

Problema 1 (15 puntos)

Implemente un procedimiento iterativo *ultimoPrimero* en C++ que dada una lista de enteros l , elimine de l su último elemento y lo inserte al comienzo de l . Si la lista tiene largo 0 o 1, el procedimiento no tendrá efecto.

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

```
void ultimoPrimero (Lista & l)
```

Ejemplo: $l=[2,6,3,9,7]$ --ultimoPrimero--> $l=[7,2,6,3,9]$

¿Cuál es el orden de tempo de ejecución en el peor caso de *ultimoPrimero*? Justifique brevemente.

Problema 2 (15 puntos)

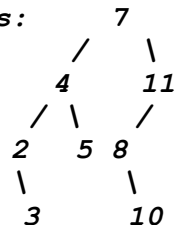
Considere la siguiente definición del tipo *ABB* de los árboles binarios de búsqueda de enteros, en memoria dinámica:

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

Implemente una función recursiva *sucesor* que, dado un *ABB* t y un entero x , retorne el mínimo elemento mayor estricto que x en t (el sucesor de x en t). Si no hay elementos mayores que x en t , en particular si t es vacío, la función deberá retornar el valor `INT_MAX`. La función debe tener $O(n)$ de tiempo de ejecución en el peor caso, siendo n la cantidad de nodos en t , aunque debe evitarse recorrer nodos de t que no sean necesarios.

```
int sucesor (ABB t, int x)
```

Ejemplo: Si t es:



x es 7, el resultado es: 8

x es 1, el resultado es: 2

x es 12, el resultado es: `INT_MAX`