

Módulo Imperativo

Práctica Árboles II

1. Descargar el programa ImperativoEjercicioClase4.pas. y completar.
2. Descargar el programa ImperativoEjercicioClase3.pas de la clase anterior e incorporar lo necesario para:
 - i. Informar el número de socio más grande. Debe invocar a un módulo recursivo que retorne dicho valor.
 - ii. Informar los datos del socio con el número de socio más chico. Debe invocar a un módulo recursivo que retorne dicho socio.
 - iii. Leer un valor entero e informar si existe o no existe un socio con ese valor. Debe invocar a un módulo recursivo que reciba el valor leído y retornar verdadero o falso.
 - iv. Leer *dos valores* e informar la cantidad de socios cuyos códigos se encuentran comprendidos entre los valores leídos. Debe invocar a un módulo recursivo que reciba los valores leídos y retorne la cantidad solicitada.
3. Implementar un programa modularizado para una librería. Implementar módulos para:
 - a. Almacenar los productos vendidos en una estructura eficiente para la búsqueda por código de producto. De cada producto deben quedar almacenados su código, la cantidad total de unidades vendidas y el monto total. De cada venta se lee código de venta, código del producto vendido, cantidad de unidades vendidas y precio unitario. El ingreso de las ventas finaliza cuando se lee el código de venta -1.
 - b. Imprimir el contenido del árbol ordenado por código de producto.
 - c. Retornar el código de producto con mayor cantidad de unidades vendidas.
 - d. Retornar la cantidad de códigos que existen en el árbol que son menores que un valor que se recibe como parámetro.
 - e. Retornar el monto total entre todos los códigos de productos comprendidos entre dos valores recibidos (sin incluir) como parámetros.
4. Una biblioteca nos ha encargado procesar la información de los préstamos realizados durante el año 2021. De cada préstamo se conoce el ISBN del libro, el número de socio, día y mes del préstamo y cantidad de días prestados. Implementar un programa con:
 - a. Un módulo que lea préstamos y retorne 2 estructuras de datos con la información de los préstamos. La lectura de los préstamos finaliza con ISBN 0. Las estructuras deben ser eficientes para buscar por ISBN.
 - i. En una estructura cada préstamo debe estar en un nodo. Los ISBN repetidos insertarlos a la derecha.
 - ii. En otra estructura, cada nodo debe contener todos los préstamos realizados al ISBN. (prestar atención sobre los datos que se almacenan).

- b. Un módulo recursivo que reciba la estructura generada en i. y retorne el ISBN más grande.
- c. Un módulo recursivo que reciba la estructura generada en ii. y retorne el ISBN más pequeño.
- d. Un módulo recursivo que reciba la estructura generada en i. y un número de socio. El módulo debe retornar la cantidad de préstamos realizados a dicho socio.
- e. Un módulo recursivo que reciba la estructura generada en ii. y un número de socio. El módulo debe retornar la cantidad de préstamos realizados a dicho socio.
- f. Un módulo que reciba la estructura generada en i. y retorne una nueva estructura ordenada ISBN, donde cada ISBN aparezca una vez junto a la cantidad total de veces que se prestó.
- g. Un módulo que reciba la estructura generada en ii. y retorne una nueva estructura ordenada ISBN, donde cada ISBN aparezca una vez junto a la cantidad total de veces que se prestó.
- h. Un módulo recursivo que reciba la estructura generada en g. y muestre su contenido.
- i. Un módulo recursivo que reciba la estructura generada en i. y dos valores de ISBN. El módulo debe retornar la cantidad total de préstamos realizados a los ISBN comprendidos entre los dos valores recibidos (incluidos).
- j. Un módulo recursivo que reciba la estructura generada en ii. y dos valores de ISBN. El módulo debe retornar la cantidad total de préstamos realizados a los ISBN comprendidos entre los dos valores recibidos (incluidos).