

1. Desenvolva um programa que calcule a soma de todos os números pares no intervalo de 1 a 50 e exiba o resultado.
2. Com base na questão anterior, desenvolva um programa que solicite ao usuário um valor positivo x , some todos os números inteiros de 1 até x , e ao final, identifique se a soma é par ou ímpar, exibindo o resultado por exemplo: "A soma x é par, ou A soma x é ímpar".
3. Desenvolva um programa que solicite ao usuário que insira um número inteiro positivo x . O programa deve então contar e exibir a quantidade de dígitos presentes no número x e indicar se essa quantidade é par ou ímpar.
4. Desenvolva um programa que solicite ao usuário um número inteiro e em seguida, verifique se é divisível por 3. Se for divisível por 3, exiba uma mensagem indicando que é divisível por 3; caso contrário, exiba uma mensagem indicando que não é divisível por 3. **(Regra: Divisibilidade por 3 Um número é divisível por 3 quando a soma dos seus algarismo é um número divisível por 3.)**
5. Escreva um programa em C que imprima a tabuada de multiplicação de um número fornecido pelo usuário. O programa deve pedir ao usuário para inserir um número e, em seguida, imprimir a tabuada desse número de 1 a 10.
6. Escreva um programa em C que calcule o fatorial de um número fornecido pelo usuário usando um loop. O programa deve solicitar ao usuário para inserir um número inteiro não negativo e imprimir o fatorial desse número.
7. Faça um programa em C que verifique se um número fornecido pelo usuário é primo ou não. O programa deve pedir ao usuário para inserir um número inteiro positivo e imprimir uma mensagem indicando se o número é primo ou não.
8. Considere uma situação em que um programa em linguagem C está fazendo uma série de chamadas para um servidor. Suponha que o programa precise fazer um total de n chamadas, e que a probabilidade de uma chamada estar

ocupada seja de $p=40\%$. Agora, queremos calcular a

$$P(X = k) = \binom{n}{k} \times p^k \times (1 - p)^{n-k}$$

probabilidade de exatamente 3 chamadas estarem ocupadas.

- X : número de chamadas ocupadas (sucesso).
 - n : número total de chamadas.
 - p : probabilidade de uma chamada estar ocupada (sucesso).
 - $q=1-p$: probabilidade de uma chamada não estar ocupada (fracasso).
9. Desenvolva um programa que solicite ao usuário um número inteiro positivo e, em seguida, verifique se ele é um número de Armstrong. Um número de Armstrong é um número inteiro tal que a soma dos seus dígitos elevados à potência do número total de dígitos resulta no próprio número. Por exemplo, 153 é um número de Armstrong, pois $1^3 + 5^3 + 3^3 = 153$. O programa deve informar se o número dado pelo usuário é um número de Armstrong ou não.
 10. Durante um evento na universidade, houve a presença de 25 pessoas. Por um erro de impressão, foram distribuídos cupons de sorteio contendo apenas números ímpares. Para corrigir essa situação, é necessário desenvolver um programa em linguagem C que sorteie 5 brindes de forma aleatória. No entanto, caso um número par seja sorteado, ele deve ser convertido para o próximo número ímpar. Além disso, é fundamental garantir que não haja repetições nos números sorteados.
 11. Você foi solicitado a desenvolver um programa em C para simular o lançamento de um dado de 6 lados 100 vezes. Após os 100 lançamentos, o programa deverá exibir a porcentagem de vezes que cada lado do dado foi sorteado. Para calcular a porcentagem de vezes que um lado do dado foi sorteado, você pode dividir o número de ocorrências desse lado pelo total de lançamentos (100) e multiplicar por 100 para obter a porcentagem.
 12. Desenvolva um programa em C que solicite ao usuário um número inteiro positivo e, em seguida, desenhe um triângulo equilátero com base no número fornecido. Utilize asteriscos (*) para desenhar o triângulo. Por exemplo, se o usuário digitar o número 4, o programa deve exibir o seguinte padrão:

*
