



## Lista de Estruturas

- 1) Implemente um programa que leia o nome, a idade e o endereço de uma pessoa e armazene esses dados em uma estrutura. Em seguida, imprima na tela os dados da estrutura lida.
- 2) Crie uma estrutura para representar as coordenadas de um ponto no plano (posições X e Y). Em seguida, declare e leia do teclado dois pontos e exiba a distância entre eles.
- 3) Crie uma estrutura chamada Retângulo. Essa estrutura deverá conter o ponto superior esquerdo e o ponto inferior direito do retângulo. Cada ponto é definido por uma estrutura Ponto, a qual contém as posições X e Y. Faça um programa que declare e leia uma estrutura Retângulo e exiba a área e o comprimento da diagonal e o perímetro desse retângulo.
- 4) Escreva um programa que contenha uma estrutura representando uma data válida. Essa estrutura deve conter os campos dia, mês e ano. Em seguida, leia duas datas e armazene nessa estrutura. Calcule e exiba o número de dias que decorreram entre as duas datas.
- 5) 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.
- 6) 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.

7) 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.

8) 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.

9) Crie uma estrutura representando um aluno de uma disciplina. Essa estrutura deve conter o número de matrícula do aluno, seu nome e as notas de três provas. Agora, escreva um programa que leia os dados de cinco alunos e os armazene nessa estrutura. Em seguida, exiba o nome e as notas do aluno que possui a maior média geral dentre os cinco.

10) 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;  
};
```

Lembre-se de considerar anos bissextos.

11) 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.

12) 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.

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