

SLTESOS – Lista de Exercícios

Fábio Modesto

Bacharelado em Ciência da Computação
Instituto Federal de Educação, Ciência e Tecnologia de São Paulo
fabiomodesto@ifsp.edu.br
slt.ifsp.edu.br

Exercício 1

Implemente uma função que retorne a quantidade de nós de uma lista ligada.

Exercício 2

Implemente uma função que verifique se duas listas encadeadas são iguais. Duas listas são consideradas iguais se têm a mesma sequência de elementos. O protótipo da função deve ser dado por: `int igual (Lista* l1, Lista* l2)`

Exercício 3

Implemente uma função que crie uma cópia de uma lista encadeada. O protótipo da função deve ser dado por: `Lista* copia (Lista* l)`

Exercício 4

Desenvolva uma função que inverta os elementos de uma lista encadeada sem criar uma nova lista. A inversão deve ser feita alterando os ponteiros dos nós existentes. Desafie os alunos a implementarem a inversão de forma eficiente e a testarem a função imprimindo a lista antes e depois da inversão.

Exercício 5

Desenvolva um programa em C++ que simule um sistema de controle de estoque utilizando lista simplesmente encadeada. Cada nó da lista deve representar um produto e armazenar, além do ponteiro para o próximo nó, três campos: o nome do produto (`std::string`), a quantidade em estoque (`int`) e o preço unitário (`double`). O programa deve oferecer as seguintes funcionalidades implementadas como funções separadas:

- Inicializar o estoque, retornando ponteiro nulo.
- Adicionar um produto: recebendo nome, quantidade e preço, a função deve inserir um novo nó ao final da lista; se um produto com o mesmo nome já existir, apenas acumule a quantidade e atualize o preço.

- Remover um produto por nome: trate corretamente os casos de remoção do primeiro nó, de um nó intermediário e de tentativa de remoção de um produto inexistente, retornando a nova cabeça da lista. Imprimir o estoque completo: percorra a lista exibindo nome, quantidade, preço unitário e valor total de cada item ($\text{quantidade} \times \text{preço}$).
- Calcular valor total do estoque: percorra a lista somando $\text{quantidade} \times \text{preço}$ de cada produto e retorne esse valor.
- Liberar toda a memória ao final, desalocando cada nó.

No `main()`, apresente um menu em loop que permita ao usuário escolher entre adicionar produto, remover produto, listar estoque, mostrar valor total e sair. Ao inserir ou remover, solicite ao usuário o nome do produto (e demais dados quando for adicionar) e exiba mensagens de confirmação ou de erro (por exemplo, “Produto não encontrado”). Ao encerrar, chame a função de liberação de memória e saia do programa.