

# Simulados de Programação em C – Cabeçalhos, Arquivos, Ponteiros e Structs

---

## Simulado 1

1. Explique a finalidade de um arquivo de cabeçalho (.h) em C e o que geralmente se declara nele.
  2. O que são as diretivas `#ifndef`, `#define` e `#endif`? Para que servem?
  3. Crie uma struct chamada `Livro` com campos: título, autor e ano de publicação.
  4. Escreva um código em C que leia os dados de 3 livros e os exiba formatados na tela.
  5. Diferencie variáveis globais e locais com um exemplo em código.
  6. Escreva uma função que receba um inteiro por valor e tente alterá-lo dentro da função. A alteração se refletirá fora da função? Justifique.
  7. Refaça a função anterior utilizando ponteiros para que a alteração aconteça fora da função.
  8. Implemente um programa que aloque dinamicamente um vetor de `n` floats, leia os valores e calcule a média.
  9. Como acessar o quinto elemento de um vetor `float *vet` alocado dinamicamente?
  10. Escreva um programa que use `malloc` para alocar uma matriz 3x3 de inteiros e a preencha com os valores da multiplicação de linha por coluna.
- 

## Simulado 2

1. O que é uma struct em C e para que serve?
2. Declare uma struct chamada `Produto` com campos: nome, preço e quantidade.
3. Crie um vetor de 5 produtos e leia os dados do usuário.
4. Imprima o total em estoque de cada produto (preço x quantidade).
5. Qual a diferença entre `fgets` e `scanf` para leitura de strings?
6. Escreva um programa que grave os nomes de 3 alunos em um arquivo e depois leia esse arquivo e exiba o conteúdo.
7. Explique o uso da função `ferror` em operações com arquivos.
8. Qual a vantagem da modularização de código em projetos maiores?
9. Faça um programa que leia 10 números inteiros e grave cada um em uma linha de um arquivo `numeros.txt`.
10. Escreva um programa que leia uma string e use ponteiros para imprimir cada caractere em uma nova linha.