

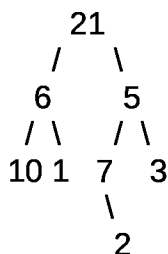
**Estructuras de Datos**  
**Curso 2015-2016. Convocatoria de Febrero**  
**Grado en Ingeniería Informática.**  
**Doble Grado en Ingeniería Informática y Matemáticas**

1. (1 punto) Construye una clase **agenda** que de soporte al manejo de información del tipo persona: <nombre, num\_telefono, direccion>. Los nombres son todos distintos y se tiene que poder buscar de forma eficiente tanto por nombre como por número de teléfono o dirección. Además de en tiempo también debe ser eficiente en espacio. Implementa las funciones: persona **buscar\_nombre**(string nom), persona **buscar\_telefono**(string num), void **insertar**(persona p)
2. (1.5 puntos) Implementa una función bool **sumaligual**(const list<list<int> > &L) que devuelve true si todas las sumas por filas y por columnas da el mismo resultado (se asume que todas las listas tienen igual longitud). Por ejemplo, en la siguiente lista si se suma cada fila y cada columna siempre se obtiene 3

$L = \langle \langle 0, 1, 1, 1 \rangle, \langle 2, 1, 0, 0 \rangle, \langle 1, 0, 2, 0 \rangle, \langle 0, 1, 0, 2 \rangle \rangle$

3. (1.5 puntos) Una **secuencia guiada** sobre un árbol binario de enteros comienza por la raíz para a continuación, en cada paso, seleccionar el nodo con etiqueta más pequeña de entre los nodos disponibles en ese momento que no hayan sido listados. Se entiende por **nodo disponible** aquel cuyo padre ya ha sido procesado (excluyendo la raíz).
  - a) ¿Qué estructura auxiliar utilizarías para almacenar el conjunto de nodos disponibles? Razona tu respuesta.
  - b) Implementar una función que permita obtener una secuencia **guiada** de un árbol binario.

Ejemplo:



**Recorrido guiado: 21, 5, 3, 6, 1, 7, 2, 10**



4. (1 punto) Tenemos un contenedor de pares de elementos, {clave, vector<int>} definida como:

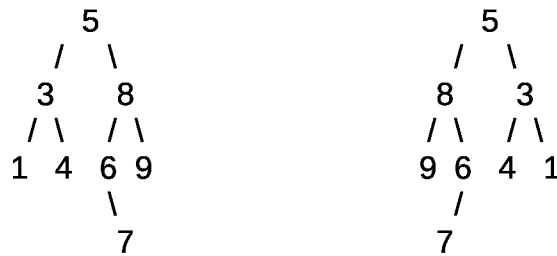
```
template <typename T>
class contenedor{
private:
    unordered_map<T, vector<int> > datos;
    .....
    .....
}
```

Implementa un iterador que itere sobre los elementos que cumplan la propiedad de que la suma de los elementos del vector<int> sea un número impar. Han de implementarse (aparte de las de la clase iteradora) las funciones begin() y end().

5. (1 punto) Implementa una función que compruebe si un árbol binario es simétrico de otro.

***bool simetricos(const bintree<T> &a, const bintree<T> &b).***

Ejemplo:



**Tiempo: 3 horas**