

Práctica 2: Operaciones con listas enlazadas

En pseudocódigo, crea las siguientes funciones:

1. Crear una función que permita insertar un elemento al final de la lista ya creada.

```
Void insertar_final(int numero){
    Nodo nodo = new Nodo();
    Nodo auxiliar = head;
    if(! head){ // si esta vacia la lista
        Head = nodo;
    }else{
        while(auxiliar diferente de nulo){
            Auxiliar = auxiliar -> next;
        }
        Auxiliar -> next = nodo;
    }
}
```

2. Crear una función que permita insertar un elemento inmediatamente después de un número que se encuentre en la lista.

```
Void insertar_ordenada(int numero){
    Nodo nodo = new Nodo();
    Nodo auxiliar = head;
    if(! head){ // si esta vacia la lista
        Head = nodo;
    }else{
        while(auxiliar diferente de nulo){
            Auxiliar = auxiliar -> next;
        }
        Auxiliar -> next = nodo;
    }
}
```

3. Crear una función que para una lista dada, elimine el nodo que tenga el mayor.

```
Void remove_mayor(){
    Nodo aux=head;
    Nodo aux2 = head -> siguiente;
    if(! head){
        cout<<"no hay nada";
    }else{
        while(auxiliar diferente de nulo){
            if(aux > aux2){
                Nodo mayor = aux;
            }
        }
    }
}
```

```

    }
    Delete aux;
}
end

```

4. Escribe función que das dos listas $L1=\{x_1, x_2, \dots, x_n\}$ y $L2=\{y_1, y_2, \dots, y_m\}$, obtenga una lista Z mezcla de ambas. Esto es, $Z = \{x_1, y_1, x_2, y_2, \dots, x_m, y_m, x_{m+1}, x_{m+2}, \dots, x_n\}$.

```

Void mezcla(L1, L2, ){
    crear_lista(Z, int tamaño);
    for(iterar L1){ //la iteracion debe ser de manera que sea de forma intercalada
        for(iterar L2){
            Metemos valores en Z;
        }
    }
    mostrar_lista(Z, tamaño);
}
end

```

5. Escribe una función que, das dos listas ordenadas de menor a mayor, devuelva otra lista con todos los elementos de las dos listas originales y ordenada también de menor a mayor.