

Lista de exercícios

- 1) Implemente um programa que conta quantas vezes a letra “a” aparece em um arquivo texto.
- 2) Implemente um programa que calcula e apresenta a soma duas matrizes de dimensão 5x5. Estas matrizes devem estar salvas em um único arquivo binário. Observe que o arquivo binário que contém as matrizes, também deve ser gerado pelo programa.
- 3) Implemente uma agenda de contatos que armazena “nome”, “idade” e “numero de telefone”. A agenda deve ser gravada em arquivo binário. O “nome” pode ter no máximo 50 caracteres, a “idade” deve ser um número inteiro e o “telefone” pode ser um vetor de no máximo 20 caracteres. Implemente as opções de “Inserir”, “Alterar”, “Excluir” e “Exibir” contatos. Para as opções de “Alterar”, “Excluir” e “Exibir”, o programa deve perguntar ao usuário o nome do contato e, em seguida, realizar a respectiva operação.

Obs.:

A) Não utilize o “struct” da linguagem C.

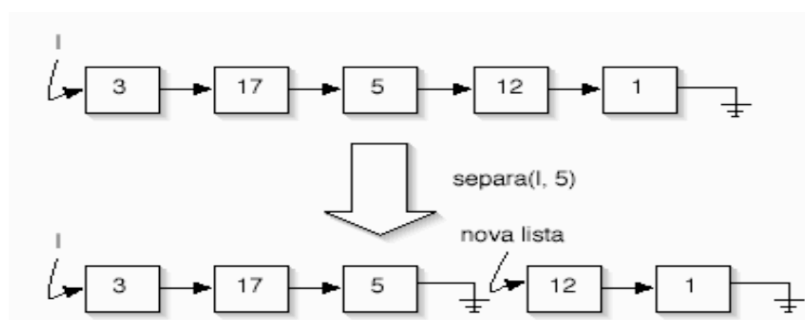
B) Uma opção de implementação seria:

- 1) no início do programa, todo o conteúdo do arquivo é carregado para vetores na memória.
- 2) durante a execução do programa, os dados são manipulados por meio dos vetores na memória.
- 3) Ao final do programa, os vetores são gravados no arquivo.

Você NÃO deve implementar o programa desta maneira. Todas as operações (“Inserir”, “Alterar”, “Excluir” e “Exibir”) devem ser feitas diretamente no arquivo. Por exemplo, quando o usuário escolher a opção “Alterar”, o programa deve encontrar o contato diretamente no arquivo e alterar seu conteúdo.

- 4) Implemente o exercício anterior usando estruturas.
- 5) Implemente um vetor de estruturas onde cada elemento é constituído por um Número positivo e por um Nome (40 caracteres). O vetor não poderá ter mais do que MAX registros (elementos). Implemente as seguintes funções:
 - a) Inserir, que adiciona um registro ao vetor e retorna a posição em que foi introduzido.
 - b) Salvar, que salva em arquivo binário o conteúdo do vetor.
 - c) Carregar, que faz a leitura dos registros existentes no arquivo e os insere no vetor. Se o registro já existir no vetor, ele não deverá ser inserido.
 - d) Listar todos, que lista todos os elementos do vetor.Remover registro, que solicita um Número e remove do vetor o registro correspondente.

- 6) Considere uma lista encadeada simples de valores inteiros, como mostra a figura.



Implemente a função “*separa*” que recebe como parâmetro a lista encadeada e um valor inteiro n e divide 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 acima ilustra esta separação.

A função deve obedecer ao protótipo: *Lista* separa (Lista* l, int n)*; A função deve retornar um ponteiro para a segunda sub-divisão da lista original, e o parâmetro “*l*” deve continuar apontando para o primeiro elemento da primeira subdivisão da lista. Implemente o programa principal para testar sua função. A lista deve ser alocada dinamicamente e preenchida no programa principal.

7)

Considere estruturas de listas encadeadas que armazenam valores inteiros. O tipo que representa um nó da lista é dado por:

```
struct lista {
    int info;
    struct lista* prox;
};
typedef struct lista Lista;
```

Implemente uma função que receba um vetor de valores inteiros com n elementos e construa uma lista encadeada armazenando os elementos do vetor nos nós da lista. Assim, se for recebido o vetor $v[5] = \{3, 8, 1, 7, 2\}$, a função deve retornar uma nova lista cujo primeiro nó tem a informação 3, o segundo a informação 8, e assim por diante. Se o vetor tiver zero elementos, a função deve ter como valor de retorno uma lista vazia. O protótipo da função é dado por:

```
Lista* constroi (int n, int* v);
```

8) Implemente uma agenda de contatos que armazena o “nome”, “celular” e “telefone”. Salve os dados da agenda em arquivo binário. Durante a execução do programa, os dados da agenda deverão ser manipulados utilizando-se uma lista dinâmica. Ao iniciar o programa, carregue os dados do arquivo para a lista dinâmica e, ao fechar o programa, salve as informações da lista no arquivo binário. Implemente as funções de “incluir”, “excluir” e “procurar” um contato.