Workshop C - Aula 03

Funções e Ponteiros
02/10/2025

Samuel Abrão

IDP
abrao.sam.2006@gmail.com
github.com/samuka7abr

Cronograma Aula 03

- **Explicação teórica**

Cronograma Aula 03

- **Explicação teórica**
- **Exercícios de cada tópico com código pré-feito**

Cronograma Aula 03

- **Explicação teórica**
- **Exercícios de cada tópico com código pré-feito**
- **Hoje não é no codeforces**

Cronograma Aula 03

- **Explicação teórica**
- **Exercícios de cada tópico com código pré-feito**
- **Hoje não é no codeforces**
- **Monitores para ajuda**

Declaração e Definição de Funções

Funções - Sintaxe Básica

```c
#include <stdio.h>

// Declaração (ou protótipo) geralmente são feitos em um arquivo de
↪  cabeçalho (.h)
int soma(int a, int b);

// Definição (implementação)
int soma(int a, int b) {
    return a + b;
}

int main() {
    int x = soma(5, 3);  // Chamada da função
    printf("5 + 3 = %d\n", x);
    return 0;
}
```

Funções - Exemplo com void

```c
#include <stdio.h>

// Função que não retorna valor
void imprime_linha(char c, int n) {
    for (int i = 0; i < n; i++) {
        printf("%c", c);
    }
    printf("\n");
}

int main() {
    imprime_linha('*', 10);  // imprime 10 asteriscos
    printf("Olá, mundo!\n");
    imprime_linha('-', 10);  // imprime 10 hífens
    return 0;
}
```

Hora da prática!

Questão 1 — Soma de n números

Crie uma função `somaN` que recebe um valor `n` e retorna a soma dos primeiros `n` números naturais (1 + 2 + 3 + … + n).

Exemplo: `somaN(5)` deve retornar 15 (1 + 2 + 3 + 4 + 5 = 15).

Questão 2 — Troca por referência

Crie uma função `troca` que recebe dois ponteiros para inteiros e troca os valores das variáveis correspondentes.

Exemplo: Se `a = 5` e `b = 10`, após `troca(&a, &b)`, temos `a = 10` e `b = 5`.

Questão 3 — Fatorial recursivo

Crie uma função recursiva `fatorial` que calcula o fatorial de um número `n`.

Lembrando: `n! = n × (n-1) × (n-2) × ... × 1` e `0! = 1`.

Exemplo: `fatorial(5)` deve retornar 120 (5 × 4 × 3 × 2 × 1 = 120).

Questão 4 — Fibonacci recursivo

Crie uma função recursiva `fibonacci` que retorna o n-ésimo número da sequência de Fibonacci.

Lembrando: F(0) = 0, F(1) = 1, e F(n) = F(n-1) + F(n-2) para n > 1.

Exemplo: `fibonacci(7)` deve retornar 13 (sequência: 0, 1, 1, 2, 3, 5, 8, 13).

Passagem de Parâmetros

Passagem por Valor

```c
#include <stdio.h>

// Recebe uma cópia do valor
void troca(int a, int b) {
    int temp = a;
    a = b;
    b = temp;
    printf("Dentro: a=%d, b=%d\n", a, b);
}

int main() {
    int x = 5, y = 10;
    printf("Antes: x=%d, y=%d\n", x, y);
    troca(x, y);  // Passagem por valor
    printf("Depois: x=%d, y=%d\n", x, y);
    return 0;
}
```

Passagem por Referência (Ponteiros)

```c
#include <stdio.h>

// Recebe o endereço da variável
void troca_ptr(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
    printf("Dentro: *a=%d, *b=%d\n", *a, *b);
}

int main() {
    int x = 5, y = 10;
    printf("Antes: x=%d, y=%d\n", x, y);
    troca_ptr(&x, &y);  // Passagem por referência
    printf("Depois: x=%d, y=%d\n", x, y);
    return 0;
}
```

Introdução a Ponteiros

Ponteiros - Básico

```c
#include <stdio.h>

int main() {
    int x = 42;
    int *p = &x;  // p é um ponteiro para int

    printf("Valor de x: %d\n", x);
    printf("Endereço de x: %p\n", &x);
    printf("Valor de p: %p\n", p);
    printf("Valor apontado por p: %d\n", *p);

    *p = 100;  // Modifica x através do ponteiro

    printf("Novo valor de x: %d\n", x);
    return 0;
}
```

Ponteiros - Operadores & e *

```c
#include <stdio.h>

int main() {
    int a = 5;
    int *ptr = &a;

    printf("a = %d\n", a);
    printf("&a = %p\n", &a);       // & = endereço
    printf("ptr = %p\n", ptr);
    printf("*ptr = %d\n", *ptr);    // * = valor apontado

    *ptr = 10;  // *ptr é um alias para a
    printf("Após *ptr = 10: a = %d\n", a);
    return 0;
}
```

Hora da prática!

Questão 1 (Básica — endereço e valor)

Escreva um programa que leia um número inteiro e mostre:

• o valor digitado • o endereço de memória onde ele está armazenado (usando ponteiro)

Dica: Use & para obter o endareço e * para acessar o valor apontado.

Questão 2 (Intermediária — troca de valores)

Implemente uma função `troca` que receba dois ponteiros para inteiros e troque os valores apontados por eles.

No `main`, leia dois inteiros, chame a função e exiba os valores trocados.

Exemplo: Se `a = 5` e `b = 10`, após `troca(&a, &b)`, temos `a = 10` e `b = 5`.

Questão 3 (Intermediária — soma com ponteiro)

Crie uma função `soma` que receba dois ponteiros para inteiros e retorne a soma dos valores apontados.

No `main`, leia dois inteiros, passe seus endereços para a função e exiba o resultado.

Exemplo: Para `a = 7` e `b = 3`, `soma(&a, &b)` deve retornar `10`.

Questão 4 (Avançada — vetor e ponteiro aritmético)

Escreva um programa que leia o tamanho `n` de um vetor de inteiros, aloque dinamicamente a memória com `malloc`, preencha os valores, e use aritmética de ponteiros para:

• imprimir todos os elementos • calcular a soma • encontrar o maior elemento

Dica: Use `malloc(n * sizeof(int))` e não esqueça do `free`!

Relação entre Ponteiros e Arrays

Arrays como Ponteiros

```c
#include <stdio.h>

int main() {
    int v[5] = {10, 20, 30, 40, 50};
    int *p = v;  // v é um ponteiro para o primeiro elemento

    // v[0] é equivalente a *v
    printf("v[0] = %d, *v = %d\n", v[0], *v);

    // v[2] é equivalente a *(v + 2)
    printf("v[2] = %d, *(v + 2) = %d\n", v[2], *(v + 2));

    // Iterando com ponteiro
    for (int i = 0; i < 5; i++) {
        printf("v[%d] = %d\n", i, *(p + i));
    }
    return 0;
}
```

Aritmética de Ponteiros

```c
#include <stdio.h>

int main() {
    int v[5] = {10, 20, 30, 40, 50};
    int *p = v;

    printf("p aponta para: %d\n", *p);
    p++;  // Avança para o próximo elemento
    printf("p++ aponta para: %d\n", *p);

    p += 2;  // Avança 2 posições
    printf("p += 2 aponta para: %d\n", *p);

    // Percorrendo o array
    p = v;  // Volta para o início
    while (p < v + 5) {
        printf("%d ", *p);
        p++;
    }
    printf("\n");
    return 0;
}
```

Obrigado!

Perguntas?