

Lista de Exercícios 1

1. Um empregado recebe R\$ 16,78 por cada hora trabalhada em uma semana. Se esse empregado fizer hora extra, deve receber o valor base acrescido de 50%. Do pagamento bruto do empregado, 6% são retidos pela Previdência Social, 14%, pelo Receita Federal (imposto de renda), 5% pelo estado, e R\$ 10 por semana para o Sindicato. Se o empregado tiver três ou mais dependentes, um adicional de R\$ 35 é retido para cobrir o custo extra do seguro de saúde. Com base nessas informações, escreva um programa que receba como entrada o número de horas trabalhadas em uma semana e o número de dependentes do empregado. O programa deve apresentar como saída o pagamento bruto e o pagamento líquido do empregado, bem como os valores retidos por impostos.
2. Uma loja comercializa aparelhos com o seguinte plano de crediário: zero de entrada, juros de 1,5% ao mês, e prestações mensais de R\$ 50. A prestação mensal de R\$ 50 é utilizada para pagar os juros, e o restante é utilizado para amortizar a dívida remanescente. Por exemplo, se o aparelho custa R\$ 1.000, no primeiro mês você paga 1,5% de R\$ 1.000 em juros. Isso dá R\$ 15. Os R\$ 35 restantes são deduzidos do principal, o que o deixa com um débito de R\$ 965,00. No mês seguinte você paga um juro de 1,5% sobre R\$ 965,00, que dá R\$ 14,48. Assim, você pode deduzir R\$ 35,52 (R\$ 50 – R\$ 14,48) do que deve. Com base nessas informações, escreva um programa que receba como entrada o preço de um aparelho e lhe diga quantos meses você levará para pagar o que deve, bem como a soma total paga em juros. Utilize um loop para calcular a soma paga em juros e o tamanho do débito a cada mês. O último pagamento pode ser inferior a R\$ 50 se o débito for menor, mas não se esqueça dos juros. Se você deve R\$ 50, então sua prestação mensal de R\$ 50 não saldará seu débito, embora vá chegar perto disso.
3. A fórmula para calcular o Índice de Massa Corporal (IMC) é a seguinte: $IMC = \text{peso (kg)} / \text{altura}^2 \text{ (m)}$. Implemente uma calculadora de IMC que leia o peso do usuário em quilogramas e a altura em metros e, em seguida, calcule e exiba o índice de massa corporal do usuário. Além disso, a aplicação deve exibir a situação que se encontra o usuário, de acordo com a tabela de IMC disponível em: https://pt.wikipedia.org/wiki/Índice_de_massa_corporal
4. Implemente um programa que empregue um loop for e os recursos de manipulação de strings para produzir a tabela a seguir com os números alinhados à direita em cada coluna.

| Número: | Quadrado: | Cubo: |
|---------|-----------|-------|
| 0 | 0 | 0 |
| 1 | 1 | 1 |
| 2 | 4 | 8 |
| 3 | 9 | 27 |
| 4 | 16 | 64 |
| 5 | 25 | 125 |

5. Escreva um programa que receba um inteiro contendo 0s e 1s representando um número binário e exiba seu equivalente decimal.
6. Escreva um programa que calcule a média e a moda de uma sequência de valores inteiros fornecidos pelo usuário.
7. Escreva um programa para o jogo "adivinha o número". Escolha o número a ser adivinhado selecionando um número inteiro aleatório no intervalo de 1 a 1000. Não revele esse número ao usuário. Exiba o prompt "Adivinha o número que está entre 1 e 1000 com o menor número de tentativas:". O jogador insere um primeiro palpite. Se o palpite estiver incorreto, exiba "Muito alto. Tente novamente." ou "Muito baixo. Tente novamente." conforme apropriado para ajudar o jogador a "acertar" a resposta correta e, em seguida, solicite ao usuário o próximo palpite. Quando o usuário inserir a resposta correta, exiba "Parabéns. Você acertou o número!" e permita que o usuário escolha se deseja jogar novamente.
8. Neste problema, recriaremos a clássica corrida da lebre e da tartaruga. Use geração de números aleatórios para desenvolver uma simulação deste evento memorável. Nossos competidores começam a corrida no quadrado 1 de 70 quadrados. Cada quadrado representa uma posição ao longo do percurso. A linha de chegada é no quadrado 70. O primeiro competidor a alcançar ou passar o quadrado 70 é recompensado com um balde de cenouras e alface frescas. O percurso serpenteia a encosta de uma montanha escorregadia, então ocasionalmente os competidores perdem terreno. A cada segundo, sua aplicação deve ajustar a posição dos animais. Use variáveis para acompanhar as posições dos animais (ou seja, os números das posições vão de 1 a 70). Comece cada animal na posição 1 (o "ponto de partida"). Se um animal deslizar para antes do quadrado 1, mova-o de volta para o quadrado 1. Crie duas funções que gerem os movimentos da tartaruga e da lebre, respectivamente, produzindo um inteiro aleatório i no intervalo $1 \leq i \leq 10$. Na função para a tartaruga, execute um "andar lento" quando $1 \leq i \leq 5$, um "escorregão" quando $6 \leq i \leq 7$ ou um "andar lento" quando $8 \leq i \leq 10$. Elabore uma técnica semelhante na função para a lebre.
9. Escreva uma função que receba uma string como parâmetro e retorne uma string idêntica, mas com todos os caracteres convertidos para maiúsculo. Não use métodos prontos da classe string. Dicas: O código ASCII da letra 'a' (minúsculo) é 97 e o código da letra 'A' (maiúsculo) é 65, o código da letra 'z' é 122 e o código da letra 'Z' é 90. Cuidado para não acabar convertendo aquilo que não é nem texto e nem está em minúsculas!.
10. Faça uma função que receba duas strings e retorne uma string formada pela concatenação das strings recebidas. Exemplo: Se a primeira string digitada for "Bom dia, " e a segunda "moçada !", então o resultado é "Bom dia, moçada !".
11. Escreva um programa para ajudar um aluno do ensino fundamental a aprender multiplicação. Crie uma função que gere e retorne aleatoriamente dois números inteiros positivos de um dígito. Use o resultado dessa função em seu script para solicitar ao usuário uma pergunta, como "Quanto é 6 vezes 7?". Para uma resposta correta, exiba a mensagem "Muito bom!" e faça outra pergunta de multiplicação. Para uma resposta incorreta, exiba a mensagem "Não. Tente novamente." e deixe o aluno tentar a mesma pergunta repetidamente até que finalmente acerte.

12. Faça um programa que leia uma string e a imprima de trás para frente.
13. Escreva um programa que leia uma linha de texto e separe as “palavras” usando caracteres de espaço como delimitadores. Mostre as palavras obtidas.
14. Construa um programa que leia duas strings do teclado. Imprima uma mensagem informando se a segunda string lida está contida dentro da primeira.
15. Escreva uma função que calcule a distância entre dois pontos cujas coordenadas devem ser informadas como parâmetro (x1, y1) e (x2, y2) .
16. Escreva uma função que receba 3 valores reais X, Y e Z e que verifique se esses valores podem ser os comprimentos dos lados de um triângulo. Para que X, Y e Z formem um triângulo é necessário que a seguinte propriedade seja satisfeita: o comprimento de cada lado de um triângulo precisa ser menor do que a soma dos comprimentos dos outros dois lados. A função deve retornar 0 se não for triângulo; 1 se o triângulo for equilátero (3 lados são iguais); 2 se o triângulo for isósceles (2 lados são iguais); 3 se o triângulo for escaleno (3 lados são diferentes). Escreva um programa para testar a função.
17. Escreva uma função que receba por parâmetro dois valores X e Z e calcule x^z . (sem utilizar funções ou operadores de potência prontos).
18. Escreva uma função que receba como parâmetro um valor inteiro e positivo N e retorne o valor $S = 1 + 1/1! + 1/2! + 1/3! + \dots + 1/N!$
19. Dado que podemos calcular e^x por: $e^x = 1 + x + x^2/2! + x^3/3! + \dots$ Faça uma função que leia o valor de x e calcule o valor de e^x . O valor deve ser calculado enquanto o termo calculado for maior que $10E-6$.
20. Implemente um função genérica (template) chamada `maximoDeTres()` que receba três argumentos de mesmo tipo e retorne o maior deles. A função deve funcionar para **tipos primitivos** como `int`, `double`, `char`, etc. O programa principal (`main`) deve demonstrar a função com três inteiros, três números de ponto flutuante e três caracteres.
21. Implemente uma função chamada `gerarID(string grupo)` que retorne um **ID numérico único** a cada chamada, com a base no grupo informado como parâmetro. Existem dois grupos: “alunos e professores. Para cada grupo o primeiro ID deve ser 1, o segundo 2, e assim por diante. Além disso, o número do ID deve ser precedido pelo número do grupo.
22. Defina um função que receba um array c++ de elementos float como parâmetro e calcule a média dos valores. Use o método `at()` Teste a função.
23. Defina um função que receba um array c++ de elementos int como parâmetro e encontre o maior valor. Use um laço for simplificado. Teste a função.
24. Defina um função que receba duas matrizes de elementos float como parâmetros e retorne a multiplicação dessas matrizes. Teste a função.