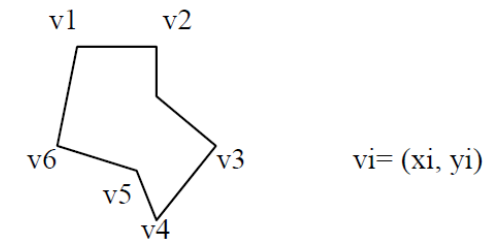


## Exercícios: Listas

1. O que significa alocação sequencial de memória para um conjunto de elementos?
2. O que significa alocação estática de memória para um conjunto de elementos?
3. Qual a diferença entre alocação seqüencial e alocação encadeada?
4. Quais as vantagens de se utilizar alocação encadeada para um conjunto de elementos? Quais as possíveis desvantagens?
5. Qual a diferença entre alocação estática e alocação dinâmica?
6. Faça uma função para concatenar duas listas em uma terceira lista que deverá ser retornada pela função. Não use funções já implementadas.
7. Faça uma função para remover os  $n$  primeiros elementos de uma lista. A função deve retornar se a operação foi possível ou não.
8. Faça uma função para buscar a posição e o valor do maior elemento da lista, os quais deverão ser retornados por referência. A função deve retornar se a operação foi possível ou não.
9. Faça uma função para trocar de posição dois elementos de uma lista. A função deve retornar se a operação foi possível ou não.
10. Considere que a **struct** abaixo está armazenada na sua lista. Faça uma função para buscar o produto de menor preço. A função deve retornar se a operação foi possível ou não.

```
struct produto{  
    int codProd; //código do produto  
    char nomeProd[10]; //nome do produto  
    float valor; //valor do produto  
    int qtdeEstoque; //quantidade disponível em estoque  
}
```

11. Considere uma lista contendo números inteiros positivos. Faça uma função que retorne quantos números pares existem na lista.
12. Considere uma lista contendo números inteiros positivos. Faça uma função que retorne uma nova lista contendo apenas os números pares da lista.
13. Considere uma lista contendo números inteiros positivos. Faça uma função que retorne a média da lista.
14. Um polígono geométrico pode ser considerado como uma lista de vértices. Para desenhar tal polígono, basta traçar uma reta ligando vértices adjacentes. Projete um TAD para armazenar esse polígono (defina as operações necessárias). Considere a necessidade de desenhar um polígono, eliminar arestas e incluir novas arestas



15. Fazer uma função para verificar se uma lista está ordenada ou não (a ordem pode ser crescente ou decrescente).
16. Fazer uma função que copia uma lista L1 em uma outra lista L2.
17. Fazer uma função que copia uma lista L1 em outra L2, eliminando os elementos repetidos.
18. Fazer uma função para inverter uma lista L1 colocando o resultado em L2.
19. Fazer uma função que realiza o merge de duas listas ordenadas.
20. Considerando listas de valores inteiros, implemente uma função que receba como parâmetro uma lista e um valor  $n$  e divida a lista em duas, de tal forma que a segunda lista começa no primeiro nó após o valor  $n$ . A função deve retornar a segunda lista.
21. Escreva uma função que crie uma lista encadeada a partir de um vetor.
22. Escreva uma função que copie uma lista encadeada para um vetor.
23. Escreva uma função que concatena duas listas encadeadas (isto é, “amarra” a segunda no fim da primeira) e retorna um ponteiro para a lista concatenada. As listas originais não devem ser alteradas.
24. Escreva uma função que verifica se duas listas dadas são iguais (mesmo conteúdo)
25. Escreva uma função que inverte a ordem das células de uma lista encadeada (a primeira passa a ser última, segunda passa a ser penúltima etc.). Faça isso sem usar espaço auxiliar; apenas altere os ponteiros.
26. Dadas duas listas encadeadas e dinâmicas L1 e L2, implemente a operação UNION, que cria uma terceira lista L3 com a união entre as duas listas.
27. Dadas duas listas encadeadas e dinâmicas L1 e L2, sem elementos repetidos, implemente a operação INTER, que cria uma terceira lista L3 com a intersecção entre as duas listas, também sem elementos repetidos
28. Seja uma lista dinâmica e encadeada com elementos repetidos, implemente uma rotina para eliminar todas as ocorrências de X da lista.
29. Implemente uma rotina recursiva para calcular o tamanho de uma lista dinâmica e encadeada.
30. Implemente uma função não recursiva para verificar se duas listas dinâmicas e encadeadas são iguais.
31. Implemente uma função recursiva para verificar se duas listas dinâmicas e encadeadas são iguais.

32. Faça uma função recursiva para imprimir uma lista encadeada
33. Seja uma lista encadeada cujos registros possuam informação do tipo inteiro. Escreva um programa que ordena a lista em ordem crescente em relação à informação.
34. Dada uma lista encadeada que armazena números inteiros, escreva uma função que transforma a lista dada em duas listas encadeadas: a primeira contendo os elementos cujo conteúdo é par e a segunda contendo os elementos com conteúdos ímpares
35. Faça uma função para intercalar listas: a função recebe as duas listas ordenadas e retorna a lista com os elementos das duas listas intercalados conforme a ordem com que elas se dispõem na lista.
36. Faça uma função para intercalar listas ordenadamente: a função recebe as duas listas ordenadas e retorna a lista com os elementos das duas listas intercalados, porém obedecendo a ordem crescente dos números.
37. Dada uma lista encadeada que armazena números inteiros, escreva uma função que transforma a lista dada em duas listas encadeadas: a primeira contendo os elementos cujo conteúdo é par e a segunda contendo os elementos com conteúdos ímpares. Sua função deve manipular somente os apontadores e não o conteúdo das células.
38. Implementar uma lista encadeada para manipulação de polinômios do tipo

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x^1 + a_0$$

onde cada elemento  $i$  da lista deve armazenar o  $n$ -ésimo termo do polinômio (diferente de 0). Cada item da lista deve conter o valor da potência de (inteiro) e o coeficiente correspondente (inteiro). Faça funções para

- (a) Inicializar um polinômio.
  - (b) Inserir um novo termo no polinômio existente
  - (c) Imprimir
  - (d) Eliminar o termo associado à  $n$ -ésima potência
  - (e) Reinicializar um polinômio
39. Escreva uma função que remove o  $k$ -ésimo nó da lista encadeada.
  40. Escreva uma função recursiva e outra não recursiva para contar o número de elementos na lista
  41. Generalize o conceito de lista circular para Lista Circular Duplamente Encadeada. Implemente as funções necessárias para o gerenciamento da mesma.
  42. Faça uma função para retornar o número de nós da lista que possuem valor maior que  $n$ .
  43. Considere uma lista de inteiros. Faça uma função para somar os valores contidos em uma lista.
  44. Considere uma lista de inteiros. Faça uma função para retornar o número de nós da lista que possuem um número primo armazenado.