

Módulo Imperativo

Práctica Árboles 1

1. Escribir un programa que:

a. Implemente un módulo que almacene información de socios de un club en un árbol binario de búsqueda. De cada socio se debe almacenar número de socio, nombre y edad. La carga finaliza con el número de socio 0 y el árbol debe quedar ordenado por número de socio. La información de cada socio debe generarse aleatoriamente (*nombre se puede leer por teclado*).

b. Una vez generado el árbol, realice módulos independientes que reciban el árbol como parámetro *para*:

- i. Informar los datos de los socios en orden creciente *por número de socio*.
- ii. Informar los datos de los socios en orden decreciente *por número de socio*.
- iii. Informar el número de socio con mayor edad. Debe invocar a un módulo recursivo que retorne dicho valor.
- iv. Aumentar en 1 la edad de los socios con edad impar e informar la cantidad de socios que se les aumentó la edad.
- vi. Leer un nombre e informar si existe o no existe un socio con ese nombre. Debe invocar a un módulo recursivo que reciba el nombre leído y retorne verdadero o falso.
- vii. Informar la cantidad de socios. Debe invocar a un módulo recursivo que retorne dicha cantidad.
- viii. Informar el promedio de edad de los socios. *Debe invocar al módulo recursivo del inciso vii e invocar a un módulo recursivo que retorne la suma de las edades de los socios.*

2. Escribir un programa que:

a. Implemente un módulo que genere aleatoriamente información de ventas de un comercio. Para cada venta generar código de producto, fecha y cantidad de unidades vendidas. Finalizar con el código de producto 0. Un producto puede estar en más de una venta. Se pide:

- i. Generar y retornar un árbol binario de búsqueda de ventas ordenado por código de producto. Los códigos repetidos van a la derecha.
- ii. Generar y retornar otro árbol binario de búsqueda de productos vendidos ordenado por código de producto. Cada nodo del árbol debe contener el código de producto y la cantidad total de unidades vendidas.
- iii. Generar y retornar otro árbol binario de búsqueda de productos vendidos ordenado por código de producto. Cada nodo del árbol debe contener el código de producto y la lista de las ventas realizadas del producto.

Nota: El módulo debe retornar TRES árboles.

b. Implemente un módulo que reciba el árbol generado en i. y una fecha y retorne la cantidad total de productos vendidos en la fecha recibida.

c. Implemente un módulo que reciba el árbol generado en ii. y retorne el código de producto con mayor cantidad total de unidades vendidas.

c. Implemente un módulo que reciba el árbol generado en **iii.** y retorne el código de producto con mayor cantidad de ventas.

3. Implementar un programa que contenga:

a. Un módulo que lea información de los finales rendidos por los alumnos de la Facultad de Informática y los almacene en una estructura de datos. La información que se lee es legajo, código de materia, fecha y nota. La lectura de los alumnos finaliza con legajo 0. La estructura generada debe ser eficiente para la búsqueda por número de legajo y para cada alumno deben guardarse los finales que rindió en una lista.

b. Un módulo que reciba la estructura generada en a. y retorne la cantidad de alumnos con legajo impar.

c. Un módulo que reciba la estructura generada en a. e informe, para cada alumno, su legajo y su cantidad de finales aprobados (nota mayor o igual a 4).

c. Un módulo que reciba la estructura generada en a. y un valor real. Este módulo debe retornar los legajos y promedios de los alumnos cuyo promedio supera el valor ingresado.