

Exercícios de fixação de conteúdo: sobre STRUCT's

1. Utilizando estrutura, fazer um programa em C que permita a entrada de nome, endereço e telefone de 5 pessoas e os imprima em ordem alfabética.
2. 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.

O programa deve ter um menu principal oferecendo as opções acima.

3. Considerando a estrutura:

```
struct Ponto{  
    int x;  
    int y;  
};
```

para representar um ponto em uma grade 2D, implemente um programa que indique se um ponto p está localizado dentro ou fora de um retângulo. O retângulo é definido por seus vértices inferior esquerdo vl

(x1, y1) e superior direito v2 (x2,y2). A função deve imprimir caso o ponto esteja localizado dentro do retângulo ou não.

4. Considerando a estrutura:

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

para representar um vetor no R3, implemente um programa que calcule a soma de dois vetores. A soma de dois vetores:

Vetor_1 = (x1, y1, z1) e Vetor_2 = (x2, y2, z2) é o vetor Vetor_3 = (x1+x2, y1+y2, z1+z2).

5. 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.
6. Crie uma estrutura representando os alunos do curso de Introdução a Programação de Computadores. 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.
7. Escreva um trecho de código em "C/C++" para fazer a criação dos novos tipos de dados conforme solicitado abaixo:
- o Horário: composto de hora, minutos e segundos.
 - o Data: composto de dia, mês e ano.
 - o Compromisso: composto de uma data, horário e texto que descreve o compromisso.
 - o Byte: usado para armazenar 8 bits (definido em função dos tipos básicos da linguagem C).
8. 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 Biológico (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.

9. 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.
10. 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.
11. 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.
12. Faça um programa que leia um inteiro n e:
 - Crie e leia um vetor com os dados de n 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.
13. Faça um programa que leia um inteiro n e:
 - Crie e leia um vetor com dados de n 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.
14. Faça um programa com N=5 e:
 - Crie e leia um vetor de alunos, sendo que cada aluno contém os dados: nome (máximo 15 letras), notas de 3 provas, média final e nível (inteiro). Este último campo não deve ser lido agora.
 - Preencha o campo nível. Seu valor deve ser igual à parte inteira de $(5 * \text{média final} / \text{média da sala})$. Na sua função main(), mostre o nome e o nível de cada aluno.
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.

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 kWh) 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.
18. Faça um programa que controle o fluxo de voos nos aeroportos de um país. Com $v=5$ (voos) 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.
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 a 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. Crie uma estrutura representando os alunos do curso de Métodos e Técnicas de Programação. 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.

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 Biológico

O ENDEREÇO é composto de:

- Rua
- Bairro
- Cidade
- Estado
- CEP

- (a) Crie um vetor Cadastro com 5 elementos do tipo do registro.
- (b) Permita ao usuário entrar com dados para preencher esses 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 o desejado pelo Usuário.
- (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;  
};
```