

# Lista de Exercícios

## Vetores, Recursividade, Ordenação, Listas, Pilhas e Filas

Estruturas de Dados — Bacharelado em Sistemas de Informação — Instituto Federal da Bahia

**Instrução:** Responda todos os exercícios em **Linguagem C**. Faça upload do **código-fonte completo** de cada questão no Classroom.

**Exercício 1.** Implemente uma lista duplamente encadeada para armazenar os dados dos clientes de uma academia. Deve-se armazenar nome, peso, altura, cpf, idade, telefone, data de entrada na academia, e um campo que indique se a mensalidade do mês está paga ou não. Crie um Menu no seu programa com as seguintes funções:

- Inserir alunos
- Excluir alunos por cpf
- Mostrar alunos em ordem alfabética de nomes
- Mostrar os alunos em ordem de antiguidade na academia, do mais recente ao mais antigo.
- Mostrar os alunos em ordem de idade, do mais velho ao mais novo.
- Calcular o IMC de todos os alunos, criar e armazenar um campo em seus próprios usuários. Mostrar o IMC de cada aluno.
- Listar todos os dados de um aluno  
aqui deve-se permitir que o usuário escolha um aluno pelo cpf.

**Exercício 2.** Escreva uma função que receba uma Pilha de inteiros (P1) e copie-a para uma segunda Pilha (P2).

**Exercício 3.** Escreva uma função para remover elementos duplicados de uma Lista Duplamente Encadeada de Inteiros informados pelo usuário.

- Obs1: sua função deve assumir que os elementos da lista estão desordenados.
- Obs2: sua função deve remover quantos elementos duplicados forem necessários, de forma que ao término da chamada da função a lista não possua nenhum elemento duplicado
- Ex: lista de entrada = 3 → 4 → 1 → 3 → 1 → 5 → 9 → 4 → 3 lista na saída = 3 → 4 → 1 → 5 → 9 (quarto, quinto, oitavo e nono elementos foram removidos)

**Exercício 4.** Em um banco, há apenas um caixa e todos devem ser atendidos na ordem de chegada. No entanto, as pessoas idosas têm prioridade e passam ao início da fila. Considerando que, em vários dias, o número de idosos é muito grande, foram estipuladas as seguintes regras:

1. uma pessoa é atendida na ordem de chegada.
2. no máximo, 2 idosos podem passar à frente de uma pessoa que não é idosa.

Escreva um programa que leia uma sequência de informação da ordem de chegada e categoria do cliente e imprima a ordem de atendimento (considere que todos chegaram antes de começar o atendimento, na ordem em que são apresentados).

Ex. de entrada:

geral, geral, idoso, idoso, idoso, geral, idoso

ou pode ser lido tb: 1,1,2,2,2,1,2 (considerando que 1 é geral e 2 é idoso) Ex. saída:  
3 4 1 2 5 7 6

OBS.: Crie e use as funções enfileirar, desenfileirar, imprimir fila, etc, faça o algoritmo para funcionar com qualquer entrada fornecida. Pode usar números em lugar das palavras 'geral' e 'idoso'.

**Exercício 5.** Implemente uma busca linear em um vetor de inteiros. O programa deve informar se o valor  $X$  está presente e, em caso afirmativo, sua posição.

**Exercício 6.** Leia dois vetores  $A$  e  $B$ , ambos de tamanho  $N$ , e construa um vetor  $C$  de tamanho  $2N$  intercalando os elementos de  $A$  e  $B$ .

**Exercício 7.** Implemente os seguintes algoritmos de ordenação aplicados a um vetor de inteiros Bubble Sort, Selection Sort e Insertion Sort. Mostre o vetor antes e depois de cada ordenação.

**Exercício 8.** Modifique o algoritmo Insertion Sort para ordenar os elementos em ordem decrescente. Explique quais partes do código precisaram ser alteradas.

**Exercício 9.** Leia um vetor de  $N$  inteiros, ordene-o pelo Selection Sort e, em seguida, realize uma busca binária para encontrar o valor  $X$  informado pelo usuário.

**Exercício 10.** Leia dois vetores  $A$  e  $B$  já ordenados em ordem crescente e implemente a fusão em um terceiro vetor  $C$ , também ordenado. Não utilize algoritmos de ordenação prontos, apenas a técnica de mesclagem.

**Exercício 11.** Desenvolva uma lista simplesmente encadeada circular cujos nós armazenem identificadores do tipo inteiro. A lista deve manter seus elementos sempre ordenados em ordem crescente. Dessa forma:

- a inserção de novos elementos deve posicioná-los automaticamente na ordem correta;
- a remoção de elementos deve preservar a ordem da lista após a exclusão.

Implemente as seguintes operações:

- Inserção de um elemento inteiro na lista, respeitando a ordenação crescente;
- Remoção de um elemento inteiro, mantendo a lista ordenada;

- Consulta (busca) de um elemento específico na lista;
- Impressão de todos os elementos da lista, na ordem crescente.

**Exercício 12.** Implemente o sistema para a biblioteca usando uma lista duplamente encadeada:

- Cada livro deve ser representado por um registro com Nome do livro, disponibilidade e fila de espera
- Ao requisitar um livro, a pessoa entra na fila de espera, caso o livro não esteja disponível, seguindo a política FIFO. Ou seja, quando um livro fica disponível, o primeiro da fila de espera do livro deve receber o livro
- Implemente as demais funcionalidades: cadastrar livro, remover livro, emprestar livro, devolver livro e cadastrar leitor.

**Exercício 13.** Implemente, em Linguagem C, uma **lista duplamente encadeada de nomes**, contendo obrigatoriamente as seguintes funções:

- Criação da lista vazia;
- Inserção de um nome no início da lista;
- Inserção de um nome no final da lista;
- Remoção de um nome informado pelo usuário;
- Busca de um nome na lista;
- Impressão de todos os nomes da lista;
- Ordenação alfabética da lista

Deve-se também:

- Utilizar `struct` para representar o nó da lista;
- Utilizar alocação dinâmica de memória (`malloc` e `free`);
- O programa deve possuir um menu interativo no `main()` ou numa função a ser invocada no `main()` para o menu.

Teste seus programas com vetores de diferentes tamanhos, incluindo casos com vetores vazios, um único elemento e elementos repetidos.