

Problema 1 (15 puntos)

Considere la siguiente definición del tipo **Lista** para listas de enteros:

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

Implemente la función iterativa **unicos** que dada una lista *l* de tipo **Lista** que puede contener valores exclusivamente en el rango $[0 : n]$ (entre 0 y *n* inclusive, con $n > 0$), retorne *true* si y solo si la lista no tiene elementos repetidos. Si la lista es vacía, el resultado debe ser *true*. La función debe ser $O(\max(n,m))$, siendo *m* el largo de *l*.

PRE: Cada elemento x de la lista l cumple: $0 \leq x \leq n$, con $n > 0$
bool unicos(Lista l, int n)

Problema 2 (15 puntos)

Considere la siguiente definición para árboles binario de enteros no negativos (**AB**):

```
struct nodoAB{
    unsigned int dato;
    nodoAB *izq, *der;
}
typedef struct nodoAB * AB;
```

Implemente una función recursiva **nivelDato** que, dado un árbol binario *t* de tipo **AB** y dado un entero *x*, retorne el nivel en *t* en el que se encuentra *x*. Si *x* no está en *t* (en particular si *t* es NULL), el resultado debe ser -1. Asumimos que *t* no tiene elementos repetidos, aunque no puede asumirse que los elementos en *t* están ordenados como en un árbol binario de búsqueda. No defina operaciones auxiliares para implementar **nivelDato**, función que no deberá recorrer el árbol *t* más de una vez y su orden en el peor caso debe ser $O(n)$, siendo *n* la cantidad de nodos de *t*.

PRE: t no tiene elementos repetidos
int nivelDato (AB t, int x)