

Módulo Imperativo

Práctica Adicionales

1. El administrador de un edificio de oficinas tiene la información del pago de las expensas de dichas oficinas. Implementar un programa con:
 - a) Un módulo que retorne un vector, sin orden, con a lo sumo las 300 oficinas que administra. Se deben cargar, para cada oficina, el código de identificación, DNI del propietario y valor de la expensa. La lectura finaliza cuando llega el código de identificación 0.
 - b) Un módulo que reciba el vector retornado en a) y retorne dicho vector ordenado por código de identificación de la oficina. Ordenar el vector aplicando uno de los métodos vistos en la cursada.
 - c) Un módulo que realice una búsqueda dicotómica. Este módulo debe recibir el vector generado en b) y un código de identificación de oficina. En caso de encontrarlo, debe retornar la posición del vector donde se encuentra y en caso contrario debe retornar 0. Luego el programa debe informar el DNI del propietario o un cartel indicando que no se encontró la oficina.
 - d) Un módulo recursivo que retorne el monto total acumulado de las expensas.
2. Una agencia dedicada a la venta de autos ha organizado su stock y, tiene la información de los autos en venta. Implementar un programa que:
 - a) Genere la información de los autos (patente, año de fabricación (2015..2024), marca, color y modelo, finalizando con marca 'MMM') y los almacene en dos estructuras de datos:
 - i. Una estructura eficiente para la búsqueda por patente.
 - ii. Una estructura eficiente para la búsqueda por marca. Para cada marca se deben almacenar juntas las patentes y colores de los autos pertenecientes a ella.
 - b) Invoque a un módulo que reciba la estructura generado en a) i y una marca y retorne la cantidad de autos de dicha marca que posee la agencia.
 - c) Invoque a un módulo que reciba la estructura generado en a) ii y una marca y retorne la cantidad de autos de dicha marca que posee la agencia.
 - d) Invoque a un módulo que reciba el árbol generado en a) i y retorne una estructura con la información de los autos agrupados por año de fabricación.
 - e) Invoque a un módulo que reciba el árbol generado en a) i y una patente y devuelva el modelo del auto con dicha patente.
 - f) Invoque a un módulo que reciba el árbol generado en a) ii y una patente y devuelva el color del auto con dicha patente.
3. PlayStation Store requiere procesar las compras realizadas por sus clientes durante el año 2023.

- a) Implementar un módulo que lea **compras** de videojuegos. De cada compra se lee código del videojuego, código de cliente y mes. La lectura finaliza con el código de cliente 0. *Se sugiere utilizar el módulo leerCompra()*. El módulo debe retornar un árbol binario de búsqueda ordenado por código de videojuego. En el árbol, para cada código de videojuego debe almacenarse una lista con código de cliente y mes perteneciente a cada compra.
- b) Implementar un módulo que reciba el árbol generado en **a)** y un código de videojuego. El módulo debe retornar la lista de las compras de ese videojuego.
- c) Implementar un módulo **recursivo** que reciba la lista generada en **b)** y un mes. El módulo debe retornar la cantidad de clientes que compraron en el mes ingresado.

NOTA: Implementar el programa principal, que invoque a los incisos a, b y c.

```
type
  compra = record
    cod_videojuego : integer;
    cod_cliente : integer;
    mes : integer;
  end;

procedure leerCompra (var c : compra);
begin
  c.cod_cliente := Random(200);
  if (c.cod_cliente <> 0)
  then begin
    c.mes := Random(12) + 1;
    c.cod_videojuego := Random(200) + 1000;
  end;
end
```

4. La Feria del Libro necesita un sistema para obtener estadísticas sobre los libros presentados.

- a) Implementar un módulo que lea información de los libros. De cada libro se conoce: ISBN, código del autor y código del género (1: literario, 2: filosofía, 3: biología, 4: arte, 5: computación, 6: medicina, 7: ingeniería) . La lectura finaliza con el valor 0 para el ISBN. *Se sugiere utilizar el módulo leerLibro()*. El módulo deber retornar **dos** estructuras:
 - i. Un árbol binario de búsqueda ordenado por código de autor. Para cada código de autor debe almacenarse la cantidad de libros correspondientes al código.
 - ii. Un vector que almacene para cada género, el código del género y la cantidad de libros del género.
- b) Implementar un módulo que reciba el vector generado en **a)**, **lo ordene** por cantidad de libros de mayor a menor y retorne el nombre de género con mayor cantidad de libros.
- c) Implementar un módulo que reciba el árbol generado en **a)** y dos códigos. El módulo debe retornar la cantidad total de libros correspondientes a los códigos de autores entre los dos códigos ingresados (incluidos ambos).

NOTA: Implementar el programa principal, que invoque a los incisos a, b y c.

```
type subGenero = 1..7;
  libro = record
    isbn : integer;
    codAutor : integer;
    genero : subGenero;
  end;

var v: array [1..7] of string = ('literario', 'filosofía', 'arte', 'biología', 'computación', 'medicina',
'ingeniería');

procedure leerLibro (var l : libro);
begin
  l.isbn := Random(1000);
  if (l.isbn <> 0) then begin
    l.codAutor := Random(300) + 100;
    l.genero := Random(7) + 1;
  end;
end;
```