

Lista de Exercícios - Semanas 3

Estruturas Condicionais

Introdução a Técnicas de Programação (2025.2)

Problema 1 - Classificação de IMC

Modifique o programa do Problema 1 para que, além de calcular o IMC, também classifique o resultado de acordo com a tabela:

- Abaixo de 18.5: "Abaixo do peso"
- Entre 18.5 e 24.9: "Peso normal"
- Entre 25.0 e 29.9: "Sobrepeso"
- 30.0 ou mais: "Obesidade"

Problema 2 - Calculadora de energia elétrica

Uma companhia de energia cobra da seguinte forma:

- Residencial: R\$ 0,60 por kWh
- Comercial: R\$ 0,48 por kWh
- Industrial: R\$ 1,29 por kWh

Além disso, há uma taxa fixa de R\$ 15,00 para todos os tipos. Crie um programa que leia o consumo em kWh e o tipo de consumidor (R para residencial, C para comercial, I para industrial) e calcule o valor total da conta.

Problema 3 - Sistema de notas

Desenvolva um programa que leia três notas de um aluno e calcule sua média. O programa deve:

- Calcular a média aritmética
- Determinar se o aluno foi aprovado ($\text{média} \geq 7.0$), reprovado ($\text{média} < 4.0$) ou está em recuperação ($4.0 \leq \text{média} < 7.0$)
- Se estiver em recuperação, calcular qual nota precisa tirar na prova final para ser aprovado ($\text{média final} \geq 5.0$, onde $\text{média final} = (\text{média} + \text{nota final})/2$)

Problema 4 - Jogo: pedra, papel, tesoura

Crie um programa que simule uma partida de "pedra, papel, tesoura". O programa deve:

- Ler a jogada do jogador 1 (P para pedra, A para papel, T para tesoura)
- Ler a jogada do jogador 2 (P para pedra, A para papel, T para tesoura)
- Determinar e exibir o vencedor seguindo as regras:
 - Pedra vence Tesoura
 - Tesoura vence Papel
 - Papel vence Pedra
 - Jogadas iguais resultam em empate

Problema 5 - Calculadora de desconto progressivo

Uma loja oferece descontos progressivos baseados no valor da compra:

- Até R\$ 100,00: sem desconto
- De R\$ 100,01 a R\$ 500,00: 10% de desconto
- De R\$ 500,01 a R\$ 1000,00: 15% de desconto
- Acima de R\$ 1000,00: 20% de desconto

Crie um programa que leia o valor da compra e calcule:

- O valor do desconto
- O valor final a ser pago
- A porcentagem de desconto aplicada

Problema 6 - Diagnóstico médico simples

Desenvolva um programa que ajude em um diagnóstico médico básico baseado em sintomas. O programa deve perguntar:

- Tem febre? (S/N)
- Tem dor de cabeça? (S/N)
- Tem dor no corpo? (S/N)
- Tem tosse? (S/N)

Com base nas respostas, o programa deve sugerir:

- Febre + Dor de cabeça + Dor no corpo: "Possível gripe"
- Tosse + Febre: "Possível resfriado"
- Apenas dor de cabeça: "Possível enxaqueca"
- Apenas febre: "Consulte um médico"
- Nenhum sintoma: "Você parece estar bem"
- Qualquer outra combinação: "Consulte um médico para avaliação"

Problema 7 - Sistema de equações do 2º grau

Crie um programa que resolva equações do segundo grau ($ax^2 + bx + c = 0$). O programa deve:

- Ler os coeficientes a, b e c
- Verificar se é uma equação do segundo grau ($a \neq 0$)
- Calcular o discriminante ($\Delta = b^2 - 4ac$)
- Determinar e exibir as raízes conforme o caso:
 - Se $\Delta > 0$: duas raízes reais distintas
 - Se $\Delta = 0$: uma raiz real
 - Se $\Delta < 0$: não possui raízes reais

Fórmula das raízes: $x = (-b \pm \sqrt{\Delta}) / (2a)$

Problema 8 - Validador de triângulos

Desenvolva um programa que leia três valores representando os lados de um triângulo e determine:

- Se os valores podem formar um triângulo (soma de dois lados sempre maior que o terceiro)
- Se formar um triângulo, classificá-lo como:
 - Equilátero (todos os lados iguais)
 - Isósceles (dois lados iguais)
 - Escaleno (todos os lados diferentes)
- Também classifique quanto aos ângulos:
 - Retângulo ($a^2 = b^2 + c^2$, onde a é o maior lado)
 - Acutângulo ($a^2 < b^2 + c^2$)
 - Obtusângulo ($a^2 > b^2 + c^2$)

Observações Gerais

1. **Tipos de variáveis:** Use `int` para números inteiros, `float` para números com decimais, e `char` para caracteres únicos.
2. **Entrada e saída:** Use `scanf()` para leitura e `printf()` para exibição. Lembre-se dos especificadores corretos (`%d`, `%f`, `%c`).
3. **Validação:** Alguns problemas podem exigir validação de entrada (verificar se valores são positivos, se caracteres estão corretos, etc.).
4. **Bibliotecas:** Para usar funções matemáticas como `pow()` e `sqrt()`, inclua `#include <math.h>` e compile com a flag `-lm`.
5. **Boas práticas:**
 - Use nomes descritivos para variáveis
 - Comente seu código quando necessário
 - Mantenha a indentação consistente
 - Teste seu programa com diferentes valores de entrada