# Lista de Exercícios Avançados — Listas Lineares Encadeadas

## 1. Inserção Ordenada em Lista Encadeada

Implemente uma função que insira um elemento em uma lista encadeada de forma ordenada, mantendo a ordem crescente dos elementos. A função deve garantir que duplicatas não sejam inseridas.  
  
Exemplo 1:  
Entrada: [2, 5, 7], elemento = 6  
Saída Esperada: [2, 5, 6, 7]  
  
Exemplo 2:  
Entrada: [1, 3, 4], elemento = 3  
Saída Esperada: [1, 3, 4]

## 2. Remoção de Elementos Duplicados

Desenvolva um algoritmo que remova todos os elementos duplicados de uma lista encadeada, mantendo apenas a primeira ocorrência de cada elemento.  
  
Exemplo 1:  
Entrada: [4, 2, 4, 3, 2, 5]  
Saída Esperada: [4, 2, 3, 5]  
  
Exemplo 2:  
Entrada: [1, 1, 1, 2, 3]  
Saída Esperada: [1, 2, 3]

## 3. Inversão Parcial de Lista

Implemente uma função que inverta parcialmente uma lista encadeada entre duas posições dadas (início e fim). Utilize manipulação de ponteiros para realizar a inversão.  
  
Exemplo 1:  
Entrada: [1, 2, 3, 4, 5], início = 2, fim = 4  
Saída Esperada: [1, 4, 3, 2, 5]  
  
Exemplo 2:  
Entrada: [10, 20, 30, 40], início = 1, fim = 3  
Saída Esperada: [30, 20, 10, 40]

## 4. Interseção de Duas Listas Encadeadas

Crie uma função que receba duas listas encadeadas e retorne uma nova lista contendo os elementos presentes em ambas (sem duplicatas).  
  
Exemplo 1:  
Entrada: L1 = [1, 2, 3, 4], L2 = [3, 4, 5, 6]  
Saída Esperada: [3, 4]  
  
Exemplo 2:  
Entrada: L1 = [7, 8, 9], L2 = [10, 8, 11]  
Saída Esperada: [8]

## 5. União de Listas Ordenadas sem Duplicação

Implemente uma função que receba duas listas encadeadas ordenadas e retorne uma nova lista com a união ordenada e sem duplicações.  
  
Exemplo 1:  
Entrada: L1 = [1, 3, 5], L2 = [2, 3, 4, 6]  
Saída Esperada: [1, 2, 3, 4, 5, 6]  
  
Exemplo 2:  
Entrada: L1 = [10, 20, 30], L2 = [15, 20, 25]  
Saída Esperada: [10, 15, 20, 25, 30]

## 6. Remover Elementos Maiores que um Limite

Crie uma função que remova da lista todos os elementos cujo valor seja superior a um limite informado.  
  
Exemplo 1:  
Entrada: [5, 10, 15, 20], limite = 12  
Saída Esperada: [5, 10]  
  
Exemplo 2:  
Entrada: [8, 3, 17, 1], limite = 5  
Saída Esperada: [3, 1]

## 7. Mesclagem Alternada de Duas Listas

Implemente uma função que mescle duas listas encadeadas intercalando seus elementos. Caso uma lista seja maior, os elementos restantes devem ser adicionados ao final.  
  
Exemplo 1:  
Entrada: L1 = [1, 2, 3], L2 = [4, 5, 6, 7]  
Saída Esperada: [1, 4, 2, 5, 3, 6, 7]  
  
Exemplo 2:  
Entrada: L1 = [10, 20], L2 = [30, 40, 50]  
Saída Esperada: [10, 30, 20, 40, 50]

## 8. Detectar Ciclo em uma Lista Encadeada

Desenvolva um algoritmo que detecte se há um ciclo (loop) em uma lista encadeada. Caso exista, retorne o nó onde o ciclo se inicia.  
  
Exemplo 1:  
Entrada: Lista com ciclo em nó de valor 3  
Saída Esperada: 'Ciclo detectado no nó 3'  
  
Exemplo 2:  
Entrada: Lista sem ciclo [1, 2, 3, 4]  
Saída Esperada: 'Nenhum ciclo detectado'

## 9. Comparação Estrutural e de Conteúdo

Implemente uma função que verifique se duas listas encadeadas são idênticas em estrutura e conteúdo.  
  
Exemplo 1:  
Entrada: L1 = [1, 2, 3], L2 = [1, 2, 3]  
Saída Esperada: Verdadeiro  
  
Exemplo 2:  
Entrada: L1 = [1, 2, 3], L2 = [3, 2, 1]  
Saída Esperada: Falso

## 10. Particionar Lista em Duas Sublistas

Implemente um algoritmo que divida uma lista encadeada em duas sublistas: uma contendo elementos pares e outra contendo ímpares.  
  
Exemplo 1:  
Entrada: [1, 2, 3, 4, 5]  
Saída Esperada: ListaPar = [2, 4], ListaImpar = [1, 3, 5]  
  
Exemplo 2:  
Entrada: [8, 7, 6, 5]  
Saída Esperada: ListaPar = [8, 6], ListaImpar = [7, 5]