UNIRIO – Universidade Federal do Estado do Rio de Janeiro
Bacharelado em Sistemas de Informação
Técnicas de Programação 1 – 01/2020
Prof. Jobson Massollar

8ª Lista de Exercícios
Structs

1. Crie a struct Ponto para representar um ponto no plano cartesiano. Em seguida, crie a struct Circulo para representar um círculo (formado por um ponto central e um raio). Por fim, leia dois círculos e informe se esses círculos colidem. A colisão deverá ser verificada por uma função booleana `colide`.

2. Crie uma struct Ponto para representar um ponto no plano cartesiano. Em seguida, crie a struct Retangulo para representar um retângulo (formado por um ponto superior esquerdo e outro ponto inferior direito). Por fim, os dados de um retângulo e sua área e perímetro. Essas informações deverão ser calculadas pelas funções `area` e `perimetro`, respectivamente.

3. Crie um programa que controla o consumo de energia dos eletrodomésticos de uma casa. Leia até 10 eletrodomésticos com nome, potência (real em kW) e tempo ativo por dia (real, em horas). Ao final leia um tempo t (em dias) e apresente: nome, potência, horas de uso diário, consumo diário e consumo relativo (percentual) de cada eletrodoméstico nesse período e o total de Kw consumido no período.

4. Crie um programa que lê os dados de um estacionamento: placa do carro, modelo e hora de entrada e de saída, com horas e minutos. Ao final, imprima os dados lidos com o valor a ser pago pelo estacionamento:
   a) Primeira hora: R$ 5,00
   b) Hora extra: R$ 2,00
   c) Horas incompletas deverão ter cobrança proporcional

5. Crie um programa que faz o gerenciamento das contas de um banco. Inicialmente devem ser criadas n contas com: número da conta, nome do correntista e saldo inicial. Em seguida, apresente um menu com as opções 1-Sacar, 2-Depositar, 3-Consultar saldo e 4-Sair. Para cada operação solicite o número da conta, e para as operações 2 e 3 solicite o valor a sacar ou depositar e imprima o valor do saldo resultante. As operações só podem ser realizadas se a conta existir e o saque só pode ser efetivado caso haja saldo suficiente. Crie funções para a entrada de dados e para cada uma das operações. O programa só termina quando o usuário selecionar a opção 4.

6. Crie um programa para gerenciar o estoque de um fornecedor. Para tal, leia dados de n produtos, onde n é fornecido pelo usuário: código, nome, preço unitário e quantidade. Em seguida, leia dados de k pedidos, onde k é fornecido pelo usuário: número, código do produto e quantidade. Ao final, imprima um relatório com todos os pedidos informando:

   a) Dados do pedido e o nome do produto

   b) Situação do pedido: produto não existe, sem estoque ou atendido.

   c) Valor a pagar (somente se o pedido foi atendido).

   Ao final, deve ser apresentado o total a ser faturado pelo fornecedor com os pedidos atendidos. Atenção: cada vez que um pedido for atendido, a sua quantidade deverá ser abatida do estoque.

7. Crie um programa que faz a leitura dos dados dos alunos de uma turma com matrícula, nome, nota1, nota2 e nota3. As notas devem ter valor de 0 a 10 e, se o usuário digitar -1 significa que o aluno faltou àquela prova. A leitura dos dados deverá parar quando o usuário digitar matrícula zero, mas a turma não pode ter mais de 30 alunos. Ao final, imprima a lista de alunos com matrícula, nome, as 3 notas (caso o aluno tenha faltado à prova deverá ser impressa a letra 'F' no local da nota), a média do aluno e a situação ("Aprovado" ou "Reprovado"):

   a) A média do aluno é calculada como a média aritmética das duas maiores notas ou a nota dividida por 2, se ele compareceu a apenas uma prova. Se o aluno faltou às 3 provas a média dele é zero.

   b) Estará aprovado o aluno com média ≥ 6.0.

   Crie uma função para leitura dos dados, outra para impressão e uma terceira para calcular a média de um aluno.

Desafios

8. Crie um programa para cadastrar empresas (máximo de 10 empresas) e seus respectivos funcionários. Cada empresa tem CNPJ, razão social e uma lista com, no máximo, 30 funcionários. Cada funcionário tem CPF, nome, cargo e salário. No cadastro deve ser solicitado: CNPJ, razão social (somente se a empresa já não tiver sido cadastrada anteriormente), CPF, nome, cargo e salário. A entrada termina quando o usuário digitar CNPJ = 0. Ao final, imprimir a lista de empresas e, para cada uma delas, a lista de respectivos funcionários. Atente para as seguintes regras:

   a) Não pode haver dois funcionários cadastrados com o mesmo CPF.

   b) Se um funcionário já está cadastrado em uma empresa, ele não pode ser cadastrado em outra e o usuário deve receber uma mensagem avisando em que empresa esse funcionário já está cadastrado.

9. Crie um programa para gerenciar a consulta de livros em uma biblioteca. Inicialmente leia os dados de n livros com: título, nome do autor e ano. Em seguida, apresente um menu com as opções 1-Consultar por título, 2-Consultar por autor e 3-Sair. A consulta pode ser feita com apenas parte do título ou nome (sem considerar diferença entre maiúsculas e minúsculas), e devem ser impressos

todos os livros que combinam com o dado fornecido. Por exemplo: se um livro se chama "Programacao C" e o outro "Gramatica", a busca por "GRAMA" vai encontrar os dois livros. O programa deverá terminar somente quando o usuário selecionar a opção 3.

10. Uma fração é representada por dois valores inteiros (numerador e denominador). Crie a struct Fracao e implemente uma função para criar uma fração, além de mais quatro funções para as operações básicas com frações (soma, subtração, multiplicação e divisão). As frações devem ser representadas sempre na forma mais reduzida possível, ou seja, as funções devem usar MDC e MMC para reduzir o numerador e o denominador ao máximo. Dica: trabalhe sempre com denominador positivo. Assim, se numerador e denominador forem negativos, transforme ambos em positivos, e se numerador for positivo e denominador for negativo, inverta o sinal deixando numerador negativo e denominador positivo. Para testar as funções implementadas, leia duas frações e imprima os resultados das quatro operações.

    Dicas:

    a) MDC e MMC são sempre valores positivos. Logo, se os parâmetros do MDC ou MMC forem negativos, converta-os para positivo para realizar o cálculo.
    b) A função de MDC já foi implementada no exercício 9 da lista 7. Adapte-a para esse exercício para tratar parâmetros negativos.
    c) MMC(x, y) = (x * y) / MDC(x, y)