

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

1. Exiba seu nome na tela.
2. Inicialize variáveis adequadas para: sua idade, sua altura, primeira letra do seu nome, seu nome completo. Depois exiba os dados na tela.
3. Atribua com valores vindos do teclado variáveis adequadas para: sua idade, sua altura, primeira letra do seu nome, seu nome completo. Depois exiba os dados na tela.
4. Receba do teclado dois números inteiros, calcule e exiba os resultados das seguintes operações: adição, subtração, multiplicação, quociente da divisão e o resto da divisão.
5. Calcular e exibir a média aritmética de três números inteiros.
6. Calcular e exibir a média aritmética de três números reais.
7. Construa um programa que gerencie a conta de uma pizzaria.
Preço do refrigerante: R\$ 1,50. Preço da fatia de pizza: R\$ 3,00. Taxa do garçom: 10%.
Receba do usuário: a quantidade de refrigerantes, a quantidade de fatias e a quantidade de pessoas na mesa.
Calcule e exiba: o total sem a taxa, o total com a taxa, o rateio por pessoa com a taxa.
8. Calcular uma divisão entre dois números reais, mas produzir um alerta em caso de divisão por zero.
9. Calcular o IMC (índice de massa corporal) de uma pessoa.
10. Construa um programa que calcule a área de um círculo, tendo como entrada o valor do raio, que deve ser positivo.
11. Construa um programa que leia um número inteiro e identifique se ele é par ou ímpar.
12. Construa um programa que simule uma transferência bancária, entre duas contas. Primeiro, inicialize cada conta com R\$ 100. Depois, permita que o usuário defina quanto deve transferir, da conta1 para a conta2, porém a transferência só deve ser realizada caso haja saldo suficiente.
13. Distinguir, com base na média parcial do aluno, se ele está aprovado, reprovado ou na final. Aplique as regras da UFERSA.
14. 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.
15. 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".
16. 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.
17. 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**.
18. 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**.
19. 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**.
20. 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
21. Construa um programa que calcule o fatorial de um número. Use uma estrutura de repetição. Fatorial: $n! = n \times (n - 1)!$
Exemplo: $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$ ou $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$
22. Construa um programa que identifique se um número é primo.

23. 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 <i>pow</i> da biblioteca <i>math.h</i> |
| 2 - raiz quadrada | Dica: utilize a função <i>sqrt</i> da biblioteca <i>math.h</i> |
| 3 - fatorial | Dica: crie e utilize uma função com a solução da questão anterior. |
| 0 – sair | |
24. Construa um programa pra ler e exibir um vetor de inteiros.
25. Construa um programa pra ler e exibir um vetor de inteiros. Use *alocação dinâmica de memória* pra que o usuário defina o tamanho do vetor.
26. Construa um programa pra ler e exibir uma matriz de inteiros.
27. Construa um programa pra ler e exibir uma matriz de inteiros. Use *alocação dinâmica de memória* pra que o usuário defina o tamanho da matriz. OBS: requer o uso de ponteiro de ponteiro.
28. Construa e use uma função que produz um novo vetor de inteiros com a ordem inversa do vetor original passado por parâmetro.
29. Construa um programa que identifique o maior e o menor número de um vetor de inteiros.
30. Construa um programa que mova o número da última posição de um vetor para a primeira posição. Faça isso gradativamente: use uma estrutura de repetição, e em cada iteração do loop mova esse número apenas uma posição, ou seja, troque esse número da posição n por $n-1$.
31. Construa um programa que inverta a frase digitada pelo usuário. Por exemplo, se o usuário digitar “bom dia” o programa exibirá “aid mob”.
32. Evolua a questão anterior de modo que a frase invertida seja armazenada em uma *string* antes de ser exibida.
33. Construa um programa que registre 3 pessoas com os seguintes campos: nome, idade, sexo, peso, altura. Em seguida, exiba a ficha completa de cada uma: nome, idade, sexo, peso, altura e IMC (Índice de Massa Corporal).
34. Evolua a questão anterior utilizando *alocação dinâmica de memória*. Com esse recurso, o usuário poderá definir a quantidade de pessoas que serão registradas durante a execução do programa.
35. Construa um programa que leia e escreva dados em um arquivo CSV (*comma-separated values*).
36. Construa um programa que conte a quantidades de letras de uma palavra.
Exemplo: “casa” Resultado: c=1 a=2 s=1
37. Construa um programa que receba uma string e armazene separadamente cada palavra dessa string.
Exemplo: “estude muito sempre sempre estude” Resultado: “estude” “muito” “sempre” “sempre” “estude”
38. Construa um programa que conte a quantidades de vezes que as palavras de uma frase aparecem.
Exemplo: “estude muito sempre sempre estude” Resultado: estude=2 muito=1 sempre=2
39. Evolua a questão anterior. Utilize *alocação dinâmica de memória* para armazenar os vetores necessários e *leitura de arquivo* para ler o texto de entrada vindo de um arquivo de texto externo.
40. Construa e use uma função para calcular o fatorial de um número. Utilize recursividade.
41. Calcule o somatório dos números inteiros do intervalo entre dois números. No mesmo programa, implemente esse cálculo em duas funções:
- Resolva o problema com recursividade.
 - Resolva o problema com estrutura de repetição.
 - Compare o tempo de execução entre as duas versões. Execute as duas funções com a mesma entrada. Qual delas é a mais rápida? Por quê?