

Problema 1 (15 puntos)

Considere la siguiente definición de listas de enteros de memoria dinámica:

```
typedef nodoLista* Lista
struct nodoLista{
    int dato;
    Lista sig;
}
```

a) Implemente una función iterativa `Lista union(Lista l1, Lista l2)` que dadas dos listas de enteros ordenadas de menor a mayor y sin elementos repetidos, construya y retorne una nueva lista ordenada de menor a mayor que contenga todos los elementos que pertenecen a l1 y a l2, pero sin elementos repetidos. La lista resultado no debe compartir memoria con las listas parámetro, que no pueden modificarse. La función debe tener $O(n+m)$ peor caso, con n y m los largos de las listas parámetro. No utilice funciones o procedimientos auxiliares, ni estructuras de datos adicionales como arreglos/vectores. Por ejemplo, si las listas fueran [1,8,32,45] y [12,23,32,47], el resultado debería ser: [1,8,12,23,32,45,47].

b) Justifique muy brevemente el cumplimiento del orden exigido en la parte a) para su implementación de **union**.

Problema 2 (15 puntos)

Considere la definición del tipo **ABB** de árboles binarios de búsqueda de enteros, en memoria dinámica; y la definición de listas de enteros de memoria dinámica

```
typedef nodoABB* ABB;
struct nodoABB{
    int dato;
    ABB izq, der;
}

typedef nodoLista* Lista;
struct nodoLista{
    int dato;
    Lista sig;
}
```

a) Implemente una función recursiva `Lista menores(ABB t, int x)` que, dado un árbol "t" de tipo **ABB**, y un entero "x" de tipo **int**, retorne una lista de tipo **Lista** que contenga todos los elementos del árbol que sean menores que "x". La lista resultado debe estar ordenada de menor a mayor. Si todos los elementos del árbol son mayores o iguales a "x", el resultado deberá ser la lista vacía.

Debe tenerse en cuenta que "x" es un valor cualquiera que no necesariamente pertenece al árbol.

Para la implementación de la solución, asuma que los siguientes métodos ya están implementados

```
Lista preorden(ABB t) //Retorna la lista con los elementos del árbol
//visitados en preorden
Lista entreorden(ABB t) //Retorna la lista con los elementos del árbol
//visitados en entreorden
Lista posorden(ABB t) //Retorna la lista con los elementos del árbol
//visitados en posorden
```

b) Indique el orden de tiempo de ejecución en el peor caso y en el caso promedio de la función **menores** Explique brevemente el peor caso.