

# LISTA DE EXERCÍCIOS

## Atividade avaliativa

- 1 Elabore um programa que contenha um vetor do tipo float com 10 elementos. Usando um ponteiro, imprima (exiba na tela) o conteúdo e o endereço de cada posição desse vetor.
- 2 Crie um programa que contenha um vetor do tipo int com 5 elementos. Utilizando apenas aritmética de ponteiros (acesse cada posição diretamente pela memória, sem uso do índice do vetor como visto em aula) preencha o vetor com as entradas do usuário e exiba em tela o dobro de cada valor lido.
- 3 Construa uma função que receba como parâmetro um vetor de inteiros com 10 valores aleatórios entre 0 e 6, preenchidos automaticamente. A função deve determinar o maior elemento e o número de vezes que ele ocorre no vetor. Por exemplo, para o vetor [1, 3, 5, 5, 2, 4, 5, 3, 1, 0] a função deve determinar o valor 5 e indicar que ele ocorreu 3 vezes. A função deve ser do tipo **void** e o programa deve, após a execução da função, listar o vetor e apresentar os valores manipulados por ela (maior valor e quantidade de vezes que aparece no vetor).
- 4 Crie uma struct chamada ponto2d que tenha como atributos os pontos x e y. Crie duas estruturas do tipo ponto2d chamadas ponto\_inicial e ponto\_final. A seguir, o programa deve solicitar ao usuário a entrada dos dois pontos (ponto\_inicial e ponto\_final) e exibir o resultado do cálculo da distância entre os dois pontos.  
Lembre: a distância entre dois pontos é a raiz quadrada de  $(x_1 - x_2)^2 + (y_1 - y_2)^2$
- 5 Crie uma struct para armazenar os dados dos jogadores de um time do vôlei (nome, idade, posição e número\_camisa). Crie duas estruturas: uma para armazenar os jogadores titulares e outra para os reservas. Construa um programa no qual o usuário possa cadastrar os jogadores (6 titulares e 4 reservas) e exibir um relatório completo do time (use funções e ponteiros quando possível, e um menu para as opções do usuário).
- 6 Construa um programa que tenha uma estrutura para cadastrar alunos, contendo o nome do aluno, a matrícula e a turma. Cadastre os dados de um aluno e exiba-os em tela por acesso direto à struct. Após isso, exiba-os novamente, mediante uso de um ponteiro para a struct.
- 7 Escreva um programa para controlar um estacionamento, que use uma struct para armazenar a placa e o modelo de um carro, bem como o horário de entrada e saída do estacionamento (horas e minutos). Defina e preencha uma variável do tipo da struct para 1 carro, e mostre os dados do carro junto com o valor devido ao estacionamento (preços: R\$5,00 a primeira hora, R\$2,00 para cada hora extra e R\$1,00 para tempo excedente inferior a uma hora).

**Bom estudo!**