

<b>Disciplina:</b> Técnicas de Programação para Engenharia I	<b>Código:</b> TI0044	<b>Prática:</b> 05
<b>Assunto:</b> Manipulação de tipos de dados definidos pelo usuário – <i>structs</i>		

1. Escreva um programa fazendo o uso de *struct's*. Você deverá criar uma *struct* chamada Ponto, contendo apenas a posição x e y (inteiros) do ponto. Declare 2 pontos, leia a posição (coordenadas x e y) de cada um e calcule a distância entre eles através de uma função. Apresente no final a distância entre os dois pontos.

2. Crie uma estrutura chamada retângulo, que possua duas estruturas Ponto – do exercício anterior - (o ponto superior esquerdo e o ponto inferior direito). Faça um programa que receba (via teclado) as informações acerca de um retângulo (as coordenadas dos dois pontos), e informe dados interessantes sobre o retângulo, como a área, o comprimento da diagonal e o comprimento de cada aresta.

3. Crie uma estrutura chamada círculo que possua duas estruturas Ponto – exercício 1 – que correspondem ao centro e a um ponto sobre a circunferência. No programa a ser feito você deve receber as coordenadas dos pontos pelo teclado e retornar o comprimento da circunferência e a área do círculo.

4. Uma rede de farmácias possui quatro lojas que passam diariamente informações à administração, como estoque (em unidades), arrecadação do dia e gastos do dia. A administração então identifica qual o lucro total, que lojas tiveram rendimento negativo e quais precisam de reposição de estoque (o estoque ideal é de 1000 unidades). Com o auxílio de estruturas, crie um programa que faça os cálculos da administração.

5. Considere que uma empresa precisa armazenar os seguintes dados de um cliente:

- Nome completo com no máximo 50 caracteres;
- renda mensal do cliente;
- ano de nascimento;
- possui ou não carro.

Defina um tipo e uma estrutura para armazenarem estes dados e escreva um programa que leia estes dados armazene-os em uma variável e em seguida os imprima.

6. Considerando a mesma estrutura do exercício anterior, escreva um programa que leia os dados de 100 clientes e imprima:

1. quantos clientes têm renda mensal acima da média;
2. quantos clientes têm carro;
3. quantos clientes nasceram entre 1960 (inclusive) e 1980 (exclusive).

7. Escreva um programa que solicite o nome e telefone de uma pessoa e grave essas informações num vetor de uma estrutura que contem esses dados (nome e telefone). O programa deve ter um menu com três opções apenas: uma que adiciona um novo dado, outra que lista todos os dados atualmente armazenados na memória e outra que sai do programa. Esse vetor de estrutura deve ter apenas 10 elementos e fornecer uma mensagem de erro caso o usuário tente adicionar mais pessoas que este máximo permitido.

8. Um ponto no espaço R3 pode ser representado por valores pertencentes ao conjunto dos números reais formando uma trinca ordenada (x,y,z). Um Cubo pode ser definido por oito pontos em R3. Defina os tipos Ponto3D e Cubo, este último em função do primeiro. Escreva um programa em C que leia do usuário os oito vértices e determine se os valores dados podem ser vértices de um cubo (considere sempre as arestas do cubo paralelas aos eixos x, y e z).

9. Quando se faz uma chamada telefônica, a central responsável pela chamada registra automaticamente numa *tabela de ligações* os seguintes dados: (a) número do telefone de quem fez a chamada; (b) o número de telefone chamado; (c) horário de início da chamada; e (d) horário de fim da chamada. A partir destes dados, o computador é capaz de calcular o preço da ligação e quem é o dono da linha. Defina em C esta tabela de ligações. Para isto defina o tipo de dados Telefone composto dos seguintes campos: o prefixo da área (2 dígitos) e o restante do número (8 dígitos). Um instante no tempo pode ser representado por uma Data (dia, mês e ano) e uma Hora (horas, minutos e segundos).

10. Comunicação em sistemas digitais se dá através de protocolos de comunicação, onde as duas partes comunicantes entram em acordo sobre a forma e o significado dos dados a serem enviados. Existem protocolos que usam *datagramas* que são seqüências de bytes contendo os seguintes campos: (a) número do datagrama (inteiro); (b) o tamanho da mensagem em bytes (inteiro); (c) a mensagem propriamente dita (512 caracteres no máximo); e (d) um *checksum* para verificar a exatidão dos dados (inteiro). Defina o tipo de dado Datagrama em C.