

- Para esta actividad debes entregar un documento de tipo pdf y toda la carpeta de solución de Visual Studio. Antes de comprimir la carpeta debes borrar los ficheros innecesarios. Para ello debes ir al menú Build, y elegir “Clean Solution” (limpiar solución). En el caso contrario se perderá puntos.
 - Se hace la practica en grupos de dos. Solo uno de los alumnos componentes del grupo debe entregar la práctica.
 - Es necesario escribir como un comentario los nombres de los autores al principio del fichero que contiene el programa principal y también en el documento. En el caso contrario se perderá puntos.
 - Si en algún ejercicio se ha indicado el orden o formato de entradas, debes respetar este orden/formato. En el caso contrario se perderá puntos.
-

El objetivo de este ejercicio es poner en práctica la navegación de nodos en árboles a través de punteros, mostrando el contenido de distintas maneras. La estructura de datos que se usa para esta actividad es de tipo árbol binario. Se pide implementar un árbol de búsqueda binaria con inserción ordenada, y los métodos de impresión de datos: Inorder, preorder y postorder.

Parte 1 – Experimento (6 puntos):

Para implementar el árbol de búsqueda binaria, se pide crear la clase “ArbolBinarioDeBusqueda” con las siguientes características (se puede reutilizar la implementada en la actividad 7.2, **no se piden los métodos de eliminación**):

- Atributos:
 - Puntero a la raíz del árbol.
 - Número de nodos del árbol
- Métodos:
 - Constructor que construirá un árbol vacío, y destructor que liberará la memoria de todos los nodos. El destructor será recursivo, como en la actividad 7.1.
 - Insertar un elemento: buscará la posición en donde deberíamos insertar el nuevo elemento (para ello usará el método privado descrito a continuación) y lo almacenará ahí (sólo como hojas).
 - Un método privado (recursivo) para buscar el hueco en donde se debería insertar un nuevo elemento **de forma ordenada**. Si el valor a insertar es “menor” al del nodo, se inserta a la izquierda. Si no, a la derecha. Devuelve un puntero al nodo padre de dicho hueco.
 - Imprimir. Dado un nodo raíz de un árbol ó subárbol, imprimirá el contenido del árbol. El método será implementado de forma recursiva, y admitirá 3 tipos de impresión de contenido:

Algoritmos y Estructuras de Datos

Árboles

Práctica 4

- Inorder (Left, Root, Right): Dado un nodo, imprime primero el contenido del subárbol izquierdo (si existe), luego el contenido del nodo, y por último el contenido del subárbol derecho (si existe).
- Preorder (Root, Left, Right): Dado un nodo, imprime primero el contenido del nodo, luego el contenido del subárbol izquierdo (si existe), y por último el contenido del subárbol derecho (si existe).
- Postorder (Left, Right, Root): Dado un nodo, imprime primero el contenido del subárbol izquierdo (si existe), luego el contenido del subárbol derecho (si existe), y por último el contenido del nodo.

- La cabecera de la función será la siguiente:

```
#define INORDER 0
```

```
#define PREORDER 1
```

```
#define POSTORDER 2
```

```
// ... //
```

```
void imprimir(int orden); //método público, llamado desde el main del
```

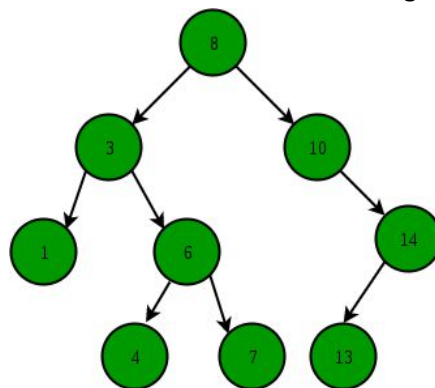
```
//programa
```

```
void imprimirRec(Nodo* root, int orden); //método privado, llamado desde el
```

```
//método imprimir con el nodo raíz por parámetros
```

- Ejemplos de uso:

- Dado el siguiente estado del árbol, los métodos de impresión de datos devolverán los siguientes resultados:



- Inorder: 1,3,4,6,7,8,10,13,14
- Preorder : 8,3,1,6,4,7,10,14,13
- Postorder : 1,4,7,6,3,13,14,10,8

A continuación, escribir un programa main con los siguientes pasos:

1. El programa mostrará un menú con las siguientes opciones:

- 1 Introducir número
- 2 Introducir 50 números aleatorios
- 3 Mostrar Inorder
- 4 Mostrar Preorder
- 5 Mostrar Postorder
- 6 Salir

Algoritmos y Estructuras de Datos
Árboles
Práctica 4

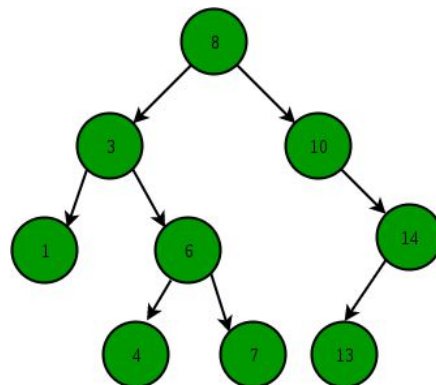
El programa leerá por terminal la opción elegida por el usuario, y realizará las operaciones indicadas.

Parte 2 – Análisis de resultado (4 puntos):

Se pide escribir un documento en formato pdf para analizar detalladamente el resultado. Probar el programa con 5 números pedidos por pantalla al usuario, y mostrar el resultado usando las opciones “inorder”, “preorder” y “postorder”. Seguidamente, introducir 50 números aleatorios (opción 2), y volver a mostrar los resultados (55 números en este caso) con los tres algoritmos implementados.

El documento debe contener:

- Para la prueba realizada con los 5 números pedidos por pantalla al usuario, incluir un diagrama de árbol que represente el contenido de los nodos y su jerarquía (un diagrama similar al siguiente, con números distintos):



- Para el resto de pruebas, incluir una captura de pantalla que contenga el resultado del programa.