



Nombre y apellidos:

DNI:

- Duración del examen: 2 horas y 30 minutos.
- Todas las hojas entregadas deben tener nombre y DNI del alumno y Nº 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/deben incluirse operaciones auxiliares, debidamente justificadas, si se considera necesario.
- En los ejercicios 1 y 4, deben especificarse claramente las operaciones básicas de los TAD (indicando los tipos de entrada y salida de las mismas y las ecuaciones de definitud) y describir su funcionamiento.

Ejercicio 1. (3 puntos). En un árbol binario se almacenan expresiones algebraicas formadas únicamente por dígitos y operaciones binarias básicas (+, *, --, /).

Partiendo de las siguientes operaciones:

- *es_dígito?* carácter \rightarrow boolean, que comprueba si un carácter dado es un dígito.
- *es_op_binaria?*:carácter \rightarrow boolean, que comprueba si un carácter dado es una de las operaciones binarias básicas.

Dar las especificaciones de los TAD básicos ARBOL_BINARIO[CARACTERES] (0,5 pts) y COLA[CARACTERES] (0,5 pts) y completarlas con las siguientes operaciones:

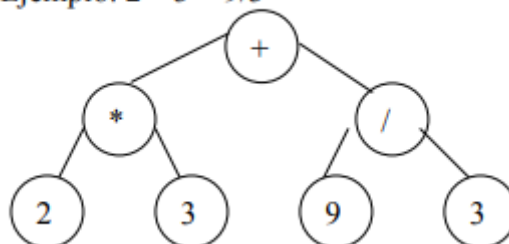
- a) ***exp_correcta***: abb \rightarrow boolean, comprueba que un ABB dado almacena una expresión algebraica correcta, es decir, un nodo que almacena una operación tiene en los hijos los operandos correspondientes y un nodo que almacena un dígito es un nodo hoja.

(1 pto)

- b) ***cola_post***: abb \rightarrow cola, a partir de un árbol binario que almacena una expresión algebraica correcta, devuelve una cola que almacena la expresión en notación postfija.

(1 pto)

Ejemplo: $2 * 3 + 9 / 3$



cola_post $\rightarrow 2\ 3\ *\ 9\ 3\ /\ +$



Ejercicio 2 (1,5 ptos). Dar la especificación del TAD ARBOL_BINARIO[ELEMENTO] y, suponiendo conocida la operación *es_menor*: elemento elemento \rightarrow boolean, ampliarla con las siguientes operaciones (pueden ser parciales), partiendo únicamente de la especificación:

- a) **es_ABB?**: a_bin \rightarrow boolean, que comprueba si un árbol binario dado es un árbol binario de búsqueda. (0,5 ptos)
- b) **es_semicompleto**: a_bin \rightarrow boolean, comprueba si un árbol binario es semicompleto. (0,5 ptos)
- c) **es_montículo_máximos**: a_bin \rightarrow boolean, comprueba si un árbol binario es un montículo de máximos. (0,5 ptos)

Ejercicio 3 (1,5 ptos). Dada la secuencia de enteros: 71,100,29,82,48,39,101,50,46:

- d) Representar gráficamente el **árbol AVL** correspondiente, explicando y realizando los giros que sean necesarios. (0,5 ptos)
- e) Representar gráficamente, paso a paso y explicando las operaciones necesarias, el **montículo de mínimos** resultante. Eliminar el 29 del montículo obtenido, detallando las operaciones necesarias. (0,5 ptos)
- f) Representar gráficamente el **árbol binario de búsqueda** resultante. Borrar el 29 en el árbol creado, de forma que siga siendo un árbol binario de búsqueda. (0,5 ptos)

Ejercicio 4 (4 puntos). El administrador de un sistema informático quiere utilizar el TAD ÁRBOL[CARACTERES] para almacenar y verificar la validez de las claves de acceso al mismo. Dichas claves se almacenan de forma que en raíz del árbol general está el primer carácter (común a todas las claves) y las posibles combinaciones válidas se obtienen recorriendo una de las ramas del mismo hasta llegar a la hoja.

Partiendo de las siguientes operaciones:

- **==** : carácter carácter \rightarrow boolean, que comprueba si dos caracteres son iguales.
- **es_válido**: carácter \rightarrow boolean, que comprueba si un carácter es válido en una clave.
- **sustituir**: carácter \rightarrow carácter, que sustituye un carácter no válido por otro que sí lo es.

Dar las especificaciones ÁRBOL[CARACTERES] (0,5 ptos) y LISTA[CARACTERES] (0,5 ptos) y escribir las siguientes operaciones:

- a) **Calcular el número de claves almacenadas en el sistema.** (1 pto)
- b) **Validar y sustituir**, si es necesario, en todas las **claves** del árbol, los caracteres que forman parte de las mismas. (1,5 ptos)
- c) **Comprobar si una clave dada**, almacenada en una lista de caracteres, **es una clave del sistema**. Se comprobará para ello si está almacenada en una de las ramas del árbol. (1,5 ptos)