



## Revisão Prova Final

### PARTE 1: FUNÇÃO BÁSICO

#### Questão 0a (Exemplo):

Crie e execute uma função que receba dois números inteiros, como parâmetros, e retorne a soma desses dois números.

```
#include <stdio.h>
```

```
int soma(int a, int b){  
    int resultado = a + b;  
    return resultado;  
}
```

```
int main(){  
    int num1,num2;  
    scanf("%d", &num1);  
    scanf("%d", &num2);  
    int r = soma(num1,num2);  
    printf("%d",r);  
}
```

#### Questão 1

Crie e execute uma função que receba, como parâmetro, a massa e a altura de uma pessoa e retorne o seu IMC.

$$IMC = \frac{massa}{(altura)^2}$$

#### Questão 2

Dada a coordenada de dois pontos no plano cartesiano, a distância entre eles pode ser calculada usando a fórmula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

onde  $(x_1, y_1)$  e  $(x_2, y_2)$  são as coordenadas dos dois pontos.

Sabendo disso, crie uma função que receba a coordenada de dois pontos ( $x_1$ ,  $y_1$ ,  $x_2$  e  $y_2$ ) e retorne a distância entre eles. Depois disso, execute a função e exiba o resultado por ela.

### Questão 3

Crie e execute uma função uma função que receba 2 números e retorne o maior dos números recebidos.

### Questão 4

O salário líquido de um empregado é calculado através do seu salário base. Para isso, calcula-se o imposto de renda e a contribuição previdenciária, que são descontadas do salário base. A contribuição previdenciária é sempre 7,5% do salário base e o cálculo do imposto de renda ocorre conforme a tabela abaixo:

Tabela Fictícia do Imposto de Renda

Faixa Salarial (R\$)	Alíquota (%)
Até 1.500,00	Isento
De 1.500,01 até 2.500,00	7,5
Acima de 4.664,68	22,5

Sabendo disso, faça um programa que peça para o usuário digitar o seu salário base e, após isso, o programa calcule e exiba o seu salário líquido. Neste programa, faça a função imposto de renda que recebe (como parâmetro) o salário base e retorna o valor do imposto de renda devido.

## PARTE 2: FUNÇÃO E VETOR

### Questão 5

Crie um programa que contenha um vetor com 10 números e depois exiba o vetor criado. Para facilitar, crie uma função que receba (como parâmetro) um vetor e exiba o vetor. Atenção: a função de exibir o vetor não deve retornar nenhum valor, ou seja, ela é **void**.

```
#include <stdio.h>

void exibir_vetor(int *vetor, int tam){
    for (int i=0; i<tam; i++){
        printf("%d ", vetor[i]);
    }
    printf("\n");
}

int main(){
    int meu_vetor[10] = {10, 1, 2, 3, 4, 9, 8, 7, 6, 5};
    exibir_vetor(meu_vetor, 10);
}
```

### Questão 6

Crie um programa que contenha um vetor com 10 números e depois exiba o vetor criado. Para facilitar, crie uma função que receba (como parâmetro) um vetor e exiba o vetor recebido.

Dicas:

- A função de exibir o vetor não deve retornar nenhum valor, ou seja, ela é **void**.
- A função também deve receber o tamanho do vetor

```
#include <stdio.h>

void exibir_vetor(int *vetor, int tam){
    for (int i=0; i<tam; i++){
        printf("%d ", vetor[i]);
    }
    printf("\n");
}

int main(){
    int meu_vetor[10]= {10, 1, 2, 3, 4, 9, 8, 7, 6, 5};
    exibir_vetor(meu_vetor, 10);
}
```

### Questão 7

Crie e execute uma função que receba um vetor de números e o seu tamanho e retorne a soma do primeiro com o último elemento do vetor.

### Questão 8

Crie e execute uma função que receba um vetor de números e o seu tamanho e retorne 1 caso o vetor possua o número 10 e 0 caso contrário.

**Dicas:** a função deve percorrer o vetor e verificar se o número 10 está presente. Se o número 10 estiver presente a função retora 1, se o número 10 não estiver presente a função deve retornar 0.

### Questão 9

Crie e execute uma função que receba um vetor de números e o seu tamanho e quantas vezes o número 4 aparece no vetor.

## PARTE 2: FUNÇÃO E ESTRUTURA

### Questão 10

Crie um programa que contenha uma estrutura do tipo Aluno. Neste programa, um aluno é formado por um nome e uma média. Preencha e crie uma variável com o tipo da estrutura Aluno, preencha os seus campos e exiba-os. Para exibir as informações de

um aluno, crie uma função que recebe (como argumento) um aluno e exiba seus campos.

```
#include <stdio.h>
#include <string.h>
struct Aluno{
    char nome[20];
    float media;
};

void exibir_aluno(struct Aluno alu){
    printf("Nome = %s\n", alu.nome);
    printf("Media = %f\n", alu.media);
}

int main(){
    struct Aluno a1;
    strcpy(a1.nome, "Josefina");
    a1.media = 9.0;
    exibir_aluno(a1);
}
```

### Questão 11

Na robótica é comum a manipulação de componentes como LEDs, Motores, Botões, entre outros. Cada componente tem características próprias para serem manipulados. O LED, por exemplo, necessita saber qual é o Pino que ele está conectado no controlador e o seu estado (APAGADO ou ACESO). Sabendo disso, crie um componente LED que possui um *Pino* e um *Estado*, preencha os campos de seus componentes e exiba seus valores.

*Dicas:*

- *Os pinos são números inteiros e o Estado pode ser representado por um número inteiro em que 0 significa apagado e 1 significa aceso.*
- *Neste exemplo não necessita de função.*

### Questão 12

Utilizando a estrutura da questão anterior crie e execute uma função que receba um LED e altere o seu estado de apagado para aceso. Ou seja, mude o campo estado para apagado, número 0.

### Questão 13

Crie uma estrutura do tipo *Funcionario* que possui um nome e um salário. Depois disso crie e execute uma função que receba (como argumento) um *Funcionario* e retorne o valor que esse funcionário deve pagar de imposto.

**Dica:** o funcionário que recebem até 2000 não pagam imposto e funcionários que recebem mais de 2000 pagam 10% do salário de imposto.

#### **Questão 14**

Na clínica VIVERMAGRO uma pessoa é representada por o seu nome, o seu peso e a sua altura. Lá o IMC é uma preocupação constante. Sabendo disso, crie um programa que possua a estrutura Pessoa, com peso, altura e nome. Depois disso crie uma variável do tipo pessoa e preencha os seus campos. Por último, crie uma função `calcular_imc`, que recebe (como argumento) uma pessoa e retorna o seu IMC.

#### **Questão 15**

A figura geométrica retângulo é representada por uma altura e uma largura. Com base nessa informação, crie o tipo *Retangulo*, com altura e largura. Em seguida, crie uma variável do tipo *Retangulo* e preencha seus campos. Por último, crie e execute uma função `calcular_area`, que recebe um *Retangulo* e retorna a sua área.