

## Práctica 2. Estructuras de datos avanzadas

**P2.1.** En este ejercicio debes completar la librería para trabajar con árboles binarios de enteros. Tienes a tu disposición el fichero `arbolBin.h`, que contiene las exportaciones de la librería. Tú debes crear el fichero `arbolBin.c`, con la implementación de todas las funciones exportadas. Ten en cuenta que el árbol debe ser dinámico, por lo que en todo momento el tamaño se debe ajustar al número de elementos almacenados en él. Las funciones que debes implementar son:

- Crear un nuevo árbol
- Construir un árbol a partir de un elemento raíz y dos subárboles
- Devolver el elemento almacenado en la raíz
- Mostrar los elementos del árbol en preorden
- Mostrar los elementos del árbol en inorden
- Mostrar los elementos del árbol en postorden
- Comprobar si el árbol está vacío

Para comprobar el correcto funcionamiento de tu librería, se te proporciona el archivo `ejecutarArbolBin.c`, que contiene un programa con la función `main`. Este fichero no debes modificarlo.

Ejemplo:

```
Introduce 7 numeros para crear el arbol: 1 2 3 4 5 6 7
El elemento de la raíz es 7
El arbol es preorden es: 7 5 3 1 2 4 6
El arbol es inorden es: 1 3 2 5 4 7 6
El arbol es postorden es: 1 2 3 4 5 6 7
```

**P2.2.** En este ejercicio debes completar la librería para trabajar con árboles binarios de búsqueda. Tienes a tu disposición el fichero `arbolBinarioBusqueda.h`, que contiene las exportaciones de la librería. Tú debes crear el fichero `arbolBinarioBusqueda.c`, con la implementación de todas las funciones exportadas. Ten en cuenta que el árbol debe ser dinámico, por lo que en todo momento el tamaño se debe ajustar al número de elementos almacenados en él. Las funciones que debes implementar son:

- Crear un nuevo árbol
- Insertar un elemento en el árbol
- Borrar un elemento del árbol
- Mostrar los elementos del árbol en preorden
- Mostrar los elementos del árbol en inorden
- Mostrar los elementos del árbol en postorden
- Comprobar si el árbol está vacío

Para comprobar el correcto funcionamiento de tu librería, se te proporciona el archivo `ejecutarArbolBB.c`, que contiene un programa con la función `main`. Este fichero no debes modificarlo.

Ejemplo:

```
-----MENU-----
1 - Insertar un elemento en el arbol
2 - Eliminar un elemento del arbol
3 - Mostrar el arbol en preorden
4 - Salir.
Escoja una opcion: 1
Introduce el entero: 10
-----MENU-----
1 - Insertar un elemento en el arbol
2 - Eliminar un elemento del arbol
3 - Mostrar el arbol en preorden
4 - Salir.
Escoja una opcion: 1
Introduce el entero: 6
-----MENU-----
1 - Insertar un elemento en el arbol
2 - Eliminar un elemento del arbol
3 - Mostrar el arbol en preorden
4 - Salir.
Escoja una opcion: 1
Introduce el entero: 2
-----MENU-----
```

```
-----MENU-----
1 - Insertar un elemento en el arbol
2 - Eliminar un elemento del arbol
3 - Mostrar el arbol en preorden
4 - Salir.
Escoja una opcion: 1
Introduce el entero: 8
-----MENU-----
1 - Insertar un elemento en el arbol
2 - Eliminar un elemento del arbol
3 - Mostrar el arbol en preorden
4 - Salir.
Escoja una opcion: 1
Introduce el entero: 18
-----MENU-----
1 - Insertar un elemento en el arbol
2 - Eliminar un elemento del arbol
3 - Mostrar el arbol en preorden
4 - Salir.
Escoja una opcion: 1
Introduce el entero: 12
```

```
-----MENU-----
1 - Insertar un elemento en el arbol
2 - Eliminar un elemento del arbol
3 - Mostrar el arbol en preorden
4 - Salir.
Escoja una opcion: 2
Introduce el entero: 10
-----MENU-----
1 - Insertar un elemento en el arbol
2 - Eliminar un elemento del arbol
3 - Mostrar el arbol en preorden
4 - Salir.
Escoja una opcion: 3
Los elementos del arbol son:
12 6 2 8 18
-----MENU-----
1 - Insertar un elemento en el arbol
2 - Eliminar un elemento del arbol
3 - Mostrar el arbol en preorden
4 - Salir.
Escoja una opcion: 4
```

**P2.3.** En este ejercicio debes completar la librería para trabajar con min-montículos. Tienes a tu disposición el fichero `minMonticulo.h`, que contiene las exportaciones de la librería. Tú debes crear el fichero `minMonticulo.c`, con la implementación de todas las funciones exportadas. Las funciones que debes implementar son:

- Crear un nuevo montículo
- Insertar un elemento en el montículo
- Borrar un elemento del montículo
- Devolver el elemento almacenado en la raíz
- Mostrar el montículo en anchura
- Comprobar si el montículo está vacío
- Comprobar si el montículo está lleno

Para comprobar el correcto funcionamiento de tu librería, se te proporciona el archivo ejecutarMinMontículo.c, que contiene un programa con la función main. Este fichero no debes modificarlo.

Ejemplo:

```
Introduce el numero maximo de elementos: 9
-----MENU-----
1 - Insertar un elemento en el monticulo
2 - Eliminar un elemento del monticulo
3 - Mostrar el elemento de la raiz
4 - Mostrar el monticulo en anchura
5 - Salir.
Escoja una opcion: 1
Introduce el entero: 5
-----MENU-----
1 - Insertar un elemento en el monticulo
2 - Eliminar un elemento del monticulo
3 - Mostrar el elemento de la raiz
4 - Mostrar el monticulo en anchura
5 - Salir.
Escoja una opcion: 1
Introduce el entero: 4
-----MENU-----
1 - Insertar un elemento en el monticulo
2 - Eliminar un elemento del monticulo
3 - Mostrar el elemento de la raiz
4 - Mostrar el monticulo en anchura
5 - Salir.
Escoja una opcion: 1
Introduce el entero: 1
-----MENU-----
1 - Insertar un elemento en el monticulo
2 - Eliminar un elemento del monticulo
3 - Mostrar el elemento de la raiz
4 - Mostrar el monticulo en anchura
5 - Salir.
Escoja una opcion: 1
Introduce el entero: 8
-----MENU-----
1 - Insertar un elemento en el monticulo
2 - Eliminar un elemento del monticulo
3 - Mostrar el elemento de la raiz
4 - Mostrar el monticulo en anchura
5 - Salir.
Escoja una opcion: 1
Introduce el entero: 6
-----MENU-----
1 - Insertar un elemento en el monticulo
2 - Eliminar un elemento del monticulo
3 - Mostrar el elemento de la raiz
4 - Mostrar el monticulo en anchura
5 - Salir.
Escoja una opcion: 1
Introduce el entero: 2
-----MENU-----
1 - Insertar un elemento en el monticulo
2 - Eliminar un elemento del monticulo
3 - Mostrar el elemento de la raiz
4 - Mostrar el monticulo en anchura
5 - Salir.
Escoja una opcion: 4
Los elementos del monticulo son:
1 5 2 8 6 4
-----MENU-----
1 - Insertar un elemento en el monticulo
2 - Eliminar un elemento del monticulo
3 - Mostrar el elemento de la raiz
4 - Mostrar el monticulo en anchura
5 - Salir.
Escoja una opcion: 2
Introduce el entero: 1
-----MENU-----
1 - Insertar un elemento en el monticulo
2 - Eliminar un elemento del monticulo
3 - Mostrar el elemento de la raiz
4 - Mostrar el monticulo en anchura
5 - Salir.
Escoja una opcion: 4
Los elementos del monticulo son:
2 5 4 8 6
-----MENU-----
1 - Insertar un elemento en el monticulo
2 - Eliminar un elemento del monticulo
3 - Mostrar el elemento de la raiz
4 - Mostrar el monticulo en anchura
5 - Salir.
Escoja una opcion: 5
```

**P2.4.** Implementa en un archivo .c la solución al siguiente problema. Queremos ordenar un conjunto de números enteros. Para ello, vamos a utilizar un árbol binario de búsqueda. Por tanto, para la resolución de este problema debes usar la librería que has creado en el ejercicio 2.2.

Ejemplo:

```
Introduce elementos para ordenar. 0 para terminar: 4 6 2 3 1 7 8 9 5 0
Los elementos ordenados son: 1 2 3 4 5 6 7 8 9
```

**P2.5.** Implementa en un archivo .c la solución al siguiente problema. Queremos ordenar un conjunto de números enteros. Para ello, vamos a utilizar un min-montículo. Por tanto, para la resolución de este problema debes usar la librería que has creado en el ejercicio 2.3.

Ejemplo:

```
Introduce el numero de elementos a ordenar: 6
Introduce los elementos: 4 8 2 1 6 7
Los elementos ordenados son: 1 2 4 6 7 8
```

**P2.6.** Implementa en un archivo .c la solución al siguiente problema. Para la resolución, puedes utilizar las librerías que has creado en los ejercicios anteriores, o crear unas muy similares si necesitas que trabajen con otro tipo de datos.

Dado un string que contiene una operación en notación postfija (operando1 operando2 operador) imprimir la misma operación en notación infija (operando1 operador operando2). Además, utiliza paréntesis para englobar cada suboperación.

Ejemplo:

```
Introduce la operacion en notacion posfija, sin espacios:
531-*42-+
(((5)*((3)-(1)))+(4)-(2)))
```

**P2.7.** Implementa en un archivo .c la solución al siguiente problema. Para la resolución, puedes utilizar las librerías que has creado en los ejercicios anteriores.

Dados los recorridos en preorden e inorden de un mismo árbol binario, construir dicho árbol y mostrar su recorrido en postorden.

Ejemplo:

```
Introduce el numero de elementos: 9
Introduce el preorden:
1 2 4 7 8 3 5 6 9
Introduce el inorden:
2 7 4 8 1 5 3 9 6
El arbol en postorden es:
7 8 4 2 5 9 6 3 1
```