

Trabajo Práctico N° 3: **Módulo Imperativo (Árboles 1).**

Ejercicio 1.

Escribir un programa que:

(a) *Implemente un módulo que lea información de socios de un club y las almacene en un árbol binario de búsqueda. De cada socio, se lee número de socio, nombre y edad. La lectura finaliza con el número de socio 0 y el árbol debe quedar ordenado por número de socio.*

(b) *Una vez generado el árbol, realice módulos independientes que reciban el árbol como parámetro y que:*

(i) *Informe el número de socio más grande. Debe invocar a un módulo recursivo que retorne dicho valor.*

(ii) *Informe los datos del socio con el número de socio más chico. Debe invocar a un módulo recursivo que retorne dicho socio.*

(iii) *Informe el número de socio con mayor edad. Debe invocar a un módulo recursivo que retorne dicho valor.*

(iv) *Aumente en 1 la edad de todos los socios.*

(v) *Lea un valor entero e informe si existe o no existe un socio con ese valor. Debe invocar a un módulo recursivo que reciba el valor leído y retorne verdadero o falso.*

(vi) *Lea un nombre e informe 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) *Informe la cantidad de socios. Debe invocar a un módulo recursivo que retorne dicha cantidad.*

(viii) *Informe 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.*

(ix) *Informe, a partir de dos valores que se leen, la cantidad de socios en el árbol cuyo número de socio se encuentra entre los dos valores ingresados. Debe invocar a un módulo recursivo que reciba los dos valores leídos y retorne dicha cantidad.*

(x) *Informe los números de socio en orden creciente.*

(xi) *Informe los números de socio pares en orden decreciente.*

Ejercicio 2.

Escribir un programa que:

(a) Implemente un módulo que lea información de ventas de un comercio. De cada venta, se lee código de producto, fecha y cantidad de unidades vendidas. La lectura finaliza 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.

(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 vendida.

Nota: El módulo debe retornar los dos árboles.

(b) Implemente un módulo que reciba el árbol generado en (i) y un código de producto y retorne la cantidad total de unidades vendidas de ese producto.

(c) Implemente un módulo que reciba el árbol generado en (ii) y un código de producto y retorne la cantidad total de unidades vendidas de ese producto.

Ejercicio 3.

Implementar un programa que contenga:

(a) Un módulo que lea información de alumnos de Taller de Programación y los almacene en una estructura de datos. De cada alumno, se lee legajo, DNI, año de ingreso y los códigos y notas de los finales rendidos. La estructura generada debe ser eficiente para la búsqueda por número de legajo. La lectura de los alumnos finaliza con legajo 0 y, para cada alumno, el ingreso de las materias finaliza con el código de materia -1.

(b) Un módulo que reciba la estructura generada en (a) y retorne los DNI y año de ingreso de aquellos alumnos cuyo legajo sea inferior a un valor ingresado como parámetro.

(c) Un módulo que reciba la estructura generada en (a) y retorne el legajo más grande.

(d) Un módulo que reciba la estructura generada en (a) y retorne el DNI más grande.

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

(f) Un módulo que reciba la estructura generada en (a) y retorne el legajo y el promedio del alumno con mayor promedio.

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