

## Exercícios sobre Estruturas

- 1- *Verdadeiro ou Falso.* A definição de uma estrutura reserva espaço de memória para conter todos os seus membros.
- 2- *Verdadeiro ou Falso.* A sintaxe da declaração de uma variável de um tipo definido com **struct** difere da sintaxe da declaração de uma variável de tipo simples.
- 3- *Verdadeiro ou Falso.* É possível passar uma variável estrutura para uma função do mesmo modo que passamos uma variável simples.
- 4- A finalidade de definir uma estrutura é:
  - a. Reservar uma quantidade de memória;
  - b. Agilizar um programa;
  - c. Criar novos tipos de dados;
  - d. Armazenar uma lista de dados.

- 5- Considere a seguinte definição:

```
typedef struct
{
    char nome[101];
    float altura;
    float peso;
} Corpo;
```

- a. Escreva uma instrução que declare uma variável chamada *p1* do tipo *Corpo*;
  - b. Escreva uma instrução que atribua o valor 1.68 para a altura da variável acima.
  - c. Escreva as instruções necessárias para definir o tipo *Casal*, contendo dois membros do tipo *Corpo*;
  - d. Escreva a instrução necessária para declarar uma matriz de 10 elementos do tipo *Casal*;
  - e. Escreva as instruções necessárias para preencher o quinto elemento da matriz criada acima com os dados do casal Toinho (*nome = Antonio Pegatudo, peso = 80kg, altura = 1.80m*) e Tiana (*nome = Sebastiana Pulabrejo, peso = 65kg, altura = 1.55m*).
- 6- Considere um pequeno programa para controle de um estacionamento.
- a) Escreva uma estrutura para armazenar os dados de uma entrada. Ela deve ser capaz de armazenar a placa do carro (8 caracteres), a marca (15 caracteres) e a hora de entrada (5 caracteres). Dê a ela o nome de *entrada*.
  - b) Escreva uma instrução que defina uma matriz do tipo *entrada* chamada *vagas*, com 50 linhas.
  - c) Escreva uma função chamada *ObterDadosDaVaga*. Esta função deve obter os dados de uma *entrada* (estrutura definida no item a) e retornar uma variável do tipo *entrada* contendo os dados informados.
  - d) Escreva um programa que utilize a estrutura *entrada*, o vetor *vagas* e a função *ObterDadosDaVaga* para preencher toda a matriz (os dados devem ser informados pelo usuário).