

Nombre:	
C.I.:	

Examen 17 febrero de 2010

Ejercicio 1 (45 puntos)

Se desea modelar el concepto de libro para una biblioteca. De un libro nos interesa: **el título, el autor, el código (int) y su contenido**. El contenido es un conjunto de capítulos (suponga que ningún libro tiene más de **MAX_CAPITULOS** capítulos). Un capítulo consta de un conjunto de caracteres (suponga que ningún capítulo tiene más de **MAX_CARACTERES** caracteres).
Suponga que los libros están ordenados por su código en la lista ListaLibros.

a) Implementar la siguiente especificación del TAD ListaLibros utilizando la representación (definir el tipo Libro según los requerimientos):

```
struct NodoListaLibros{
    Libro infoLibro;
    NodoListaLibros sig;
};
```

TAD ListaLibros

```
typedef struct NodoListaLibros* ListaLibros;
//El TAD ListaLibros consta de las siguientes operaciones
```

```
//Constructoras
```

```
ListaLibros crearListaLibros();
//Crea una lista de Libros vacía
```

```
ListaLibros agregarListaLibros(ListaLibros l, char* titulo, char* autor, int codigo);
//post: agrega el libro en la lista que quedará ordenada según el código de sus libros
```

```
//predicados
```

```
void esVaciaLista(ListaLibros l, bool& esvacia);
//devuelve en el parámetro esvacia verdadero si la lista l es vacía
```

```
bool perteneceLista(ListaLibros l, int codigo);
//devuelve TRUE si existe un libro con el código codigo en la lista l, FALSE en otro caso
```

```
//selectoras
```

```
int cantidadDeElementos(ListaLibros l);
//Devuelve si la lista l no es vacía la cantidad de libros que contiene
//de lo contrario retorna cero.
```

```
int agregarCapituloLibro(ListaLibros &l, int codigo, int capitulo, char* cont_capitulo);
// Le agrega un capítulo a un libro perteneciente a la ListaLibros l (si ya había algo escrito en ese lugar lo sobrescribe, acordarse de ponerle un fin a lo que se escriba).
//Pre: l no es vacía. Pre: existe un libro con el código codigo en la lista l
//Post: En caso de que el número de capítulo y la cantidad de caracteres del capítulo sean válidos se //devuelve 0 y se agrega el capítulo al libro. Si el capítulo no es válido devuelve -1 y no hace nada. //Si la cantidad de caracteres del capítulo no son válidos devuelve 1 y no hace nada.
```

```
void destruirListaLibros(ListaLibros &l);
//Post libera toda la memoria utilizada por la lista l
```

b) Módulo de entrada y salida de datos:

Implementar el siguiente procedimiento

```
void administracionLibros(ListaLibros &l);
// Debe mostrar un menú con las siguientes opciones:
//     1 – Ingresar libro (título, autor y código)
//     2 – Ingresar capítulo (código del libro, número y contenidos del capítulo)
//     3 – Destruir lista
//     4 – Mostrar lista
//     5 – Salir (del menú, sin destruir la lista)

// Cuando se muestra la lista deberá aparecer ordenada según el código y con el siguiente formato:

//     código
//     título
//     autor
//           capítulo 1
//           capítulo 2

//           capítulo n
```

Ejercicio 2(30 puntos)

Dar una representación e implementar el TAD **Árbol Binario de Búsqueda de enteros CON REPETICIÓN** con las siguientes operaciones:

```
typedef struct Nodo* ABB;

//constructoras
void crearABB(ABB &d);
//Crea un ABB vacío

void agregarABB(ABB &d, int n);
//post: agrega al ABB d el entero n, en caso de ya existir el entero en el árbol se debe incrementar la //cantidad de
ocurrencias del mismo.

//predicados
bool perteneceABB(ABB d, int n)
//post: retorna verdadero si el entero n se encuentra en el ABB.

//selectoras
int cantidadOcurrenciasABB(ABB d, int n);
//devuelve la cantidad de ocurrencias del entero n en el ABB d
// pre: n pertenece a d

//destructoras
void quitarABB(ABB &d, int n);
//pre: n pertenece a d
//post: elimina una ocurrencia de n en d. En caso de que hubiera sólo un elemento n en d lo borra totalmente del ABB

void destruirABB(ABB &d);
//post: libera el espacio en memoria ocupado por ABB.
```

Ejercicio 3(25 puntos)

Explique cómo funciona y luego codifique algún método de ordenación de arreglos de enteros.

El cabezal del procedimiento será: ***void ordena (int arreglo[], int tamaño);***