

Ejercicios

Utilizando la siguiente estructura:

```
typedef struct nodo {  
    int valor;  
    struct nodo *next;  
}*Nodo;
```

1.- Implemente una función que reciba un nodo y lo agregue **ENTRE** las posiciones donde el nodo anterior sea menor o igual y el nodo siguiente sea mayor al “**valor**” definido dentro de la estructura nodo.

Obs: En el caso que el Nodo nuevo sea menor al nodo inicial se enlaza al principio.

Obs: En el caso que el Nodo nuevo sea mayor al nodo final se enlaza en la última posición.

Nodo agregarEnOrden(Nodo lista, Nodo nuevo)

Ejemplo 1:

Nodo a Ingresar: 7

Nodo Lista: 1 -> 3 -> 8 -> 10 -> NULL

Resultado: 1 -> 3 -> **7** -> 8 -> 10 -> NULL

Ejemplo 2:

Nodo a Ingresar: 9

Nodo Lista: 1 -> 3 -> 8 -> 10 -> NULL

Resultado: 1 -> 3 -> 8 -> **9** -> 10 -> NULL

Ejemplo 3:

Nodo a Ingresar: 20

Nodo Lista: 1 -> 3 -> 8 -> 10

Resultado: 1 -> 3 -> 8 -> 10 -> **20** -> NULL

Ejemplo 4:

Nodo a Ingresar: 5

Nodo Lista: NULL

Resultado: 5 -> NULL

2.- Implemente una función que reciba una lista enlazada y elimine todos los nodos en donde **eliminar** sea igual al contenido de **valor (Declarado en estructura)**.

Nodo eliminarTodosLosNodosQueContenga(Nodo lista, int eliminar)

Ejemplo 1:

eliminar: 10

Nodo Lista: 12 -> 20 -> 10 -> 10 -> 23 -> 30 -> 10 -> NULL

Resultado: 12 -> 20 -> 23 -> 30 -> NULL

Ejemplo 2:

eliminar: 5

Nodo Lista: 5 -> 5 -> 5 -> 5 -> 5 -> NULL

Resultado: NULL

3.- Implemente una función que reciba dos listas e indique a través de un **0(false)** o **1(true)** si la lista B está contenida dentro de la lista A.

Obs: Se dice que está contenida cuando todos los elementos de B se encuentran en A

int esContenido(Nodo listaA, Nodo listaB)

Ejemplo 1:

Lista A : 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> 7 -> NULL

Lista B : 3 -> 4 -> 5 -> NULL

Resultado: 1

Ejemplo 2:

Lista A: 1-> 2 -> 3 -> 4 -> 5 -> NULL

Lista B: 2 -> 3 -> 5 -> NULL

Resultado: 0

Ejemplo 3:

Lista A: NULL

Lista B: NULL

Resultado: 1

4.- Implemente una función que reciba una lista y retorne la misma lista pero invertida, donde el nodo final es el inicial y el inicial es ahora el final.

Nodo invertirLista(Nodo lista)

Ejemplo 1:

Nodo Lista: 1 -> 2 -> 3 -> 4 -> 5 -> NULL

Resultado: 5 -> 4 -> 3 -> 2 -> 1 -> NULL

Ejemplo 2:

Nodo Lista: NULL

Resultado: NULL

Ejemplo 3:

Nodo Lista: 4 -> 2 -> 1 -> NULL

Resultado: 1 -> 2 -> 4 -> NULL