



Exercícios: Structs

1. Implemente um programa que leia o nome, a idade e o endereço de uma pessoa e armazene os dados em uma estrutura.
2. Utilizando uma estrutura, fazer um programa que permita a entrada de nome, endereço e telefone de 5 pessoas e os imprima em ordem alfabética.
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 converta coordenadas polares para cartesianas:
 - Crie e leia um ponto em coordenada polar, composto por raio (r) e argumento (a) em radianos.
 - Crie outro ponto, agora em coordenada cartesiana, composto por x e y , sabendo que $x = r * \cos a$ e $y = r * \sin a$.

No programa principal, leia um ponto em coordenada polar e mostre as coordenadas do ponto gerado no plano cartesiano.
7. Faça um programa que faça operações simples de números complexos:
 - Crie e leia dois números complexos z e w , compostos por parte real e parte imaginária.
 - Apresente a soma, subtração e produto entre z e w , nessa ordem, bem como o módulo de ambos.
8. 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.
9. 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.
 10. 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.
 11. Considerando a estrutura

```
struct Vetor{
float x;
float y;
float z;
};
```

para representar um vetor no R^3 , implemente um programa que calcule a soma de dois vetores.

12. Peça ao usuário para digitar seus dados pessoais (Nome, Endereço, Data de Nascimento, Cidade, CEP, email), verifique se as informações de Data de Nascimento, CEP e email fazem sentido, e mostre ao usuário as informações, se estão todas corretas, ou mostre que alguma informação estava errada.
13. 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 $p = 0$.
14. Faça um programa que leia um vetor com dados de 5 livros: título (máximo 30 letras), autor (máximo 15 letras) e ano. Procure um livro por título, perguntando ao usuário qual título deseja buscar. Mostre os dados de todos os livros encontrados.
15. Faça um programa que seja uma agenda de compromissos e:
 - Crie e leia um vetor de 5 estruturas de dados com: compromisso (máximo 60 letras) e data. A data deve ser outra estrutura de dados contendo dia, mês e ano.
 - Leia dois inteiros M e A e mostre todos os compromissos do mês M do ano A . Repita o procedimento até ler $M = 0$.

Dica: use `fgets(string, tamanho, stdin)` para ler uma string, precedido imediatamente por `fflush(stdin)`.

16. Faça um programa que controle o consumo de energia dos eletrodomésticos de uma casa e:
 - Crie e leia 5 eletrodomésticos que contêm nome (máximo 15 letras), potência (real, em kW) e tempo ativo por dia (real, em horas).

- Leia um tempo t (em dias), calcule e mostre o consumo total na casa e o consumo relativo de cada eletrodoméstico (consumo/consumo total) nesse período de tempo. Apresente este último dado em porcentagem.

17. Faça um programa que gerencie o estoque de um mercado e:

- 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.

18. Faça um programa que controle o fluxo de vôos nos aeroportos de um país. Com $V = 5$ (vôos) e $A = 5$ (aeroportos) e:

- Crie e leia um vetor de voos, sendo que cada voo contém um código de aeroporto de origem e um de destino.
- Crie um vetor de aeroportos, sendo que cada aeroporto contém seu código, quantidade de voos que saem e quantidade de voos que chegam.

Nota: Cada aeroporto é identificado por um código inteiro entre 0 e $(A-1)$. Não aceite aeroportos de código inexistente.

19. Faça um programa para armazenar um livro de receitas e:

- Crie um vetor de 5 receitas, que deve ter nome (máximo 25 letras), quantidade de ingredientes e ingredientes.
- Para cada receita, leia seu nome e a quantidade de ingredientes. Então crie e leia o vetor de ingredientes, sendo que cada ingrediente contém nome e quantidade.
- Procure receita por nome, mostrando seus ingredientes se encontrar. Se não encontrar, informe ao usuário. Repita o processo até digitar uma string vazia.

20. Faça um programa que armazena filmes produzidos por vários diretores e:

- Crie e leia um vetor de 5 diretores, cada um contendo nome (máximo 20 letras), quantidade de filmes e filmes. O membro filmes é um vetor, que deve ser criado após ter lido quantidade de filmes. Cada filme é composto por nome, ano e duração.
- Procure um diretor por nome, mostrando todos os filmes que ele já produziu. Repita o processo até digitar uma string vazia.

21. Escolha um jogo de cartas, baseado em um “baralho tradicional” (cada carta tem seu naipe e seu valor) ou tipo “super trunfo” (cada carta possui um conjunto de atributos). Implemente a parte de distribuição (sorteio) de cartas para 2 jogadores, considerando que cada jogador irá receber 5 cartas. Exiba na tela as cartas que cada um dos jogadores recebeu.

22. Definir a estrutura cuja representação gráfica é dada a seguir, definir os campos com os tipos básicos necessários.

Cadastro

NOME	ENDEREÇO	SALÁRIO
IDENTIDADE	CPF	ESTADO CIVIL
TELEFONE	IDADE	SEXO

O ENDEREÇO é composto de:

RUA	BAIRRO	CIDADE
ESTADO	CEP	

- (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
23. Escreva um programa que receba dois structs do tipo dma, cada um representando uma data válida, e calcule o número de dias que decorreram entre as duas datas.

```
struct dma {  
    int dia;  
    int mes;  
    int ano;  
};
```

24. 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.