



INTRODUÇÃO AO C++

PENSAMENTO COMPUTACIONAL E INTRODUÇÃO À LINGUAGEM

LISTA DE EXERCÍCIOS

ENGENHARIA DE COMPUTAÇÃO

PRIMEIROS PASSOS

- ① Escreva um programa em C++ que exiba a mensagem “Olá, mundo!” na tela.

Output esperado

Olá, mundo!

- ② Crie um programa em C++ que faça uma saudação personalizada para o usuário.

Para isso, o usuário deve informar o seu nome e o programa deve exibir a saudação personalizada.

Exemplo de input

Digite o seu nome: André

Output esperado

Olá, André!

- ③ Escreva um programa em C++ que imprima o padrão abaixo na tela usando múltiplos comandos de saída:

Output esperado

```
*  
**  
***  
****  
*****
```

- ④ Crie um programa em C++ que recebe um valor numérico digitado pelo usuário, incremente esse número em uma unidade e exiba para o usuário.

Exemplo de input

Digite o valor de N: 5

Output esperado

N+1 = 6

- ⑤ Crie um programa em C++ faz a leitura de dois valores numéricos digitados pelo usuário e troque o valor entre eles.

Exemplo de input

Digite o valor de X: 2

Digite o valor de Y: 7

Output esperado

X = 7

Y = 2

TIPOS DE VARIÁVEIS E OPERAÇÕES

- ⑥ Em C++, existem diferentes tipos de variáveis. Estão entre as mais simples (e mais usadas):

Tipo	Descrição	Exemplo
<code>int</code>	Números inteiros	<code>int x = 10;</code>
<code>float</code>	Números decimais (precisão simples)	<code>float y = 3.14;</code>
<code>double</code>	Números decimais (precisão dupla)	<code>double z = 3.14159;</code>
<code>char</code>	Um único caractere	<code>char c = 'A';</code>
<code>bool</code>	Valores verdadeiro/falso	<code>bool ok = true;</code>

Crie um programa que exibe quantos **bytes** ocupa cada um destes tipos de variáveis.

- 7) Os tipos **float** e **double** armazenam números decimais, mas com diferentes níveis de precisão. O **float** possui aproximadamente 6-7 dígitos de precisão, enquanto o **double** possui cerca de 15-16 dígitos.

Crie um programa que realize a mesma operação matemática com ambos os tipos e compare os resultados. Realize o seguinte cálculo com cada tipo:

$$x = \frac{1.0}{3.0} + \frac{1.0}{3.0} + \frac{1.0}{3.0}$$

O resultado matemático exato é 1.0. O programa deve exibir:

1. O resultado obtido com **float**
2. O resultado obtido com **double**
3. A diferença entre cada resultado e o valor esperado (1.0)

Dica: Defina o resultado esperado como uma constante e calcule o erro absoluto de cada aproximação.

Exemplo de saída esperada

```
Com float: 0.9999999403953552
Com double: 0.9999999999999999
Erro float: 5.9604645e-08
Erro double: 1.1102230e-16
```

- 8) Em C++, cada caractere (**char**) possui um valor inteiro correspondente na tabela ASCII (American Standard Code for Information Interchange). A tabela ASCII associa números de 0 a 127 a caracteres específicos:

- Códigos 48-57: dígitos ‘0’ a ‘9’
- Códigos 65-90: letras maiúsculas ‘A’ a ‘Z’
- Códigos 97-122: letras minúsculas ‘a’ a ‘z’

Por exemplo, o caractere ‘A’ corresponde ao valor inteiro 65.

Crie um programa que declare uma variável do tipo **char** com o valor ‘A’, e exiba:

1. O caractere em si
2. O valor inteiro correspondente (utilizando **type casting** para **int**)

- 9) Crie um programa em C++ capaz de transformar um caractere digitado pelo usuário em maiúsculo.

Exemplo de entrada

Digite um caractere: a

Exemplo de saída

A

- 10) Construa um programa que permita que o usuário informe os 3 lados de um triângulo, A, B e C. Apresente para ele o valor da área do triângulo utilizando a fórmula de Herão.

A fórmula de Herão calcula a área a partir dos três lados do triângulo:

$$A = \sqrt{s(s - a)(s - b)(s - c)}$$

Onde s é o semi-perímetro, calculado por:

$$s = \frac{a + b + c}{2}$$

Exemplo de entrada

Digite o lado A: 3

Digite o lado B: 4

Digite o lado C: 5

Exemplo de saída

Area: 6.0

- ⑪ Elabore um programa capaz de converter a temperatura monitorada em graus Celsius (C), informada por um usuário, para graus Kelvin (K) e Fahrenheit (F).

As fórmulas de conversão são:

- Kelvin: $K = C + 273.15$
- Fahrenheit: $F = \left(C \times \frac{9}{5}\right) + 32$

Exemplo de entrada

Digite a temperatura em Celsius: 25

Exemplo de saída

Kelvin: 298.15

Fahrenheit: 77.0

ESTRUTURAS CONDICIONAIS

- ⑫ Crie um programa que receba três números inteiros do usuário e os exiba em ordem crescente.

Exemplo de entrada

Digite o primeiro número: 25

Digite o segundo número: 10

Digite o terceiro número: 18

Exemplo de saída

Ordem crescente: 10 18 25

- ⑬ Crie um programa que receba um número inteiro do usuário e verifique se ele é par ou ímpar.

Exemplo de entrada 1

Digite um número: 42

Exemplo de saída 1

Par

Exemplo de entrada 2

Digite um número: 17

Exemplo de saída 2

Ímpar

- (14)** Crie um programa que receba um caractere do usuário e o transforme em maiúsculo, mas apenas se for possível capitalizá-lo (ou seja, se for uma letra minúscula).

O programa deve verificar:

- Se o caractere é uma letra minúscula (a-z), convertê-lo para maiúsculo
- Se o caractere já é maiúsculo, exibir uma mensagem informando
- Se for um número ou símbolo, exibir que não é uma letra

Dica: Utilize condicionais (**if-else**) para verificar se o caractere está na faixa de letras minúsculas (valores ASCII 97 a 122).

Exemplo de entrada 1

Digite um caractere: a

Exemplo de saída 1

Maiúsculo: A

Exemplo de entrada 2

Digite um caractere: 5

Exemplo de saída 2

Nao e uma letra.

Exemplo de entrada 3

Digite um caractere: Z

Exemplo de saída 3

Ja esta em maiusculo.

- 15) Tendo como dado de entrada a altura (h) de uma pessoa e seu sexo (M ou F), construa um programa que calcule seu peso ideal, utilizando as seguintes fórmulas:

- Para homens: $(72.7 \times h) - 58$
- Para mulheres: $(62.1 \times h) - 44.7$

O programa deve solicitar a altura e o sexo, e exibir o peso ideal calculado.

Exemplo de entrada 1

```
Digite sua altura em metros: 1.75  
Digite seu sexo (M/F): M
```

Exemplo de saída 1

Peso ideal: 69.225 kg

Exemplo de entrada 2

```
Digite sua altura em metros: 1.60  
Digite seu sexo (M/F): F
```

Exemplo de saída 2

Peso ideal: 54.66 kg

- 16) Crie um programa que resolva equações de segundo grau da forma $ax^2 + bx + c = 0$ utilizando a fórmula de Bhaskara.

A fórmula de Bhaskara é:

$$x = \frac{-b \pm \sqrt{\Delta}}{2a}$$

Onde

$$\Delta = b^2 - 4ac$$

O programa deve:

- Verificar se $a = 0$ (não é equação de segundo grau)
- Calcular o discriminante (Δ)
- Verificar se há raízes reais:
 - Se $\Delta < 0$: equação não possui raízes reais
 - Se $\Delta = 0$: equação possui uma raiz real
 - Se $\Delta > 0$: equação possui duas raízes reais

Exemplo de entrada 1

```
Digite o valor de a: 1  
Digite o valor de b: -5  
Digite o valor de c: 6
```

Exemplo de saída 1

```
Delta: 1  
Raiz 1: 3  
Raiz 2: 2
```

Exemplo de entrada 2

```
Digite o valor de a: 1  
Digite o valor de b: 2  
Digite o valor de c: 5
```

Exemplo de saída 2

```
Delta: -16  
Equacao nao possui raizes reais.
```

- 17) Uma loja de tintas possui tres tipos de latas disponiveis, com precos que incentivam a compra de latas maiores (custo por litro menor):

- Lata de 20 litros: R\$ 180,00 (R\$ 9,00/litro)
- Lata de 12 litros: R\$ 130,00 (R\$ 10,83/litro)
- Lata de 5 litros: R\$ 55,00 (R\$ 11,00/litro)

Cada litro de tinta pinta 3 metros quadrados de parede. Crie um programa que receba a area a ser pintada e calcule a combinacao de latas que minimize o custo total.

O programa deve considerar todas as combinacoes possiveis e escolher a mais economica, garantindo que a quantidade total de tinta seja suficiente (pode haver sobra de tinta).

Exemplo de entrada

Digite o tamanho da area a ser pintada (m²): 85

Exemplo de saida

Litros necessarios: 28.33

Melhor combinacao:

- 1 lata de 20L (R\\$ 180.00)
- 1 lata de 12L (R\\$ 130.00)

Custo total: R\\$ 310.00

Exemplo de entrada 2

Digite o tamanho da area a ser pintada (m²): 30

Exemplo de saida 2

Litros necessarios: 10.00

Melhor combinacao:

- 1 lata de 12L (R\\$ 130.00)

Custo total: R\\$ 130.00

Dica: Teste diferentes combinacoes considerando que pode ser mais barato comprar uma lata maior do que varias menores (por exemplo, 1 lata de 20L custa menos que 4 latas de 5L).

LAÇOS DE REPETIÇÃO

- ⑯ 18 Escreva um programa em C++ que imprima os números de 1 a 10 na tela, um por linha, utilizando um laço de repetição.
- ⑯ 19 Escreva um programa em C++ que calcule e imprima a soma dos números pares de 1 a 100.
- ⑯ 20 Crie um programa que receba um número inteiro N do usuário e exiba todos os números primos de 2 até N.

Um número primo é aquele que possui exatamente dois divisores: 1 e ele mesmo.

Exemplo de entrada

Digite um número: 30

Exemplo de saída

Números primos até 30:
2 3 5 7 11 13 17 19 23 29

Dica avançada: Pesquise sobre o “Crivo de Eratóstenes”, um algoritmo eficiente para encontrar todos os primos até N. Ele funciona marcando os múltiplos de cada número primo encontrado, eliminando-os da lista de candidatos.

- 21) Crie um programa que calcule o fatorial de um número N fornecido pelo usuário.

O fatorial de N (representado como $N!$) é o produto de todos os números inteiros positivos de 1 até N:

$$N! = 1 \times 2 \times 3 \times \dots \times N$$

Por definição: $0! = 1$

Exemplo de entrada

Digite um número: 5

Exemplo de saída

$5! = 120$

Exemplo de entrada 2

Digite um número: 0

Exemplo de saída 2

$0! = 1$

Desafio extra: Após resolver com um laço de repetição, pesquise sobre recursão e tente implementar uma versão recursiva do cálculo do fatorial. Uma função recursiva chama a si mesma com um valor menor até atingir o caso base.

- ② Crie um programa que calcule o Máximo Divisor Comum (MDC) entre dois números inteiros positivos fornecidos pelo usuário.

O MDC é o maior número que divide ambos os valores sem deixar resto.

Exemplo de entrada

```
Digite o primeiro número: 48  
Digite o segundo número: 18
```

Exemplo de saída

```
MDC(48, 18) = 6
```

Dica: Utilize o Algoritmo de Euclides, que é muito eficiente:

1. Divida o maior número pelo menor
2. Se o resto for 0, o MDC é o menor número
3. Se não, substitua o maior pelo menor e o menor pelo resto
4. Repita até o resto ser 0

- 23) Crie um programa que calcule o N-ésimo termo da sequência de Fibonacci.

A sequência de Fibonacci é definida da seguinte forma:

- Os dois primeiros termos são: $F(0) = 0$ e $F(1) = 1$
- Cada termo subsequente é a soma dos dois anteriores: $F(n) = F(n - 1) + F(n - 2)$

A sequência começa: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Exemplo de entrada

Digite o valor de N: 8

Exemplo de saída

`Fibonacci(8) = 21`

Exemplo de entrada 2

Digite o valor de N: 20

Exemplo de saída 2

`Fibonacci(20) = 6765`

Desafio extra: Após resolver com um laço de repetição, implemente uma versão recursiva. Compare o desempenho de ambas as abordagens para valores grandes de N. Você notará que a versão recursiva fica muito lenta devido à repetição de cálculos.

(24) Crie um programa que calcule o N-ésimo Número de Bernoulli.

Os Números de Bernoulli são uma sequência de números racionais importantes em matemática, especialmente na teoria dos números e nas expansões de séries de Taylor.

São definidos pela fórmula de recorrência:

- $B_0 = 1$
- Para $n \geq 1$: $\sum_{k=0}^n \binom{n+1}{k} B_k = 0$

Os primeiros números de Bernoulli são: $B_0 = 1$, $B_1 = -\frac{1}{2}$, $B_2 = \frac{1}{6}$, $B_4 = -\frac{1}{30}$, $B_6 = \frac{1}{42}$, ... (todos os ímpares maiores que 1 são zero)

Exemplo de entrada

Digite o valor de N: 6

Exemplo de saída

$B(6) = 1/42$

Valor aproximado: 0.0238095

Exemplo de entrada 2

Digite o valor de N: 10

Exemplo de saída 2

$B(10) = 5/66$

Valor aproximado: 0.0757576

Dica: Utilize a fórmula de recorrência calculando iterativamente. Para cada n , resolva a equação isolando B_n : $B_n = -\frac{1}{n+1} \times \sum_{k=0}^{n-1} \binom{n+1}{k} B_k$

Desafio extra: Implemente uma versão que armazene os números de Bernoulli já calculados (memoização) para evitar recalcular valores repetidamente.

- ② A **Transformada de Fourier** é uma das ferramentas matemáticas mais importantes da computação moderna. Ela permite decompor qualquer sinal complexo em uma soma de ondas senoidais simples, revelando as “frequências” que compõem o sinal.

Crie um programa que aproxime uma onda quadrada usando a Série de Fourier. Uma onda quadrada pode ser aproximada pela soma de senos ímpares:

$$f(x) = \frac{4}{\pi} \times \sum_{n=1,3,5\dots}^N \frac{\sin(nx)}{n}$$

Onde N é o número de harmônicos utilizados na aproximação.

O programa deve:

- Receber o número de harmônicos N
- Calcular o valor da série em vários pontos de um período (por exemplo, de 0 a 2π)
- Exibir os valores calculados

Exemplo de entrada

Digite o número de harmônicos: 5

Exemplo de saída (parcial)

```
x = 0.00: f(x) = 0.0000
x = 0.31: f(x) = 0.9355
x = 0.63: f(x) = 1.1781
x = 0.94: f(x) = 1.1781
...
```

Desafio: Aumente o número de harmônicos (N) e observe como a aproximação se torna mais próxima de uma onda quadrada perfeita. O que acontece próximo aos pontos de descontinuidade? (Este fenômeno é conhecido como **Efeito Gibbs**)

ARRAYS

- 26 Crie um programa que trabalhe com um array de 5 números inteiros. O programa deve solicitar ao usuário que digite os 5 valores, armazená-los no array e depois exibir os valores na ordem inversa (do último para o primeiro).

Este exercício introduz conceitos fundamentais de manipulação de arrays: inicialização, acesso por índice e iteração em ordem reversa.

Exemplo de entrada

Digite 5 números inteiros:

Número 1: 10

Número 2: 20

Número 3: 30

Número 4: 40

Número 5: 50

Exemplo de saída

Valores na ordem inversa: 50 40 30 20 10

- (27) Desenvolva um programa que calcule a média aritmética de 10 notas escolares. O programa deve solicitar as 10 notas ao usuário, armazená-las em um array e calcular a média final.

Exemplo de entrada

Digite 10 notas:

Nota 1: 7.5

Nota 2: 8.0

Nota 3: 6.5

Nota 4: 9.0

Nota 5: 7.0

Nota 6: 8.5

Nota 7: 6.0

Nota 8: 7.5

Nota 9: 8.0

Nota 10: 9.5

Exemplo de saída

Média aritmética: 7.75

- 28) Implemente um algoritmo de busca linear (sequential search) em um array. O programa deve ler 10 números inteiros em um array e, em seguida, solicitar um número x para busca. O programa deve verificar se x existe no array e informar sua posição (índice), ou indicar que não foi encontrado.

A busca linear percorre o array elemento por elemento até encontrar o valor desejado ou chegar ao final. Sua complexidade é $O(n)$ no pior caso.

Exemplo de entrada

```
Digite 10 números: 5 12 3 8 21 7 15 2 9 18  
Digite o número a buscar: 15
```

Exemplo de saída

Número 15 encontrado na posição 6.

Exemplo de entrada 2

```
Digite 10 números: 5 12 3 8 21 7 15 2 9 18  
Digite o número a buscar: 100
```

Exemplo de saída 2

Número 100 não encontrado no array.

- (29) Trabalhe com matrizes bidimensionais criando um programa que processe uma matriz 3×3 de números inteiros. O programa deve solicitar os 9 valores ao usuário (linha por linha) e calcular a soma dos elementos da diagonal principal.

A diagonal principal de uma matriz quadrada são os elementos onde o índice da linha é igual ao índice da coluna ($M[i][i]$).

Exemplo de entrada

Digite os elementos da matriz 3x3:

Linha 0: 1 2 3

Linha 1: 4 5 6

Linha 2: 7 8 9

Exemplo de saída

Matriz digitada:

1 2 3

4 5 6

7 8 9

Diagonal principal: 1, 5, 9

Soma da diagonal: 15

- 30) Implemente um programa que calcule o determinante de uma matriz 3×3 de números reais. O determinante é um valor escalar que fornece informações importantes sobre a matriz, como se ela é invertível ou não.

Para uma matriz 3×3 :

$$A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

O determinante é calculado por:

$$\det(A) = a(ei - fh) - b(di - fg) + c(dh - eg)$$

Ou, expandindo:

$$\det(A) = aei + bfg + cdh - ceg - bdi - afh$$

Exemplo de entrada

Digite os elementos da matriz 3×3 :

Linha 0: 1 2 3

Linha 1: 4 5 6

Linha 2: 7 8 9

Exemplo de saída

Matriz:

1 2 3

4 5 6

7 8 9

Determinante: 0

Exemplo de entrada 2

Digite os elementos da matriz 3×3 :

Linha 0: 2 -1 3

Linha 1: 1 4 2

Linha 2: 0 3 1

Exemplo de saída 2

Matriz:

2 -1 3

1 4 2

0 3 1

Determinante: -21

- (31) Implemente um programa que realize a multiplicação de duas matrizes.
- gráfica, machine learning, processamento de imagens e muitas outras áreas.

Condição de validade: A multiplicação $A \times B$ só é possível quando o número de colunas da matriz A é igual ao número de linhas da matriz B .

Se A tem dimensão $m \times n$ e B tem dimensão $n \times p$, o resultado C terá dimensão $m \times p$, onde cada elemento é calculado por:

$$C[i][j] = \sum_{k=0}^{n-1} A[i][k] \times B[k][j]$$

O programa deve:

1. Ler as dimensões das duas matrizes
2. Verificar se a multiplicação é válida
3. Se válida, ler os elementos e calcular o produto
4. Se inválida, exibir mensagem de erro apropriada

Exemplo de entrada (válida)

Dimensões da matriz A (linhas colunas): 2 3

Elementos da matriz A:

Linha 0: 1 2 3

Linha 1: 4 5 6

Dimensões da matriz B (linhas colunas): 3 2

Elementos da matriz B:

Linha 0: 7 8

Linha 1: 9 10

Linha 2: 11 12

Exemplo de saída

Matriz A (2x3):

1 2 3

4 5 6

Matriz B (3x2):

7 8

9 10

11 12

Resultado A x B (2x2):

58 64

139 154

Exemplo de entrada (inválida)

Dimensões da matriz A (linhas colunas): 2 3

Dimensões da matriz B (linhas colunas): 2 3

Exemplo de saída

Erro: Não é possível multiplicar uma matriz 2x3 por uma matriz 2x3.
O número de colunas de A (3) deve ser igual ao número de linhas de B (2).

Algoritmo: Utilize três loops aninhados:

- Loop externo: percorre as linhas de A
- Loop do meio: percorre as colunas de B
- Loop interno: calcula a soma dos produtos (produto escalar da linha i de A com a coluna j de B)

STRINGS

- ③ Crie um programa que inverta uma string fornecida pelo usuário.

O programa deve receber uma string e retornar os caracteres na ordem inversa.

Exemplo de entrada

Digite uma string: programacao

Exemplo de saída

String invertida: oacammargorp

- ③ Crie um programa que substitua cada letra em uma string pela letra seguinte no alfabeto (a vira b, p vira q, z vira a).

Números e outros caracteres devem permanecer inalterados.

Exemplo de entrada

Digite uma string: teste123

Exemplo de saída

Resultado: uftuf123

- ④ Crie um programa que capitalize a primeira letra de cada palavra em uma string.

As palavras devem estar separadas por apenas um espaço.

Exemplo de entrada

Digite uma frase: exercicios de strings

Exemplo de saída

Resultado: Exercicios De Strings

- 35) Crie um programa que encontre a maior palavra em uma string.

Caso existam várias palavras com o mesmo tamanho máximo, retorne a primeira encontrada.

Exemplo de entrada

Digite uma frase: C++ eh uma linguagem de proposito geral.

Exemplo de saída

Maior palavra: proposito

- 36) Crie um programa que ordene as letras de uma string em ordem alfabética.

Números e símbolos de pontuação não devem ser incluídos na ordenação (podem ser removidos ou mantidos nas posições originais).

Exemplo de entrada

Digite uma string: computador

Exemplo de saída

Ordenado: acdmooprtlu

- 37) Crie um programa que verifique se os caracteres ‘e’ e ‘g’ estão separados por exatamente 2 posições em algum lugar da string.

O programa deve contar quantas vezes essa condição ocorre.

Exemplo de entrada

Digite uma string: legumes

Exemplo de saída

Ocorrencias de ‘e’ e ‘g’ separados por 2 posicoes: 1

Explicação: Na string “legumes”, os caracteres nas posições 1 e 4 são ‘e’ e ‘g’ respectivamente, com exatamente 2 caracteres entre eles.

- 38 Crie um programa que conte todas as vogais (a, e, i, o, u, maiúsculas e minúsculas) em uma string.

Exemplo de entrada

Digite uma string: programacao

Exemplo de saída

Numero de vogais: 5

- 39 Crie um programa que conte todas as palavras em uma string.

Palavras são sequências de caracteres separadas por espaços.

Exemplo de entrada

Digite uma frase: Python

Exemplo de saída

Numero de palavras: 1

Exemplo de entrada 2

Digite uma frase: C++ eh incrivel

Exemplo de saída 2

Numero de palavras: 3

- 40 Crie um programa que verifique se dois caracteres distintos aparecem a mesma quantidade de vezes em uma string.

O programa deve receber a string e os dois caracteres a serem comparados.

Exemplo de entrada

Digite uma string: aabcdeef

Digite o primeiro caractere: a

Digite o segundo caractere: e

Exemplo de saída

Os caracteres 'a' e 'e' aparecem igual numero de vezes: True

- ④1 Crie um programa que verifique se uma string é um palíndromo.

Um palíndromo é uma palavra, frase ou sequência de caracteres que lida da mesma forma de trás para frente (ex: madam, racecar, arara).

O programa deve ignorar espaços, pontuação e diferença entre maiúsculas e minúsculas.

Exemplo de entrada

Digite uma string: arara

Exemplo de saída

E palindromo: True

Exemplo de entrada 2

Digite uma string: olá

Exemplo de saída 2

E palindromo: False

- ④2 Crie um programa que verifique se duas strings são anagramas.

Anagramas são palavras ou frases formadas pelas mesmas letras em ordem diferente (ex: “amor” e “roma”, “listen” e “silent”).

O programa deve ignorar espaços, pontuação e diferença entre maiúsculas e minúsculas.

Exemplo de entrada

Digite a primeira string: amor

Digite a segunda string: roma

Exemplo de saída

São anagramas: True

Exemplo de entrada 2

Digite a primeira string: teste

Digite a segunda string: festa

Exemplo de saída 2

São anagramas: False

- 43) Crie um programa que implemente a compressão Run-Length Encoding (RLE) em uma string.

A compressão RLE substitui sequências repetidas de caracteres pelo caractere seguido da quantidade de repetições (ex: “aaabbbcc” vira “a3b3c2”).

Se um caractere aparece apenas uma vez, mantenha-o sem número.

Exemplo de entrada

Digite uma string: aaabbbcc

Exemplo de saída

Comprimido: a3b3c2

Exemplo de entrada 2

Digite uma string: abcdef

Exemplo de saída 2

Comprimido: abcdef

44 Crie um programa que avalie a força de uma senha.

Uma senha é considerada forte se atender aos seguintes critérios:

- Possuir pelo menos 8 caracteres
- Conter pelo menos uma letra maiúscula (A-Z)
- Conter pelo menos uma letra minúscula (a-z)
- Conter pelo menos um número (0-9)
- Conter pelo menos um caractere especial (! . , ;)

O programa deve exibir quais critérios foram atendidos e a classificação final (Fraca, Média ou Forte).

Exemplo de entrada

Digite uma senha: Senha123!

Exemplo de saída

Critérios atendidos:

- (OK) Mínimo 8 caracteres
- (OK) Letra maiúscula
- (OK) Letra minúscula
- (OK) Número
- (OK) Caractere especial

Classificação: FORTE

Exemplo de entrada 2

Digite uma senha: abc

Exemplo de saída 2

Critérios atendidos:

- (X) Mínimo 8 caracteres
- (X) Letra maiúscula
- (OK) Letra minúscula
- (X) Número
- (X) Caractere especial

Classificação: FRACA

45 Crie um programa que remova espacos desnecessarios de uma string.

O programa deve:

- Remover espacos do inicio e do fim da string
- Substituir multiplos espacos consecutivos por um unico espaco

Exemplo de entrada

Digite uma frase: programacao em C++

Exemplo de saida

Resultado: programacao em C++

Exemplo de entrada 2

Digite uma frase: teste

Exemplo de saida 2

Resultado: teste

FUNÇÕES

46 Crie uma calculadora com funções separadas para cada operação.

Implemente quatro funções:

- `somar(a, b)` - retorna a soma de dois números
- `subtrair(a, b)` - retorna a diferença entre dois números
- `multiplicar(a, b)` - retorna o produto de dois números
- `dividir(a, b)` - retorna a divisão de dois números (verifique divisão por zero!)

O programa principal deve solicitar dois números ao usuário e exibir o resultado de todas as operações.

Exemplo de entrada

```
Digite o primeiro numero: 10  
Digite o segundo numero: 3
```

Exemplo de saída

```
Soma: 13  
Subtracao: 7  
Multiplicacao: 30  
Divisao: 3.33
```

Exemplo de entrada 2 (divisão por zero)

```
Digite o primeiro numero: 10  
Digite o segundo numero: 0
```

Exemplo de saída 2

```
Soma: 10  
Subtracao: 10  
Multiplicacao: 0  
Divisao: Erro! Nao e possivel dividir por zero.
```

(47) Crie um conversor de temperatura com funções especializadas.

Implemente três funções de conversão:

- `celsiusParaFahrenheit(c)` - converte Celsius para Fahrenheit
- `fahrenheitParaCelsius(f)` - converte Fahrenheit para Celsius
- `celsiusParaKelvin(c)` - converte Celsius para Kelvin

Fórmulas:

- Fahrenheit = $(\text{Celsius} \times 9/5) + 32$
- Celsius = $(\text{Fahrenheit} - 32) \times 5/9$
- Kelvin = Celsius + 273.15

O programa deve apresentar um menu para o usuário escolher a conversão desejada.

Exemplo de uso

```
==== Conversor de Temperatura ====
1. Celsius para Fahrenheit
2. Fahrenheit para Celsius
3. Celsius para Kelvin
Escolha uma opção: 1
```

```
Digite a temperatura em Celsius: 25
Resultado: 25°C = 77°F
```

48 Crie uma função que calcule a média ponderada de notas.

Implemente a função `mediaPonderada(notas[], pesos[], quantidade)` que recebe:

- Um array de notas
- Um array de pesos correspondentes
- A quantidade de notas

A função deve retornar a média ponderada calculada.

Exemplo de uso

Digite a quantidade de notas: 3

Nota 1: 8.0

Peso 1: 2

Nota 2: 7.5

Peso 2: 3

Nota 3: 9.0

Peso 3: 5

Media ponderada: 8.35

Cálculo: $(8.0 \times 2 + 7.5 \times 3 + 9.0 \times 5) / (2+3+5) = 83.5 / 10 = 8.35$

Dica importante: Em C++, arrays são passados por referência implicitamente (como ponteiros). Isso significa que se você modificar os valores dentro do array na função, as alterações afetarão o array original!

Crie uma função extra `ordenarNotas(notas[], quantidade)` que ordene as notas em ordem crescente usando qualquer algoritmo de ordenação (bubble sort, por exemplo). Observe como o array original é modificado após a chamada da função, demonstrando a passagem por referência.

(49) Crie um validador de CPF simplificado.

Implemente duas funções:

- `validarFormato(cpf)` - verifica se o CPF tem 11 dígitos numéricos
- `calcularDigitoVerificador(cpfParcial)` - calcula os dois dígitos verificadores

O algoritmo para calcular dígitos verificadores:

1. Primeiro dígito: multiplique os 9 primeiros dígitos por 10,9,8,...,2 e some. Divida por 11 e pegue o resto. Se o resto for < 2, o dígito é 0, senão é 11-resto.
2. Segundo dígito: repita com os 10 dígitos (incluindo o primeiro verificador)

Exemplo de entrada

Digite o CPF (somente numeros): 52998224725

Exemplo de saída

CPF valido!

Exemplo de entrada 2

Digite o CPF (somente numeros): 12345678901

Exemplo de saída 2

CPF invalido!

- 50) Crie um sistema de pedidos para uma pizzaria.

Implemente as funções:

- `calcularPreco(tamanho, quantidadeSabores)` - calcula o valor baseado em:
 - ▶ Tamanho P (pequena): R\$ 30,00 + R\$ 5,00 por sabor
 - ▶ Tamanho M (média): R\$ 45,00 + R\$ 7,00 por sabor
 - ▶ Tamanho G (grande): R\$ 60,00 + R\$ 10,00 por sabor
- `aplicarDesconto(valor, cupom)` - aplica 10% de desconto se cupom for “PIZZA10”
- `calcularTroco(valorPago, valorTotal)` - calcula o troco

Exemplo de entrada

```
==== Pizzaria do Zé ====
Tamanho (P/M/G): G
Quantidade de sabores: 3
Cupom de desconto: PIZZA10
Valor pago: 100.00
```

Exemplo de saída

Resumo do pedido:

- Pizza G com 3 sabores
- Subtotal: R\\$ 90.00
- Desconto: R\\$ 9.00
- Total: R\\$ 81.00
- Troco: R\\$ 19.00

(51) Crie um simulador de dados para RPG.

Implemente as funções:

- `rolarDado(lados)` - simula o lançamento de um dado de N lados (retorna valor aleatório entre 1 e N)
- `rolarVantagem()` - rola 2d20 e retorna o maior valor (mecânica de vantagem do D&D)
- `calcularDano(ataque, defesa)` - calcula dano: se ataque > defesa, dano = ataque - defesa + 1d6, senão dano = 0

O programa deve permitir o usuário escolher que tipo de rolagem fazer.

Exemplo de uso

```
==== Simulador de Dados RPG ===
```

1. Rolar dado simples
2. Rolar com vantagem
3. Calcular dano

Escolha: 3

```
Digite o valor do ataque: 18
```

```
Digite o valor da defesa: 15
```

```
Dano causado: 7 (18-15 + 1d6: 4)
```

52) Crie um analisador de desempenho para corredores.

Implemente as funções:

- `calcularPace(tempoMinutos, distanciaKm)` - calcula o pace (minutos por km)
- `classificarCorrida(pace)` - classifica o desempenho:
 - Pace < 5: “Excelente”
 - 5 ≤ Pace < 6: “Bom”
 - 6 ≤ Pace < 7: “Regular”
 - Pace ≥ 7: “Precisa melhorar”
- `compararCorridas(pace1, pace2)` - retorna qual pace foi melhor e a diferença
- `projetarTempo(distancia, pace)` - estima o tempo para correr uma distância dada

Exemplo de uso

```
==== Analisador de Corrida ====
Tempo da corrida (minutos): 50
Distancia percorrida (km): 10

Pace: 5.00 min/km
Classificacao: Bom
```

```
Comparar com corrida anterior:
Pace anterior (min/km): 5.5
Melhoria: 0.5 min/km mais rapido!
```

```
Projecao para 21km (meia maratona):
Tempo estimado: 105 minutos (1h45min)
```