

E. Borrar un cliente.

Se pedirá al usuario el número de cliente y se borrará toda su información del árbol. Si no existiera dicho cliente no se hará nada.

F. Mostrar la información de un cliente buscándolo por su nombre y apellido.

Se pedirá al usuario el nombre, apellido1 y apellido2 del cliente y se mostrará por pantalla toda su información. Si existiera más de un cliente con el mismo nombre y apellidos se mostrará el que tenga un número de cliente menor. Si no existiera dicho cliente se informará de dicha situación.

X. Salir del Programa. Se pedirá confirmación de salida.

Implementación:

Los módulos a implementar serán: (LA DEFINICIÓN DE LOS MÓDULOS NO SE PODRÁ MODIFICAR)

- MError: Manejo de errores.
 - Define el tipo TError como un enumerado con todos los posibles errores generados.
 - MuestraError(error): Muestra el error por pantalla.
- MCliente: Manejo del registro TCliente.
 - Define las constantes SP (espacio), SEPARADOR(#) y ENTER.
 - Define el tipo TCliente como un registro con los campos número de cliente (numérico), nombre del cliente (cadena de caracteres), Apellido1 del cliente (cadena de caracteres), Apellido2 del cliente (cadena de caracteres), Dirección del cliente (cadena de caracteres), Código Postal del cliente (numérico), Población del cliente (cadena de caracteres) y Provincia del cliente (cadena de caracteres).
 - LeeCliente(cliente): Lee un cliente desde teclado.
 - EscribeCliente(cliente): Escribe un cliente por pantalla.
 - LeeClienteFichero(fichero, cliente): Lee un cliente desde un fichero de texto cuyo descriptor se pasa como parámetro.
 - EscribeClienteFichero(fichero, cliente): Escribe un cliente en un fichero de texto cuyo descriptor se pasa como parámetro.
- MABB: Manejo de un Árbol binario de búsqueda de clientes cuya clave de inserción/búsqueda es el número de cliente.
 - Define los tipos TABB y TNode para representar un árbol binario de búsqueda de clientes.
 - CrearABB: Crea una TABB vacío.
 - DestruirABB(arbol): Destruye un TABB.
 - ABB_Vacio(arbol): Nos dice si un árbol binario de búsqueda está vacío.
 - ABB_Lleno(arbol): Nos dice si un árbol binario de búsqueda está lleno (no hay memoria disponible).
 - Raiz_ABB(arbol, raiz, error): Retorna (si existe) la información de la raíz de un árbol binario de búsqueda.
 - Izq_ABB(arbol, izq, error): Retorna (si existe) el hijo izquierdo de un árbol binario de búsqueda.
 - Dch_ABB(arbol, dch, error): Retorna (si existe) el hijo derecho de un árbol binario de búsqueda.
 - InsABB(arbol, cliente, error): Inserta el cliente en el árbol binario de búsqueda. Si ya existiera un nodo con la misma clave (número de cliente) se entenderá que estamos actualizando los datos del cliente y se copiará la información suministrada en dicho nodo sin generar un nodo nuevo. Si no hubiera memoria se informará del error.
 - EliminaABB(arbol, numCli, error): Elimina del árbol el nodo cuyo número de cliente coincida con el suministrado. Si no hay ningún nodo que cumpla dicha condición no se hará nada.

NOTAS

1. El directorio de trabajo será C:\LPDIC10
2. Mínimo para aprobar: OPCIONES A, B, y C
3. OPCIONES D, E, F y calidad de la implementación (opciones para subir nota) 5 puntos
4. **El uso de detalles de implementación del árbol fuera del módulo de implementación** del mismo (MABB.cpp) será **CAUSA de SUSPENSO**, es decir, en el programa principal no se podrá hacer cosas tipo: a= NULL, ptr=ptr->izq, etc.
5. **Añadir procedimientos o funciones a la definición de un módulo** será **CAUSA de SUSPENSO**
6. Todo **PROGRAMA QUE NO COMPILE** o tenga **efectos laterales** se considerará **SUSPENSO**.
7. El menú no distinguirá entre mayúsculas y minúsculas.
8. Se recomienda y valora el tratamiento de errores y la buena descomposición del programa principal en procedimientos y funciones, así como el uso de procedimientos y funciones auxiliares dentro de la implementación de los módulos cuando estas sean necesarias.
9. Recordad que todo fichero de cabecera debe llevar su pragma #ifndef ... #endif . Por ejemplo:

```
#ifndef _MError_h_
#define _MError_h_
. . .
#endif
```