



- Duración del examen: 3 horas.
  - Todas las hojas entregadas deben tener nombre y DNI del alumno y N° de página/Total de páginas.
  - Las respuestas a los ejercicios deberán escribirse en pseudocódigo y estar claramente justificadas, es decir, acompañadas de una breve y clara explicación.
  - En todas las preguntas pueden incluirse operaciones auxiliares, debidamente justificadas, si se considera necesario.
  - En todas las preguntas deben especificarse claramente las operaciones básicas del TAD (tipos de entrada y salida de las mismas y ecuaciones de definitud) y describir su funcionamiento.
- 

## Ejercicio 1. (1,5 puntos)

- Definir árbol binario, árbol binario de búsqueda y árbol AVL. Explicar con un ejemplo la diferencia entre estos conceptos. (0,5 puntos)
- Insertar los siguientes nodos en un AVL vacío, indicando el factor de desequilibrio en cada paso y las rotaciones realizadas: 12, 7, 4, 13, 9, 10, 6 en un árbol AVL vacío. (0,5 puntos)
- Eliminar el nodo 9 en el árbol resultante del apartado anterior. (0,5 puntos)

**Ejercicio 2.** (2 puntos)- Suponiendo que los elementos de las listas pueden compararse con una operación  $\leq$ :  $\text{elemento elemento} \rightarrow \text{bool}$ , que comprueba si un elemento es menor o igual que otro, extender la especificación del TAD LISTA2[ELEMENTO] vista en clase con las siguientes operaciones (pueden ser parciales):

- estaOrdenada**:  $\text{lista} \rightarrow \text{bool}$ , comprueba si la lista esta ordenada. (1 punto)
- insertaEnOrden**:  $\text{lista objeto} \rightarrow \text{lista}$ , añade un nuevo objeto a la lista en el orden correcto, únicamente si la lista está ordenada. (1 punto)

## Ejercicio 3. (1,5 puntos)

- Escribir y explicar una representación para el TAD LISTA2[ELEMENTO] con memoria dinámica, utilizando una lista doblemente enlazada. (0,5 puntos)
- Escribir en pseudocódigo la operación b) del ejercicio 2 utilizando la representación del TAD LISTA2[ELEMENTO] con memoria dinámica descrita en el apartado a). (1 punto)

**Ejercicio 4.** (2,5 puntos)- El administrador de un sistema informático quiere utilizar el TAD ÁRBOL\_BINARIO[CARACTERES] para almacenar y verificar la validez de las claves de acceso al mismo. Dichas claves se almacenan de forma que en raíz está el primer carácter (común a todas las claves) y cada uno de sus hijos almacena el siguiente posible carácter en la clave y así sucesivamente. Cada una de las claves se encontrará almacenada en uno de los caminos desde la raíz hasta una de las hojas del árbol binario.

Partiendo de la operación ==:carácter carácter-->boolean que comprueba si dos caracteres son iguales y la especificaciones ÁRBOLES\_BINARIOS[CARACTERES] y COLA[CARACTERES], escribir las siguientes operaciones:

- a) Comprobar si un carácter dado es uno de los caracteres permitidos en las claves del sistema, es decir, si forma parte de alguna de las claves del árbol. (0,75 puntos)
- b) Sustituir, en todas las claves del árbol, el carácter \$ por el carácter #. (0,75 puntos)
- c) Comprobar si una clave dada, almacenada en un cola de caracteres, es o no válida. Se comprobará para ello si está almacenada en una de las ramas del árbol.

(1 punto)

**Ejercicio 5.** (2,5 puntos)- El administrador del sistema informático del ejercicio 4, ante el aumento del número de usuarios del sistema, decide utilizar un árbol general para almacenar las claves del mismo. Todas ellas comienzan por el valor almacenado en la raíz del árbol y las posibles combinaciones válidas se obtienen recorriendo una de las ramas del mismo hasta llegar a la hoja.

Partiendo de la operación ==:carácter carácter-->boolean que comprueba si dos caracteres son iguales y las especificaciones ÁRBOLES[CARACTERES] y COLA[CARACTERES], reescribir las operaciones del ejercicio 4 (la puntuación de cada apartado es la misma que en el ejercicio 4).