

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.

21 - Crie um programa que permita que um usuário gerencie sua lista de filmes. Para tanto, o programa deverá apresentar para o usuário um menu com as seguintes opções:

Menu:

- 1 - Inserir um filme no final da lista
- 2 - Inserir um filme em uma posição específica da lista
- 3 - Remover um filme da lista
- 4 - Remover um filme em uma posição específica da lista
- 5 - Pesquisar se um filme consta na lista
- 6 - Listar todos os filmes da lista
- 7 - Inverter a ordem dos filmes presentes na lista
- 8 - Encerrar o programa

O programa deverá ler a opção informada pelo usuário e executar a operação selecionada. Em seguida o programa deverá apresentar novamente o menu para o usuário, ler e executar a operação selecionada. Esse processo deverá ser repetido até que o usuário digite a opção 8.

22 - Crie um programa que permita ao usuário inserir números inteiros em uma lista. Em seguida, implemente as seguintes funcionalidades:

- 1- Insira um número na lista
- 2 - Verifica se um número se encontra na lista
- 3 - Exiba a soma de todos os números na lista
- 4 - Exiba o maior número na lista
- 5 - Exiba o menor número na lista
- 6 - Remova todos os números pares da lista
- 7 - Exiba os números que estão na lista, após a remoção dos números pares
- 8 - Inverta os elementos da lista
- 9 - Encerre o programa

O programa deverá ser encerrado quando o usuário selecionar a opção 9.

23 - Crie uma aplicação de agenda de contatos que permita ao usuário adicionar, atualizar, listar e excluir contatos. Use uma lista para armazenar os contatos. Implemente as seguintes funcionalidades:

- a) Adicionar um novo contato com nome, telefone e e-mail
- b) Atualizar informações de um contato existente
- c) Excluir um contato da agenda
- d) Listar todos os contatos na agenda