

# Exercícios de Programação em C - Variáveis Estruturadas Heterogêneas, Uniões e Enumerações

## Exercício 1

Declare e inicialize uma estrutura para armazenar as informações de um livro (título, autor, ano de publicação) e imprima seus valores.

## Exercício 2

Utilize a declaração de tipos (typedef) para simplificar a definição de uma estrutura para armazenar as informações de um carro (marca, modelo, ano) e imprima seus valores.

## Exercício 3

Atribua os valores de uma estrutura para outra e imprima os valores da nova estrutura.

## Exercício 4

Acesse os componentes de uma estrutura de aluno (nome, matrícula, nota) utilizando o operador ponto ‘.’ e imprima seus valores.

## Exercício 5

Acesse os componentes de uma estrutura de aluno (nome, matrícula, nota) utilizando o operador “->” e imprima seus valores.

## Exercício 6

Declare e inicialize uma estrutura aninhada para armazenar as informações de uma data (dia, mês, ano) e imprima seus valores.

## **Exercício 7**

Crie uma função que recebe uma estrutura de aluno (nome, matrícula, nota) como parâmetro e imprima seus valores.

## **Exercício 8**

Declare um arranjo de estruturas para armazenar as informações de 5 produtos (nome, código, preço) e imprima seus valores.

## **Exercício 9**

Utilize o operador `sizeof()` para determinar o tamanho de uma estrutura de pessoa (nome, idade, altura) e imprima o resultado.

## **Exercício 10**

Declare e inicialize uma união para armazenar um valor inteiro ou um valor flutuante e imprima seus valores.

## **Exercício 11**

Utilize a definição de tipos enumeráveis para representar os dias da semana e imprima os valores.

## **Exercício 12**

Resolva um problema de cadastro de alunos utilizando estruturas.

## **Exercício 13**

Declare e inicialize uma estrutura para armazenar as informações de um funcionário (nome, salário, departamento) e imprima seus valores.

## **Exercício 14**

Utilize a declaração de tipos (`typedef`) para simplificar a definição de uma estrutura para armazenar as informações de um cliente (nome, idade, endereço) e imprima seus valores.

## **Exercício 15**

Atribua os valores de uma estrutura de endereço para outra e imprima os valores da nova estrutura.

## **Exercício 16**

Acesse os componentes de uma estrutura de filme (título, diretor, ano de lançamento) utilizando o operador ponto ‘.’ e imprima seus valores.

## **Exercício 17**

Acesse os componentes de uma estrutura de filme (título, diretor, ano de lançamento) utilizando o operador “->” e imprima seus valores.

## **Exercício 18**

Declare e inicialize uma estrutura aninhada para armazenar as informações de um endereço (rua, número, cidade) e imprima seus valores.

## **Exercício 19**

Crie uma função que recebe uma estrutura de carro (marca, modelo, ano) como parâmetro e imprima seus valores.

## **Exercício 20**

Declare um arranjo de estruturas para armazenar as informações de 3 livros (título, autor, ano de publicação) e imprima seus valores.

## **Exercício 21**

Utilize o operador sizeof() para determinar o tamanho de uma estrutura de produto (nome, código, preço) e imprima o resultado.

## **Exercício 22**

Declare e inicialize uma união para armazenar um valor char ou um valor inteiro e imprima seus valores.

## Exercício 23

Utilize a definição de tipos enumeráveis para representar os meses do ano e imprima os valores.

## Exercício 24

Resolva um problema de cadastro de funcionários utilizando estruturas.

## Exercício 25

Declare e inicialize uma estrutura para armazenar as informações de um aluno (nome, matrícula, nota) e imprima seus valores.

## Exercício 26

Utilize a declaração de tipos (typedef) para simplificar a definição de uma estrutura para armazenar as informações de um professor (nome, disciplina, salário) e imprima seus valores.

## Exercício 27

Atribua os valores de uma estrutura de contato para outra e imprima os valores da nova estrutura.

## Exercício 28

Acesse os componentes de uma estrutura de cliente (nome, idade, endereço) utilizando o operador ponto ‘.’ e imprima seus valores.

## Exercício 29

Acesse os componentes de uma estrutura de cliente (nome, idade, endereço) utilizando o operador “->” e imprima seus valores.

## Exercício 30

Declare e inicialize uma estrutura aninhada para armazenar as informações de um ponto (x, y, z) e imprima seus valores.

### **Exercício 31**

Crie uma função que recebe uma estrutura de funcionário (nome, salário, departamento) como parâmetro e imprima seus valores.

### **Exercício 32**

Declare um arranjo de estruturas para armazenar as informações de 4 produtos (nome, código, preço) e imprima seus valores.

### **Exercício 33**

Utilize o operador `sizeof()` para determinar o tamanho de uma estrutura de funcionário (nome, salário, departamento) e imprima o resultado.

### **Exercício 34**

Declare e inicialize uma união para armazenar um valor `double` ou um valor `char` e imprima seus valores.

### **Exercício 35**

Utilize a definição de tipos enumeráveis para representar as estações do ano e imprima os valores.

### **Exercício 36**

Resolva um problema de cadastro de livros utilizando estruturas.

### **Exercício 37**

Declare e inicialize uma estrutura para armazenar as informações de um cliente (nome, idade, endereço) e imprima seus valores.

### **Exercício 38**

Utilize a declaração de tipos (`typedef`) para simplificar a definição de uma estrutura para armazenar as informações de um contato (nome, telefone, email) e imprima seus valores.

### **Exercício 39**

Atribua os valores de uma estrutura de produto para outra e imprima os valores da nova estrutura.

### **Exercício 40**

Acesse os componentes de uma estrutura de ponto (x, y, z) utilizando o operador ponto '.' e imprima seus valores.