

Prezados(as),

Segue a lista de exercícios sobre o tema "Ponteiros em Linguagem C". O objetivo desta lista é avaliar a compreensão e a aplicação prática dos conceitos de declaração, manipulação e uso de ponteiros, inclusive em conjunto com funções.

A entrega desta atividade deverá ser realizada através do **envio do link do seu repositório Git pessoal na plataforma Digital Fametro**, na área designada para esta atividade.

Instruções para a Entrega

1. **Repositório:** Crie um repositório no GitHub com o nome Estrutura de Dados 2 - Fametro.
2. **Organização:** Dentro do repositório, crie uma pasta chamada Lista_Exercicios_1_Ponteiros.
3. **Arquivos:** Cada questão prática deve ser respondida em um arquivo de código-fonte .c separado. O nome do arquivo deve ser o número da questão, por exemplo, **questao_1.c**, **questao_2.c**, e assim por diante. As questões teóricas podem ser respondidas em um único arquivo de texto ou Markdown (respostas_teoricas.txt ou respostas_teoricas.md) dentro da mesma pasta.
4. **Código:** Quando necessário, o código deve estar bem comentado, explicando a lógica utilizada para resolver o problema.

Lista de Exercícios: Ponteiros em C

Parte I: Questões Práticas

Instrução: Para cada questão abaixo, crie um programa em C que atenda ao solicitado.

Declaração e Manipulação Básica

1. **Troca de Valores:** Crie uma função que receba dois ponteiros para inteiros como parâmetros e troque os valores das variáveis para as quais eles apontam. No main, declare duas variáveis, leia seus valores, chame a função e imprima os valores trocados.
2. **Maior e Menor:** Escreva uma função que receba um array de inteiros e seu tamanho, e através de ponteiros, retorne o maior e o menor elemento do array. A função deve ter o seguinte protótipo: void encontrarMaiorMenor(int *array, int tamanho, int *maior, int *menor);.

3. **Endereços de Memória:** Declare uma variável inteira e um ponteiro para inteiro. Atribua o endereço da variável ao ponteiro. Imprima o valor da variável, o endereço da variável, o valor do ponteiro (que deve ser o mesmo endereço da variável) e o valor apontado pelo ponteiro.
4. **Dobro do Valor:** Escreva uma função chamada dobrar que receba um ponteiro para um número inteiro como argumento e modifique o valor da variável original para que seja o dobro do seu valor. Crie uma função main para testar a sua função.

Ponteiros e Funções

5. **Cópia de String:** Implemente uma função que copie uma string para outra usando ponteiros. O protótipo deve ser: void copiarString(char *origem, char *destino);. Não utilize a função strcpy() da biblioteca padrão.
6. **Contagem de Vogais e Consoantes:** Escreva uma função que receba uma string (ponteiro para char) como parâmetro e retorne, através de ponteiros passados como argumentos, o número de vogais e o número de consoantes na string. Protótipo: void contarVogaisConsoantes(char *str, int *vogais, int *consoantes);.
7. **Inversão de String:** Crie uma função que receba uma string e a inverta "in-place", ou seja, modificando a própria string original, utilizando ponteiros. Por exemplo, "brasil" deve se tornar "lisarb".
8. **Calculadora Básica:** Crie uma função que receba dois números de ponto flutuante e ponteiros para outras quatro variáveis de ponto flutuante. A função deve calcular a soma, subtração, multiplicação e divisão dos dois primeiros números e armazenar os resultados nas variáveis apontadas pelos ponteiros.

Parte II: Questões Teóricas (Múltipla Escolha)

Instrução: Para cada questão, escolha a alternativa correta.

1. Qual das seguintes afirmações sobre ponteiros em C é VERDADEIRA?
 - a) Um ponteiro armazena o valor de uma variável.
 - b) Um ponteiro armazena o endereço de memória de uma variável.
 - c) A declaração int *p; aloca memória para um inteiro.
 - d) O operador & é usado para acessar o valor apontado por um ponteiro.

2. O que o operador * (asterisco) faz quando usado na frente de uma variável do tipo ponteiro que já foi inicializada?

- a) Declara um novo ponteiro.
- b) Retorna o endereço de memória do ponteiro.
- c) Realiza uma operação de multiplicação.
- d) Acessa (dereferencia) o valor armazenado no endereço de memória para o qual o ponteiro aponta.

3. Para que serve a função malloc() da biblioteca <stdlib.h>?

- a) Para liberar memória previamente alocada.
- b) Para declarar uma variável do tipo ponteiro.
- c) Para alocar um bloco de memória de um tamanho especificado em bytes e retornar um ponteiro para o início desse bloco.
- d) Para copiar o conteúdo de um bloco de memória para outro.

4. Considere a função void modifica(int *a, int b) que altera o valor da variável para a qual a aponta. Como seria a chamada correta desta função na main para modificar uma variável int num = 5;?
