



**Universidade Federal do Ceará – Departamento de Engenharia de Teleinformática**  
**Atividade Didática - 2022.2**  
**Introdução à Programação**  
**Prof. Tarcisio Ferreira Maciel, Dr.-Ing.**

**Objetivos:**

- Explorar programas que expressem fórmulas matemáticas nos termos da linguagem de programação.
- Explorar programas que realizem validação de dados de entrada.
- Explorar programas que utilizem estruturas condicionais **if ... else**.

Para cada questão de programação da atividade, comece criando uma aplicação do tipo “Console application” seguindo o passo-a-passo apresentado em laboratório. Em seguida, modifique o programa “Hello World!” para resolver uma questão. Cada projeto deverá ser chamado QuestaoXX, onde XX é o número da questão. Por exemplo, para a Questão 1, o projeto deverá se chamar Questao01. **É mandatório validar todos os dados de entrada.**

**Exercício 1.** (L02) Escreva um programa que calcule e imprima na tela o volume  $V$  e área  $A$  de um cone circular reto de altura  $h > 0$  e raio da base  $r > 0$ , sendo  $h$  e  $r$  lidos pelo teclado. O volume  $V$  e a área  $A$  do cone são dados respectivamente por

$$V = \frac{1}{3} \cdot \pi \cdot r^2 \cdot h \quad \text{e} \quad A = \pi \cdot r \cdot (r + \sqrt{r^2 + h^2}).$$

A raiz quadrada  $\sqrt{x}$  de um número real  $x$  pode ser calculada usando a função **sqrt(x)** da biblioteca **cmath**. O valor de  $\pi$  é pré-definido pela constante **M\_PI** da mesma biblioteca. Realize a validação dos dados de entrada.

**Exercício 2.** (L02) O volume  $V$  e área  $A$  de um tetraedro regular (pirâmide triangular com todas as arestas iguais) são dados respectivamente por

$$V = \frac{1}{3} \frac{a^2 \sqrt{3}}{4} \frac{a \sqrt{6}}{3} \quad \text{e} \quad A = a^2 \sqrt{3},$$

onde  $a > 0$  é o comprimento da aresta da pirâmide. A raiz quadrada  $\sqrt{x}$  de um número real  $x$  pode ser calculada usando a função **sqrt(x)** da biblioteca **cmath**. A potenciação  $x^y$  de um número real  $x$  por um número real  $y$  pode ser calculada usando a função **pow(x, y)** da mesma biblioteca. Realize a validação dos dados de entrada.

**Exercício 3.** (L02) Faça um programa que calcule e imprima a área  $A$  de um triângulo em função das coordenadas de seus três vértices no plano  $xy$ . Assegure-se de que os três pontos formam de fato um triângulo. **Sugestão:** utilize o cálculo de distância entre pontos juntamente com a fórmula de Herão para calcular a área do triângulo.

**Exercício 4.** (L02) Um sistema de equações lineares do tipo:

$$ax + by = c$$

$$dx + ey = f$$

pode ser resolvido utilizando a regra de Cramer conforme mostrado abaixo:

$$x = \frac{ce - bf}{ae - bd},$$
$$y = \frac{af - cd}{ae - bd}.$$

Escreva um programa em C++ que lê os valores de **a, b, c, d, e e f** e que calcula e mostra os valores de **x e y**.

**Exercício 5.** (L03) Depois da liberação do governo para as mensalidades dos planos de saúde, as pessoas começaram a fazer pesquisas para descobrir um bom plano, não muito caro. Um vendedor de um plano de saúde apresentou a tabela a seguir. Crie um programa que entre com a idade de uma pessoa e imprima o valor que ela deverá pagar, segundo a seguinte tabela:

Idade	Valor
Até 10 anos	R\$ 30,00
Acima de 10 até 29 anos	R\$ 60,00
Acima de 29 até 45 anos	R\$ 120,00
Acima de 45 até 59 anos	R\$ 150,00
Acima de 59 até 65 anos	R\$ 250,00
Maior que 65 anos	R\$ 400,00

**Exercício 6.** (L03) Escreva um programa que leia um peso na Terra e o número de um planeta e imprima o valor do seu peso neste planeta. A relação de planetas é dada a seguir juntamente com o valor das gravidades relativas à Terra:

Número	Gravidade Relativa	Planeta
1	0,37	Mercúrio
2	0,88	Vênus
3	0,38	Marte
4	2,64	Júpiter
5	1,15	Saturno
6	1,17	Urano

**Exercício 7.** (L03) [1] Uma caixa de biscoitos pode conter 24 biscoitos e um contêiner pode conter 75 caixas de biscoitos. Escreva um programa que peça ao usuário para inserir o número total de biscoitos, o número de biscoitos em uma caixa e o número de caixas de biscoitos em um contêiner. O programa então exibe o número de caixas e o número de contêineres para enviar os biscoitos. Observe que cada caixa deve conter o número especificado de biscoitos, e cada contêiner deve conter o número especificado de caixas. Se a última caixa de biscoitos contém menos do que o número especificado de biscoitos, você pode descartá-la e imprimir o número de biscoitos descartados. Da mesma forma, se o último contêiner contém menos do que o número especificado de caixas, você pode descartá-lo e imprimir o número de caixas descartadas.

**Exercício 8.** (L03) Escreva um programa que receba os valores reais correspondentes aos tamanhos dos lados  $a$ ,  $b$ , e  $c$  de um triângulo. Verifique se os valores fornecidos formam de fato um triângulo (ou seja, se o tamanho de cada lado é menor que a soma dos outros dois lados) imprimindo uma mensagem adequada sobre isso (por exemplo, “Os lados formam um triângulo”). Caso os lados formem de fato um triângulo, classifique-o como equilátero (os três lados têm tamanhos iguais) e/ou isósceles (pelo menos dois lados têm tamanhos iguais) e/ou retângulo (o quadrado do tamanho de um dos lados é igual à soma dos quadrados dos tamanhos dos outros dois lados) e/ou escaleno (os tamanhos de todos os lados são distintos). Em cada caso, imprima uma mensagem adequada. Lembre-se que se assume aqui que um mesmo triângulo pode ser ao mesmo tempo equilátero e isósceles, ou ainda isósceles e retângulo, ou ainda retângulo e escaleno.

**Exercício 9.** (L01) Escreva um programa que receba o valor de três ângulos  $A$ ,  $B$ ,  $C$  e determine se os mesmos correspondem aos três ângulos internos de um triângulo. Sabe-se que a soma dos ângulos internos de um triângulo é 180 graus. Caso os três ângulos correspondam aos ângulos internos de um triângulo, imprima uma mensagem adequada e em seguida determine o tipo de triângulo de acordo com as regras a seguir:

- Se um dos ângulos for igual à soma dos outros dois ângulos, é formado um triângulo retângulo.
- Se um dos ângulos for maior que a soma dos outros dois ângulos, é formado um triângulo obtusângulo.
- Se cada ângulo for menor que a soma dos outros dois ângulos, é formado um triângulo acutângulo.

Valide os dados de entrada e imprima uma mensagem adequada para cada caso.

**Exercício 10.** (L04) [1] Faça um programa que receba três números inteiros não-negativos **A**, **B**, e **C** e os imprima em ordem decrescente. Por exemplo, se **A** = 2, **B** = 1, e **C** = 8 a saída seria 8 2 1.

**Exercício 11.** (L04) Um número palíndromo é um número que lido da esquerda para a direita ou da direita para a esquerda tem o mesmo valor. Por exemplo, os números 131 e 2332 são palíndromos. Escreva um programa que leia um número inteiro **N** entre 1 e 9999 pelo teclado e determine se ele é ou não palíndromo.

**Exercício 12.** (L04) Escreva um programa que receba três inteiros representando o dia, o mês e o ano de uma data e determine se os mesmos constituem uma data válida. Sabe-se que abril, junho, setembro e novembro têm 30 dias; janeiro, março, maio, julho, agosto, outubro e dezembro têm 31 dias; e fevereiro tem 28 ou 29 dias, dependendo se o ano é ou não bissexto. Um ano é bissexto se: i) é divisível por 4 mas não é divisível por 100, ou; ii) é divisível por 400. Caso a data seja inválida por qualquer motivo imprima a mensagem “Data inválida”. Caso a mesma seja válida imprima “A data dd/mm/aaaa é válida”, em que dd, mm e aaaa são o dia, mês e ano lidos, respectivamente.

**Exercício 13.** (L03) Escreva um programa que leia como inteiros o dia, o mês e o ano de duas datas e determine se as duas estão em ordem cronológica (ou seja, se a primeira data é anterior à segunda data). Imprima uma mensagem adequada em cada caso. Assuma que as duas datas fornecidas serão válidas.

**Exercício 14.** (L04) [2, Q. 4.7] Faça um programa que receba o dia e o mês de uma data e determine a data do dia seguinte. Dados: abril, junho, setembro e novembro têm 30 dias; janeiro, março, maio, julho, agosto, outubro e dezembro têm 31 dias; e fevereiro tem 28. Desconsidere anos bissextos.

**Exercício 15.** Escreva um que leia que pergunte se o usuário deseja: 1) validar um número ISBN-13, ou; 2) gerar o dígito verificador de um número ISBN-13. No primeiro caso, o usuário deverá fornecer um inteiro de 13 dígitos e o programa deverá verificar se o décimo terceiro dígito é válido. No segundo caso, o usuário deverá fornecer um número de 12 dígitos e o programa deverá calcular e exibir o décimo terceiro dígito verificador. Um número ISBN 13 é da forma XXX-X-XXXX-XXX-Y em que Y é o dígito verificador. Considerando como tendo índice zero o dígito X menos significativo do ISBN e como tendo índice onze o dígito X mais significativo do ISBN, o dígito verificador Y é gerado multiplicando-se por 3 os dígitos X de índice par do ISBN e por 1 os dígitos X de índice ímpar. O resto da soma desses produtos dividida por 10 é então subtraído do valor 10, gerando o dígito verificador. Por exemplo:

ISBN	9	7	8	1	8	6	1	9	7	8	7	6
Peso	1	3	1	3	1	3	1	3	1	3	1	3
Produto	9	21	8	3	8	18	1	27	7	24	7	18

151 / 10 = 15, com resto 1. Dígito de verificação é 10 - 1 = 9.

## Referências

- [1] D. S. Malik, *C++ programming: from problem analysis to program design*, 6th ed. Cengage Learning, 2013.
- [2] L. J. Aguilar, *Fundamentos da programação: algoritmos, estruturas de dados e objetos*, 3rd ed. McGraw-Hill, 2008.