

ESTRUCTURA DE DATOS Y ALGORITMOS  
UTU, Buceo

Turno nocturno - Segundo Parcial 30/11/2010

Escriba nombre y cédula de identidad en todas las hojas. El parcial suma 60 puntos.

**Ejercicio 1 (18 puntos)**

Considere la siguiente definición del tipo LISTA, de listas de enteros en memoria dinámica:

```
struct nodoLista {
    int info;
    nodoLista *sig;
};
typedef nodoLista *LISTA;
```

Se pide, sin usar funciones o procedimientos auxiliares, implementar la siguiente función iterativa:

*InsComFin*: Dada una lista L de enteros de tipo LISTA y dado un entero x, retorna una nueva lista (que no comparte registros de memoria con la lista parámetro) que contiene a todos los elementos de L en orden inverso y a x como primer y último elemento. La función no debe recorrer L más de una vez y la lista resultado no debe recorrerse.

Ejemplos:

Entradas	Resultado de <i>InsComFin</i>
L = [2,3,7,8], x = 4	[4,8,7,3,2,4]
L = [4,8], x = 8	[8,8,4,8]
L = [4], x = 4	[4,4,4]
L = [], x = 4	[4,4]

**Ejercicio 2 (30 puntos)**

Considere la siguiente definición del tipo ABB de los árboles binarios de búsqueda de enteros, en memoria dinámica:

```
struct nodoABB {
    int dato;
    nodoABB * izq; // menores que dato
    nodoABB * der; // mayores que dato
};
typedef nodoABB *ABB;
```

a) Defina una función recursiva *BorrarMin* que dado un árbol binario de búsqueda de enteros A de tipo ABB, no vacío (precondición), elimine y retorne de A su mínimo elemento. No se pueden usar funciones o procedimientos auxiliares. *BorrarMin* debe evitar

recorrer nodos innecesarios de A. Deberá liberarse la memoria de la celda cuyo elemento sea eliminado. A luego de la función debe ser un árbol binario de búsqueda.

*int BorrarMin (ABB & A)*

b) Defina un procedimiento *Borrar* que dado un árbol binario de búsqueda de enteros A de tipo ABB, no vacío (precondición), elimine el entero que se encuentra en la raíz de A. No se pueden usar funciones o procedimientos auxiliares, con excepción de la función definida en la parte (a). *Borrar* no debe recorrer A, aunque si puede hacerlo eventualmente *BorrarMin*. A luego del procedimiento debe ser un árbol binario de búsqueda.

*void Borrar (ABB & A)*

### **Ejercicio 3 (12 puntos)**

Defina un orden de recorrida de un árbol binario de búsqueda, de modo que al utilizar ese orden para imprimir el árbol, los valores se impriman de mayor a menor. Escriba la función correspondiente.