**1) Introducción a la Programación II Parcial 25/10/2017**

Se tiene una lista simple de clientes apuntada por ***clientes*** y ordenada por ***nro_cliente***. Cada nodo de esta lista contiene: ***nro_cliente***, un puntero a un árbol ***factura***, un puntero ***sig*** que apunta al siguiente nodo de la lista ordenada por ***nro_cliente***, otro puntero ***punt_num*** también a nodo de esta lista que está inicialmente en ***nil*** y un atributo ***monto_pares*** que está en cero. Cada árbol apuntado por ***factura*** tiene nodos con ***nro_factura*** y ***monto*** y está ordenada por ***nro_factura***.
Se pide actualizar ***punt_num*** de todos los nodos para ordenarlos de menor a mayor según sea la sumatoria de los nodos de cada árbol que están en los niveles pares (el nodo raíz es nivel cero y también debe ser tenido en cuenta), esta suma además debe ser cargada en el atributo ***monto_pares***.
También debe actualizar la variable ***clientes_nivel*** que inicialmente está en ***nil*** y debe quedar apuntando al nodo con menor ***monto_pares***.
Realice el DE, la declaración de tipos, programa principal y todos los módulos que considere necesarios. Considerar que la lista inicialmente está cargada.

---

**2) Introducción a la Programación II Recuperatorio 15/11/2017**

Se tiene un árbol en el que cada nodo posee nro_patente(integer) y anio(integer). Este árbol está ordenado por número de patente. Se pide que dados dos números de patente (pat_min, pat_max) y un número de profundidad (prof_maxima) recorra el árbol y genere una lista doblemente vinculada conteniendo el anio y la cantidad de patentes encontradas. Los nodos de la lista deben estar permanentemente ordenados en forma ascendente por cantidad de patentes. Los nodos del árbol a tener en cuenta son sólo los que poseen el nro_patente entre pat_min y pat_max y cuya profundidad en el árbol no supera prof_maxima (la raíz tiene profundidad = 0). No debe recorrer nodos inútilmente.
Realice el DE, la declaración de tipos, programa principal y todos los módulos que considere necesarios. Considerar que el árbol inicialmente está cargado.

---

**3) Introducción a la Programación II Prefinal 28/11/2017**

Se tienen cargados en el archivo Sucursales.dat los datos referidos a las sucursales de una multinacional. El archivo no se encuentra ordenado bajo ningún criterio. Para cada sucursal se guarda: nombre (de tipo string), facturacion (integer).
Se pide que implemente una función / procedimiento en Pascal que genere un árbol ARB_SUCURSALES en memoria principal con todos los datos del archivo teniendo en cuenta lo siguiente:

- El árbol debe estar ordenado alfabéticamente por nombre de sucursal.
- Si ingresa una sucursal con un nombre que ya fue incorporado al árbol, no debe incorporarse.
- Los nodos del árbol deben tener un vínculo adicional de tal manera que las sucursales puedan ser listadas ascendentemente por facturación mediante ese vínculo. Ese nuevo orden es accedido por un puntero inicial ORDEN_FACTURACION

**NOTA:** El árbol debe estar vinculado y ordenado correctamente por facturación en todo momento (ante cada nueva sucursal que se agrega). No puede usar estructuras auxiliares. Debe definir: diagrama de estructura, programa principal, procedimientos y funciones, constantes, tipos y variables.

4) Se tiene un árbol binario en donde se guardan los nombres de los empleados de una empresa en orden alfabético. Cada nodo 'Empleado' tiene además un puntero a su jefe inmediato. Ningún empleado tiene más de un jefe, y pueden existir empleados que no tengan ningún superior. Se

tiene además de un Archivo, donde cada registro contiene el nombre de un empleado y el nombre de su jefe.
Notas: el árbol existe y puede tener algunos jefes cargados, si es así los datos del árbol tienen mayor prioridad. El archivo existe y tiene los datos cargados. No debe utilizar estructuras auxiliares. La inexistencia de algunos de los nombres del archivo en el árbol debe ser informada mediante un mensaje y no dar de alta nodos en el árbol
Declarar las estructuras y realizar los siguientes algoritmos:

1. Realice un procedimiento/función que agregue a la estructura dada, los punteros a los jefes indicados por el archivo.

Realice un procedimiento/función que dado el nombre de un empleado impprima su jefe y sus subordinados