
1. Criando e Acessando Structs com Ponteiros

Crie uma struct chamada Computador, que tenha os seguintes atributos:

- marca (char[30])
- modelo (char[30])
- preco (float)

No programa principal, declare uma variável dessa struct e um ponteiro para ela. Use o ponteiro para atribuir valores e imprimir os dados do computador.

2. Modificando uma Struct por Referência

Crie uma struct chamada Carro com os atributos:

- marca (char[20])
- ano (int)

Implemente uma função alterarAno(Carro* c, int novoAno) que recebe um ponteiro para Carro e altera o valor do ano. No main(), crie uma variável dessa struct, chame a função e exiba o valor atualizado.

3. Array de Structs com Ponteiros

Crie uma struct chamada Produto contendo:

- nome (char[30])
- preco (float)

No main(), declare um array com **3 produtos** e um ponteiro para esse array. Use o ponteiro para percorrer os produtos e imprimir suas informações.

4. Lista Ligada Simples

Crie uma struct chamada No, que representa um nó de uma **lista ligada** e contém:

- dado (int)
- proximo (ponteiro para o próximo nó)

Implemente funções para:

- Inserir elementos no início da lista.
- Exibir a lista.

Crie um main() que adicione **3 números** à lista e os exiba.

5. Struct com Alocação Dinâmica

Crie uma struct chamada Aluno contendo:

- nome (char[50])
- nota (float)

No main(), use new para alocar dinamicamente um Aluno, preencha os valores digitados pelo usuário e exiba os dados. No final, libere a memória com delete.

6. Vetor Dinâmico de Structs

Crie uma struct chamada Funcionario contendo:

- nome (char[50])
- salario (float)

No main(), use new para criar dinamicamente um **array de 5 funcionários**, preencha os valores e exiba as informações. Após isso, libere a memória com delete[].

7. Struct com Ponteiro para String

Crie uma struct chamada Livro contendo:

- titulo (ponteiro para char)
- ano (int)

Implemente uma função criarLivro(Livro* l, const char* titulo, int ano) que recebe um ponteiro para Livro e inicializa os atributos usando new. No final, libere a memória com delete[].

8. Ponteiro para Ponteiro em Struct

Crie uma struct chamada Pessoa contendo:

- nome (char[50])
- idade (int)

Implemente uma função modificarIdade(Pessoa** p, int novaldade), que recebe um **ponteiro duplo** e altera a idade. No main(), crie um objeto Pessoa, passe para a função e exiba o novo valor.

9. Lista Ligada com Remoção

Dado a struct No da questão 4, implemente uma função removerElemento(No*& cabeca, int valor), que remova o nó da lista que contém o número valor.

No main(), crie uma lista com 3 valores e remova um deles, exibindo a lista antes e depois da remoção.

10. Troca de Valores Usando Ponteiros

Crie uma struct chamada Ponto contendo:

- x (int)
- y (int)

Implemente uma função trocarPontos(Ponto* a, Ponto* b) que troca os valores de x e y entre dois pontos. No main(), crie dois pontos, chame a função e exiba os valores antes e depois da troca.
