

## Exercício: Implementação e Revisão de Estruturas de Dados Heterogêneas em C

### Introdução

Neste exercício, você revisará e aplicará conceitos fundamentais sobre estruturas de dados heterogêneas em C, incluindo ponteiros, structs, estruturas de dados aninhadas e o uso do **typedef**.

### Ferramentas Recomendadas

#### OnLine:

- [OnlineGDB](#): Um compilador C online e ambiente de desenvolvimento.
- [JDoodle](#): Outro compilador C online.

#### Instaladas:

- [Code::Blocks](#): Um ambiente de desenvolvimento integrado (IDE) para C/C++.
- [GCC Compiler](#): Compilador C padrão para sistemas Unix.

### Sites para Dúvidas

- [Stack Overflow](#): Uma comunidade de desenvolvedores onde você pode fazer perguntas e obter respostas.
- [C Programming Forum](#): Um fórum específico para linguagem C.
- [GeeksforGeeks](#): Seção de linguagem C com tutoriais e soluções.

### Instruções

#### Ponteiro e Structs

- a. Declare um ponteiro para um inteiro e aloque dinamicamente memória para um valor inteiro usando **malloc**. Atribua o valor **100** ao inteiro e imprima via ponteiro.
- b. Crie uma struct chamada **Student** que contém um nome (string), idade (inteiro) e nota (float).
- c. Declare uma variável do tipo **Student** e inicialize-a com valores de sua escolha. Imprima os valores.

#### Estrutura de Dados Heterogênea com Ponteiros

- a. Crie um ponteiro para a struct **Student** e aloque dinamicamente memória para ela usando **malloc**.
- b. Inicialize os valores da struct alocada dinamicamente e imprima-os.

#### Estruturas de Dados Aninhadas e Vetores de Structs

- a. Crie uma nova struct chamada **Course** que contém um nome (string) e um vetor de **Student**.

- b. Declare e inicialize uma variável do tipo **Course** com pelo menos 3 estudantes. Imprima os nomes e notas dos estudantes.

### Uso de typedef

- a. Usando a instrução **typedef**, crie um novo nome **CourseList** para um vetor de **Course** que pode conter até 10 cursos.
- b. Declare uma variável do tipo **CourseList** e inicialize-a com dados de sua escolha. Imprima os dados.

### Dicas:

- Lembre-se de incluir as bibliotecas necessárias (**#include <stdio.h>**, **#include <stdlib.h>**, etc.).
- Ao usar **malloc**, sempre verifique se a memória foi alocada com sucesso e lembre-se de liberar a memória após o uso com **free**.

### Critérios de Avaliação:

- A corretude do código: Os programas executam conforme especificado?
- Organização e clareza: O código é bem estruturado e comentado?
- Cobertura: Todos os subtemas foram adequadamente abordados e implementados no código?

Ao concluir este exercício, você terá uma compreensão mais profunda das estruturas de dados heterogêneas em C e como manipulá-las usando diferentes ferramentas e recursos da linguagem.