

## Exercícios sobre Structs em C (Nível Básico)

1. **Declaração de struct simples**
  - Declare uma struct chamada Ponto, que contém dois campos: x e y (ambos do tipo int). Não é necessário utilizá-la no programa.
2. **Definição e inicialização**
  - Crie uma struct chamada Data, contendo os campos dia, mes e ano (todos inteiros).
  - Declare uma variável do tipo Data e inicialize-a com os valores 1, 1, 2000.
3. **Entrada e saída de dados**
  - Crie uma struct chamada Retangulo, com os campos largura e altura (ambos float).
  - Leia os valores do usuário para largura e altura e exiba a área do retângulo.
4. **Cálculo com struct**
  - Crie uma struct chamada Triangulo, com os campos base e altura (ambos float).
  - Leia os valores do usuário e calcule a área do triângulo
5. **Struct com múltiplos tipos**
  - Crie uma struct chamada Produto, contendo os campos:
    - id (inteiro)
    - quantidade (inteiro)
    - preco (float)
  - Leia os valores do usuário e exiba o valor total do produto (quantidade  $\times$  preço).
6. **Vetor de struct**
  - Crie uma struct chamada Aluno, contendo os campos matricula (inteiro) e nota (float).
  - Leia os dados de 5 alunos e exiba a média das notas.
7. **Maior valor em vetor de struct**
  - Usando a struct Produto do exercício 5, leia os dados de 3 produtos e exiba o ID do produto mais caro.
8. **Struct com enumeração**
  - Crie uma struct chamada Cor, contendo os campos codigo (inteiro) e nome (string de até 10 caracteres).
  - Leia os dados de 3 cores e exiba a que possui o menor código.
9. **Data válida**
  - Crie a struct Data (do exercício 2) e escreva um programa que leia uma data e verifique se ela é válida.
10. **Operações matemáticas com struct**
  - Usando a struct Ponto (do exercício 1), leia as coordenadas de dois pontos e calcule a distância entre eles.
11. **Ordenação de struct**
  - Usando a struct Aluno (do exercício 6), leia os dados de 3 alunos e exiba suas informações em ordem crescente de nota.
12. **Busca em vetor de struct**

- Usando a struct Produto, leia os dados de 5 produtos e permita que o usuário busque pelo ID de um produto para exibir seus detalhes.
- 13. Soma de valores em struct**
  - Usando a struct Retangulo (do exercício 3), leia as dimensões de 5 retângulos e exiba a soma de suas áreas.
- 14. Filtrar valores em struct**
  - Usando a struct Produto, leia os dados de 5 produtos e exiba os produtos que custam mais de 50 reais.
- 15. Atualização de dados em struct**
  - Usando a struct Aluno, permita que o usuário atualize a nota de um aluno, dado o número de matrícula.

## Exercícios sobre Structs em C (Nível Médio)

- 1. Cadastro de nomes simples**
  - Crie uma struct chamada Pessoa, contendo:
    - nome (string de até 50 caracteres)
    - idade (inteiro)
  - Leia os dados de 3 pessoas e exiba os nomes e as idades cadastrados.
- 2. Struct dentro de outra struct**
  - Crie duas structs:
    - Endereco, contendo:
      - rua (string de até 50 caracteres)
      - numero (inteiro)
    - Pessoa, contendo:
      - nome (string de até 50 caracteres)
      - idade (inteiro)
      - endereco (do tipo Endereco)
  - Leia os dados de 2 pessoas (incluindo endereço) e exiba as informações cadastradas.
- 3. Vetor de struct com string**
  - Crie uma struct chamada Produto, contendo:
    - nome (string de até 50 caracteres)
    - preco (float)
  - Leia os dados de 5 produtos e exiba uma lista com os nomes e preços cadastrados.

Mais exercícios!

Lista de Exercícios desenvolvida pelo Prof. André Backes da UFU (Universidade Federal de Uberlândia)

## Exercícios: Structs

1. Escreva um trecho de código para fazer a criação dos novos tipos de dados conforme

solicitado abaixo:

- Horário: composto de hora, minutos e segundos.
  - Data: composto de dia, mês e ano.
  - Compromisso: composto de uma data, horário e texto que descreve o compromisso.
2. Implemente um programa que leia o nome, a idade e o endereço de uma pessoa e armazene os dados em uma estrutura.
  3. Construa uma estrutura aluno com nome, número de matrícula e curso. Leia do usuário a informação de 5 alunos, armazene em vetor dessa estrutura e imprima os dados na tela.
  4. Crie uma estrutura representando os alunos de um determinado curso. A estrutura deve conter a matrícula do aluno, nome, nota da primeira prova, nota da segunda prova e nota da terceira prova.
    - (a) Permita ao usuário entrar com os dados de 5 alunos.
    - (b) Encontre o aluno com maior nota da primeira prova.
    - (c) Encontre o aluno com maior média geral.
    - (d) Encontre o aluno com menor média geral
    - (e) Para cada aluno diga se ele foi aprovado ou reprovado, considerando o valor 6 para aprovação.
  5. Faça um programa que realize a leitura dos seguintes dados relativos a um conjunto de alunos: Matrícula, Nome, Código da Disciplina, Nota1 e Nota2. Considere uma turma de até 10 alunos. Após ler todos os dados digitados, e depois de armazená-los em um vetor de estrutura, exibir na tela a listagem final dos alunos com as suas respectivas médias finais (use uma média ponderada: Nota1 com peso=1.0 e Nota2 com peso=2.0).
  6. Faça um programa que armazene em um registro de dados (estrutura composta) os dados de um funcionário de uma empresa, compostos de: Nome, Idade, Sexo (M/F), CPF, Data de Nascimento, Código do Setor onde trabalha (0-99), Cargo que ocupa (string de até 30 caracteres) e Salário. Os dados devem ser digitados pelo usuário, armazenados na estrutura e exibidos na tela.
  7. Faça um programa que leia os dados de 10 alunos (Nome, matrícula, Média Final), armazenando em um vetor. Uma vez lidos os dados, divida estes dados em 2 novos vetores, o vetor dos aprovados e o vetor dos reprovados, considerando a média mínima para a aprovação como sendo 5.0. Exibir na tela os dados do vetor de aprovados, seguido dos dados do vetor de reprovados.
  8. Faça um programa que leia um vetor com os dados de 5 carros: marca (máximo 15 letras), ano e preço. Leia um valor p e mostre as informações de todos os carros com preço menor que p. Repita este processo até que seja lido um valor p = 0.
    - Crie e leia um vetor de 5 produtos, com os dados: código (inteiro), nome (máximo 15 letras), preço e quantidade.
    - Leia um pedido, composto por um código de produto e a quantidade. Localize este código no vetor e, se houver quantidade suficiente para atender ao pedido integralmente, atualize o estoque e informe o usuário. Repita este processo até ler um código igual a zero.

Se por algum motivo não for possível atender ao pedido, mostre uma mensagem informando qual erro ocorreu.

- (a) Crie um vetor Cadastro com 5 elementos.
  - (b) Permita ao usuário entrar com dados para preencher esse 5 cadastros.
  - (c) Encontre a pessoa com maior idade entre os cadastrados
  - (d) Encontre as pessoas do sexo masculino
  - (e) Encontre as pessoas com salário maior que 1000.
  - (f) Imprima os dados da pessoa cuja identidade seja igual a um valor fornecido pelo usuário
9. Fazer um programa para simular uma agenda de telefones. Para cada pessoa devem-se ter os seguintes dados:
- Nome
  - E-mail
  - Endereço (contendo campos para Rua, número, complemento, bairro, cep, cidade, estado, país).
  - Telefone (contendo campo para DDD e número)
  - Data de aniversário (contendo campo para dia, mês, ano).
  - Observações: Uma linha (string) para alguma observação especial.
- (a) Definir a estrutura acima.
  - (b) Declarar a variável agenda (vetor) com capacidade de agendar até 100 nomes.
  - (c) Definir um bloco de instruções busca por primeiro nome: Imprime os dados da pessoa com esse nome (se tiver mais de uma pessoa, imprime para todas).
  - (d) Definir um bloco de instruções busca por mês de aniversário: Imprime os dados de todas as pessoas que fazem aniversário nesse mês.
  - (e) Definir um bloco de instruções busca por dia e mês de aniversário: Imprime os dados de todas as pessoas que fazem aniversário nesse dia e mês.
  - (f) Definir um bloco de instruções insere pessoa: Insere por ordem alfabética de nome.
  - (g) Definir um bloco de instruções retira pessoa: Retira todos os dados dessa pessoa e desloca todos os elementos seguintes do vetor para a posição anterior.
  - (h) Definir um bloco de instruções imprime agenda com as opções:
    - Imprime nome, telefone e e-mail.
    - Imprime todos os dados.
  - (i) O programa deve ter um menu principal oferecendo as opções acima.