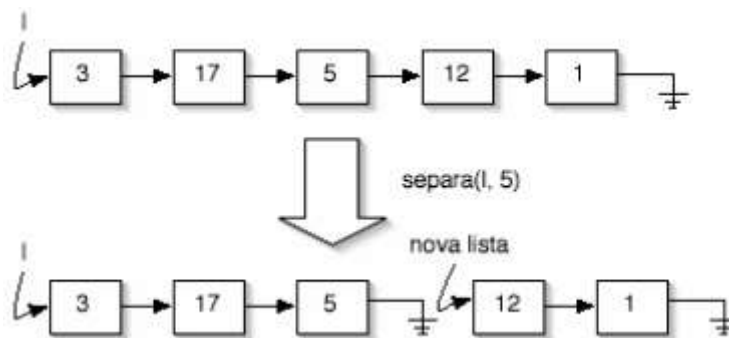


# Listas Encadeadas

## Lista de Exercícios

1. Escreva uma função que conte o número de células de uma lista encadeada. Faça duas versões: uma iterativa e uma recursiva.
2. A altura de uma célula *c* em uma lista encadeada é a distância entre *c* e o fim da lista. Mais precisamente, a altura de *c* é o número de passos do caminho que leva de *c* até a última célula da lista. Escreva uma função que calcule a altura de uma dada célula.
3. A profundidade de uma célula *c* em uma lista encadeada é o número de passos do único caminho que vai da primeira célula da lista até *c*. Escreva uma função que calcule a profundidade de uma dada célula.
4. Considerando listas de valores inteiros, implemente uma função que receba como parâmetro uma lista encadeada e um valor inteiro *n*, e divida a lista em duas, de tal forma que a segunda lista comece no primeiro nó logo após a primeira ocorrência de *n* na lista original. A figura a seguir ilustra essa separação:



Essa função deve obedecer ao protótipo:

**Lista\* separa (Lista\* l, int n);**

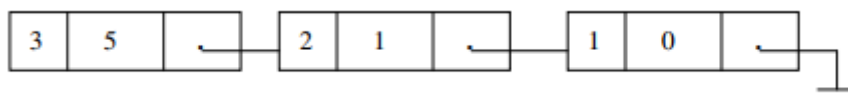
A função deve retornar um ponteiro para a segunda subdivisão da lista original, enquanto *l* deve continuar apontando para o primeiro elemento da primeira subdivisão da lista.

5. Seja uma lista encadeada que contém números inteiros. Faça as seguintes funções:
  - a) Verifique se a lista encadeada está em ordem crescente.
  - b) Encontre todas as células que contenham o menor número da lista.
6. Escreva uma função que verifique se duas listas encadeadas têm o mesmo conteúdo (não necessariamente na mesma ordem).
7. Escreva uma função que receba uma lista encadeada e devolva o endereço de uma célula que esteja o mais próximo possível do meio da lista. Faça isso sem contar explicitamente o número de células da lista.

8. Escreva uma função que concatene duas listas encadeadas (isto é, engate a segunda no fim da primeira).
9. Escreva uma função que insira em uma lista encadeada uma nova célula com conteúdo  $x$  imediatamente depois da  $k$ -ésima célula.
10. Escreva uma função que troque de posição duas células de uma mesma lista encadeada.
11. Escreva uma função que inverta a ordem das células de uma lista encadeada (a primeira passa a ser a última, a segunda passa a ser a penúltima etc.). Faça isso sem usar espaço auxiliar, apenas alterando ponteiros.
12. Escreva uma função para remover de uma lista encadeada contendo números inteiros, todas as células que contêm o número  $y$  passado como parâmetro.
13. Escrever uma função que, dado uma lista encadeada  $L$  desordenada, cria uma lista  $K$  ordenada, com os mesmos nós da lista  $L$ . A função remove os elementos da lista  $L$ , sempre do maior para o menor, e insere-os no início da lista  $K$ , que dessa forma torna-se uma lista ordenada (em ordem crescente). Obs: não devem ser criados nós extras, vocês devem utilizar os mesmos nós alocados para a lista  $L$ . No final do processo, a lista  $L$  estará vazia e a lista  $K$  conterá os nós anteriormente alocados para a lista  $L$ .
14. Escrever uma função para remover elementos repetidos de uma lista encadeada.
15. Escrever um procedimento para fazer a fusão de duas listas encadeadas ordenadas, mantendo-a ordenada. Não devem ser alocados (criados) nós extras. Os nós serão religados para compor a nova lista ordenada.

16. Polinômios podem ser representados por meio de listas, cujos nós são registros com 3 campos: coeficiente, expoente e referência ao seguinte. Por exemplo, o polinômio  $3x^5 + 2x - 1$

seria representado por:



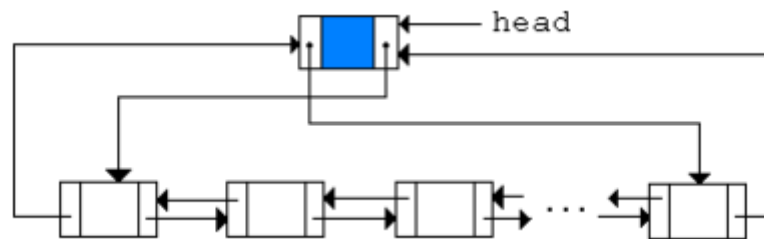
17. Usando essa representação, escreva funções para:
  - a. Criar as listas de polinômios, inserindo os elementos em ordem decrescente pelo expoente do polinômio. (Inserção em lista ordenada).
  - b. Somar polinômios. A função recebe os ponteiros para o polinômio  $P1$  e  $P2$  e cria a lista  $S$ , a qual representa a soma dos polinômios  $P1$  e  $P2$ . Protótipo da função:
 

**Lista \* SomaPolinomio(lista \* P1, lista \* P2);**

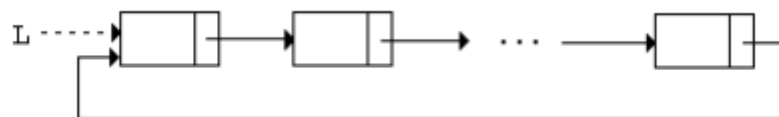
 Exemplos de soma entre dois polinômios:  $(4x^2 - 10x - 5) + (6x + 12) = 4x^2 - 4x + 7$
  - c. Escrever um programa principal que leia os polinômios, crie as correspondentes listas representando-os e faça as chamadas à função soma e imprima o resultado.
18. Faça um programa em C que resolva o problema da Torre de Hanoi, sem usar recursividade. O programa deverá usar uma lista que implementa a pilha do programa para simular as chamadas recursivas.

19. Refaça os exercícios 13, 14 e 15 para listas duplamente encadeadas.

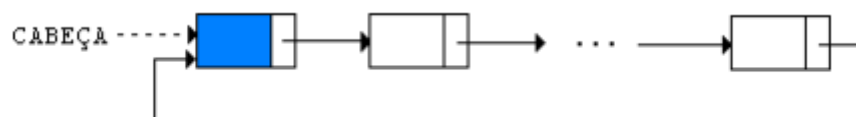
20. Uma lista duplamente encadeada possui registros que tem ligações com o sucessor e o predecessor na lista. Ainda, é usual ter um “nó head”, que é um registro auxiliar que aponta para o “primeiro” e o “último” registro da lista e é apontado por eles (ver figura abaixo). Construa um conjunto de procedimentos para busca, inserção e eliminação de elementos nessa lista.



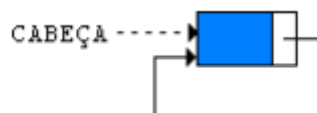
21. Uma lista encadeada circular é uma lista encadeada cujo último elemento aponta para o primeiro:



Vantagem: cada elemento é acessível a partir de qualquer outro. Numa lista circular, não faz mais sentido se falar em primeiro ou último elemento. Porém, devemos saber, durante um percurso na lista, se já demos uma volta completa, para evitarmos loops infinitos. Para isso, assumimos a existência de um registro especial, chamado Cabeça de Lista, cujo campo de informação não pertence ao conjunto de elementos da lista (poderia até servir de sentinela numa busca):



Situação lista circular vazia:



Construa algoritmos para:

- Contar o número de elementos numa lista circular;
- Inserir um elemento à esquerda da cabeça da lista;
- Eliminar o elemento de valor  $x$ ;
- Concatenar duas listas circulares;

- e. Intercalar duas listas ordenadas;
  - f. Fazer uma cópia da lista;
22. Generalize a lista circular do exercício anterior para Lista Circular Duplamente Encadeada, e repita os itens (a) até (f).
23. Considere um conjunto implementado através de listas encadeadas. Escreva os seguintes operadores:
- a) Conjunto União (Conjunto B)
  - b) Conjunto Intersecção (Conjunto B)
  - c) Conjunto Diferença (Conjunto B)
24. Considere uma lista encadeada em que cada elemento armazena um caractere. Escreva o método:
- TrocaPalavra (ListaEnc P, ListaEnc Q)**
- que troca todas as ocorrências da palavra P por Q na lista.
25. Considere o seguinte método para a criptografar mensagens secretas:
- **1ª etapa:** inverter as sequências de não vogais, incluindo espaços e pontuação.
  - **2ª etapa:** inverter a mensagem resultante.

Exemplo:

- Dada a mensagem: **ESTRUTURAS DE DADOS É MUITO LEGAL.**
- Após a primeira etapa, teremos: **ERTSUTURAD SED ADO SÉM UITOL EGA.L**
- E depois da segunda etapa teremos: **L.AGE LOTIU MÉS ODA DES DARUTUSTRE**

Dada uma lista linear encadeada, na qual cada posição contém um caractere da mensagem codificada, faça um programa que obtenha a mensagem original na mesma lista.