



Examen Final

1. Se define un **árbol binario coloreado** como aquél en el cual cada nodo tiene la propiedad de ser azul o rojo. Un árbol binario coloreado es azul si la mayor parte de sus nodos son azules, rojo si la mayor parte de sus nodos son rojos o neutro si tiene iguales cantidades. Una hoja es pura si todos sus ancestros son de su mismo color. Considere la siguiente declaración:

```
typedef enum color {Rojo, Azul, Neutro} Color;

typedef struct Nodo_ {
    char* cdato;
    Color colorNodo;
    struct Nodo_ *der, *izq;
} Nodo;

typedef Nodo* ArbolBinarioColoreado;
```

Se pide que escriba las siguientes funciones:

```
//Retorna el color del Arbol.
Color color(ArbolBinarioColoreado inicio);

//Retorna la cantidad de hojas puras.
int CantidadPuras(ArbolBinarioColoreado inicio);
```

2. Suponga que se tiene la siguiente tabla hash que contiene números enteros, implementada usando sondeo lineal como forma de resolver las colisiones (es decir, se admite sólo una clave por casilla, y en caso de colisión se busca en las casillas siguientes un lugar disponible). La función hash usada es la siguiente función: $h(x) = x \bmod 9$.

0	1	2	3	4	5	6	7	8
9	18		12	3	14	4	21	

- a) En qué orden pueden haber sido agregado los elementos a la tabla? Suponga que sólo se realizaron operaciones de inserción. Indique todas las opciones correctas.
- 1) 9, 14, 4, 18, 12, 3, 21
 - 2) 12, 3, 14, 18, 4, 9, 21
 - 3) 12, 14, 3, 9, 4, 18, 21
 - 4) 9, 12, 14, 3, 4, 21, 18
 - 5) 12, 9, 18, 3, 14, 21, 4
- b) Escriba el algoritmo para buscar un elemento en la tabla.
- c) Explique qué problema existe al remover al 3 de la tabla, y una forma de resolverlo sin modificar el algoritmo del apartado anterior.
- d) Dibuje cómo quedaría la tabla luego de remover el 3, usando la solución descripta.