

Algoritmos e Estruturas de Dados II

Terceira Lista de Exercícios – Listas

Listas

1. Faça uma função para concatenar duas listas em uma terceira lista que deverá ser retornada pela função. Não use funções já implementadas.
2. Faça uma função para remover os n primeiros elementos de uma lista. A função deve retornar se a operação foi possível ou não.
3. 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.
4. Faça uma função para trocar de posição dois elementos de uma lista. A função deve retornar se a operação foi possível ou não.
5. 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.

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

6. Considere uma lista contendo números inteiros positivos. Faça uma função que retorne quantos números pares existem na lista.
7. Considere uma lista contendo números inteiros positivos. Faça uma função que retorne uma nova lista contendo apenas os números pares da lista.
8. Fazer uma função para verificar se uma lista está ordenada ou não (a ordem pode ser crescente ou decrescente).
9. Fazer uma função que copia uma lista L1 em uma outra lista L2.
10. Fazer uma função que copia uma lista L1 em outra L2, eliminando os elementos repetidos.
11. Fazer uma função para inverter uma lista L1 colocando o resultado em L2.
12. Escreva uma função que crie uma lista encadeada a partir de um vetor.

13. 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.
14. Faça uma função insere em uma Lista Estática de modo que os elementos da lista sejam armazenados em ordem crescente.
15. Faça uma função insere em uma Lista Encadeada de modo que os elementos da lista sejam armazenados em ordem decrescente.
16. Faça uma função insere em uma Lista Duplamente Encadeada que receba como parâmetro, uma lista encadeada simples e uma lista estática, e insira em uma nova lista de forma ordenada crescentemente os elementos das duas listas recebidas por parâmetro.
17. Faça uma função remove que removerá um elemento recebido por parâmetro de modo que a lista encadeada simples continue ordenada.
18. Faça uma função remove que removerá um elemento recebido por parâmetro de modo que a lista estática continue ordenada. OBS: Terá que fazer o deslocamento dos dados no vetor.
19. Faça uma função busca que efetuará a busca de um elemento passado por parâmetro, em uma lista duplamente encadeada. Sua função deverá retornar em que posição está na lista, caso ache o elemento e -1 em caso contrário.
20. Implemente uma função recursiva para calcular o tamanho de uma lista dinâmica e encadeada.