

**PRÁTICAS PROPOSTAS**

**Versão: 10/07/2023**

Para cada um dos problemas abaixo, desenvolva um programa na linguagem Dart:

1. Calcular o IMC (índice de massa corporal) de uma pessoa. Produza um alerta em caso de divisão por zero.
2. Construa um programa que calcule a área de um círculo, tendo como entrada o valor do raio, que deve ser positivo. Use o valor da constante “pi” vindo da biblioteca matemática.
3. Construa um programa que calcule para o aluno sua média parcial e informe sua situação parcial (Aprovado, Recuperação ou Reprovado).  
Caso ele esteja em Recuperação, calcule quanto ele precisa tirar na 4ª prova para ser aprovado (média final maior ou igual que 5,0).  
Observação: utilize os pesos e regras da UFERSA.
4. Construa um programa que calcule uma equação do 2º grau.  
Crie uma função com retorno para calcular o delta.  
Crie uma função sem retorno para calcular as raízes.
5. Construa um programa que leia um número inteiro digitado pelo usuário. Caso o número pertença ao intervalo de 1 a 5, exiba o número por extenso. Caso o número não pertença a este intervalo, exiba a mensagem "valor invalido".
6. Construa um programa que leia do usuário um número inicial e um número final. Em seguida, exiba na tela uma sequência com os números desse intervalo informado pelo usuário. Exemplo: caso o usuário entre com os números 4 e 10, o resultado do programa seria: 4 5 6 7 8 9 10
  - a) Construa este programa utilizando a estrutura **while**.
  - b) Construa este programa utilizando a estrutura **do-while**.
  - c) Construa este programa utilizando a estrutura **for**.
7. Construa um programa que leia do usuário um número inicial e um número final. Em seguida, exiba na tela uma sequência apenas com os números ímpares dentro deste intervalo informado pelo usuário. Exemplo: caso o usuário entre com os números 4 e 10, o resultado seria: 5 7 9
  - a) Construa este programa utilizando a estrutura **while**.
  - b) Construa este programa utilizando a estrutura **do-while**.
  - c) Construa este programa utilizando a estrutura **for**.
8. Verificar se a senha, informada durante a execução, é correta. Quando a senha estiver correta, exiba “senha correta” e o programa é encerrado. Quando a senha estiver errada, exiba “senha incorreta”, e permita a entrada novamente da senha, repetindo esse processo até que a senha entrada seja correta.
  - a) Resolva esse problema utilizando a estrutura **while**.
  - b) Resolva esse problema utilizando a estrutura **do-while**.
9. Construa um programa para exibir a tabuada de qualquer número “n” (1 a 9), sendo “n” um número fornecido pelo usuário. Utilize estrutura de repetição.
10. Construa um programa que calcule o rendimento mensal de um investimento em poupança. Variáveis: investimento inicial, investimento mensal, quantidade de meses, saldo acumulado, taxa de juros mensal, rendimento mensal.
11. Construa um programa que identifique se um número é primo.
12. Construa um programa que calcule o somatório dos números inteiros de um intervalo, definido por um número inicial e um número final. Exemplo: caso as entradas fossem 4 e 9, o resultado seria: 39
13. Construa um programa que calcule o fatorial de um número. Use uma estrutura de repetição. Fatorial:  $n! = n(n-1)!$   
Exemplo:  $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$  ou  $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$
14. Construa um programa que simule uma calculadora. Disponibilize um menu de opções e simule a opção desejada, exibindo novamente o menu, até que o usuário escolha sair. Menu de opções:
  - 1 - potenciação Dica: utilize a função da biblioteca matemática.
  - 2 - raiz quadrada Dica: utilize a função da biblioteca matemática.
  - 3 - fatorial Dica: crie e utilize uma função com a solução da questão anterior.
  - 0 - sair

15. Construa um programa para ler do teclado as dimensões de uma matriz de inteiros e seus elementos. Em seguida, exiba seus elementos.
16. Construa e use uma função que produz um novo vetor de inteiros com a ordem inversa do vetor original passado por parâmetro.
17. Construa um programa que identifique o maior e o menor número de um vetor de inteiros.
18. Construa um programa que inverta a frase digitada pelo usuário. Por exemplo, se string1 for “bom dia”, então string2 será “aid mob”.
19. Construa um programa que registre objetos da classe Pessoa, com os seguintes atributos: nome, peso, altura, imc. Permita que o usuário defina a quantidade de pessoas em tempo de execução. Ao cadastrar uma pessoa, calcule o IMC (Índice de Massa Corporal). Ao final, exiba a lista de pessoas (com seus respectivos atributos).
20. Evolua a questão anterior. Permita que o programa salve o resultado em um arquivo binário. Permita que o programa abra o arquivo binário e exiba o conteúdo na tela.
21. Construa um programa que conte a quantidade de letras de uma palavra.  
Exemplo: “casa” Resultado: c=1 a=2 s=1
22. Construa um programa que conte a quantidade de vezes que as palavras de uma frase aparecem.  
Exemplo: “estude muito sempre sempre estude” Resultado: estude=2 muito=1 sempre=2
23. Evolua a questão anterior. Permita que o programa salve o resultado em um arquivo de texto CSV. Permita que o programa abra o arquivo de texto CSV e exiba o conteúdo na tela.
24. Construa e use uma função para calcular o fatorial de um número utilizando recursividade.
25. Calcule o somatório dos números inteiros do intervalo entre dois números. Resolva o problema com função recursiva.
26. Utilizando o recurso de programação genérica, construa um programa que tenha funcionalidades de um CRUD (create, read, update e delete). Teste seu programa com as classes Integer e Pessoa (que você mesmo deve criar).
27. Utilizando o recurso de programação genérica, implemente a estrutura de dados “pilha”.
28. Construa um método que valide se uma string possui seus parênteses colocados corretamente. A string de entrada é válida se a quantidade de parênteses abertos for a mesma dos parênteses fechados. Dica: utilize uma “pilha” para resolver o problema.  
Exemplo: entrada: str = "()" saída: true Exemplo: entrada: str = "()()" saída: true  
Exemplo: entrada: str = "(a)b" saída: true Exemplo: entrada: str = "(a)(b)" saída: true  
Exemplo: entrada: str = "(((" saída: false Exemplo: entrada: str = "(a)(b)" saída: false  
Exemplo: entrada: str = ")((" saída: false Exemplo: entrada: str = ")a()b(" saída: false
29. Construa um programa que receba duas strings, s1 e s2, e retorne true se s2 for um anagrama de s1.  
Exemplos de anagrama: PATO, PATO, TOPA e OPTA.
30. Construa um programa para implementar, apenas, um codificador de Huffman.  
Exemplo: Entrada: ABRACADABRA Saída: 1 01 000 1 0010 1 0011 1 01 000 1