como argumentos un objeto llamado *nombre* de tipo string y un arreglo de la clase vector (STL) llamado *posiciones* con enteros. La cadena *nombre* contiene el nombre de un archivo binario que almacena datos del tipo genérico. La función deberá modificar el archivo de manera que únicamente queden en el mismo los valores que se encuentren en las posiciones indicadas por el vector *posiciones*.

**Ejercicio 2 (30 pts)** Escriba una clase llamada *ManejadorDeTexto* que contenga: a) un constructor que reciba el nombre de un archivo de texto y almacene todo el contenido del mismo en un objeto de tipo string que debe ser el único atributo de la clase. Además debe tener métodos para: b) reemplazar todas las ocurrencias de una subcadena por otra (ambas cadenas se reciben como parámetros); c) guardar el texto

modificado en un archivo con un nombre que el usuario indique como parámetro; d) convertir todo el texto a mayúscula o minúscula según el valor de un argumento que reciba. Diseñar e implementar solo la clase y sus métodos.

**Ejercicio 3** (25 pts) Implemente un método llamado *rellena* perteneciente a la clase *stat\_data* que tiene como atributo un objeto *list<float> values*. La lista *values* contiene mediciones diarias de un sensor de temperatura. Por fallas en el sensor, algunos días no se registraron mediciones. Para esos días (elementos de la lista) se ha asentado el valor -1. El método debe buscar y reemplazar estos valores (-1) de la lista por el promedio de los valores adyacentes (anterior y posterior) en cada caso, y retornar el promedio total de los elementos de la lista modificada. Se sabe que en la lista no habrá nunca dos -1 consecutivos. Si hubiera un -1 en el primer o último elemento, en lugar del promedio se debe reemplazar por el único elemento adyacente válido. Proponga en la clase solo los atributos relacionados (o necesarios) para el método. Suponga que otro método se encargó de asignar los datos a la lista.

**Ejercicio 1** (20 pts) Escriba una función genérica *intercambia* que reciba como argumentos un string *nombre* y dos enteros *pos1* y *pos2*. Asumiendo que el string *nombre* contiene el nombre de un archivo binario que guarda un arreglo de datos de un mismo tipo, la función debe modificar el archivo intercambiando los datos que este guarda en las posiciones (en base 0) *pos1* y *pos2*. Por ejemplo: si el archivo "ejemplo.dat" guarda enteros, la llamada *intercambia<int>("ejemplo.dat",3,4)* debe intercambiar el cuarto con el quinto entero del archivo.

**Ejercicio 2** (30 pts) Escriba una función *listaBibliotecas* que reciba un string con el nombre de un archivo de código fuente (texto). La función debe retornar un *vector<string>* con los nombres de todas las bibliotecas que el fuente incluye. Es decir, debe buscar todas las líneas que comiencen con *#include* y de cada una obtener el nombre de la biblioteca, que puede encontrarse entre signos < y > (ej: *#include <iostream>*), o entre comillas (ej: *#include "MiClase.h"*).

**Ejercicio 3** (25 pts) En un programa C++ se definen las 2 instancias que siguen: *OperaVector <int> x;* *OperaVector <string> s.* Escriba la clase llamada **OperaVector** que posea como atributo un vector dinámico (STL vector) y su longitud. La clase debe poseer los siguientes métodos para: a) construir el vector; b) incorporar datos al vector desde un archivo binario; c) buscar y eliminar un elemento del vector (dato a eliminar como parámetro), d) ordenar el vector en forma creciente a partir de la posición (entero) de un elemento que se pasa como parámetro, e) informar los elementos del vector dentro de un rango (parámetros: 2 enteros que definen el rango).