

Lista de Exercícios sobre Estruturas de Repetição

1. Escreva um programa em C++ que solicite ao usuário uma sequência de números inteiros positivos terminada por zero e exiba o maior e o menor número da sequência. Detalhes:
 - Solicite ao usuário para digitar os números um por um, até que ele digite zero para terminar a sequência.
 - Durante a entrada dos números, armazene o maior e o menor número em variáveis separadas.
 - No final, exiba o maior e o menor número.
2. Crie um programa em C++ que solicite ao usuário um número inteiro positivo e verifique se ele é um número de Armstrong. Detalhes:
 - Peça ao usuário para digitar um número inteiro positivo.
 - Extraia os dígitos desse número e calcule a soma dos cubos dos dígitos.
 - Compare o resultado com o número original e verifique se eles são iguais.
 - Exiba uma mensagem indicando se o número é ou não um número de Armstrong.
3. Faça um programa em C++ que solicite ao usuário um número inteiro positivo e verifique se ele é um número primo. Detalhes:
 - Solicite ao usuário para digitar um número inteiro positivo.
 - Crie um loop que percorra todos os números de 2 até a raiz quadrada do número informado.
 - Verifique se o número informado é divisível por algum dos números dentro desse intervalo.
 - Se encontrar um divisor, exiba uma mensagem informando que o número não é primo.
 - Caso contrário, exiba uma mensagem informando que o número é primo.
4. Escreva um programa em C++ que solicite ao usuário um número inteiro positivo e exiba todos os seus divisores. Detalhes:
 - Solicite ao usuário para digitar um número inteiro positivo.
 - Crie um loop que percorra todos os números de 1 até o número informado.
 - Verifique se o número informado é divisível por cada número dentro desse intervalo.
 - Se encontrar um divisor, exiba-o na tela.
5. Crie um programa em C++ que solicite ao usuário um número inteiro positivo e verifique se ele é um número perfeito. Detalhes:
 - Solicite ao usuário para digitar um número inteiro positivo.
 - Crie um loop que percorra todos os números de 1 até o número informado (excluindo ele mesmo).
 - Verifique se o número informado é divisível por cada número dentro desse intervalo.
 - Se encontrar um divisor, some-o a uma variável de soma.
 - No final, verifique se a soma dos divisores é igual ao número informado e exiba uma mensagem indicando se é um número perfeito ou não.
6. Faça um programa em C++ que solicite ao usuário um número inteiro positivo e exiba a sequência de Collatz para esse número. Detalhes:
 - Solicite ao usuário para digitar um número inteiro positivo.

- Crie um loop que itere até que o número informado seja igual a 1.
 - A cada iteração, verifique se o número é par ou ímpar.
7. Se for par, divida-o por 2. Se for ímpar, multiplique-o por 3 e escreva um programa em C++ que solicite ao usuário uma sequência de números inteiros positivos terminada por zero e exiba o maior e o menor número da sequência. Detalhes:
- Solicite ao usuário para digitar os números um por um, até que ele digite zero para terminar a sequência.
 - Durante a entrada dos números, armazene o maior e o menor número em variáveis separadas.
 - No final, exiba o maior e o menor número.
8. Crie um programa em C++ que solicite ao usuário um número inteiro positivo e verifique se ele é um número de Armstrong. Detalhes:
- Peça ao usuário para digitar um número inteiro positivo.
 - Extraia os dígitos desse número e calcule a soma dos cubos dos dígitos.
 - Compare o resultado com o número original e verifique se eles são iguais.
 - Exiba uma mensagem indicando se o número é ou não um número de Armstrong.
9. Faça um programa em C++ que solicite ao usuário um número inteiro positivo e verifique se ele é um número primo. Detalhes:
- Solicite ao usuário para digitar um número inteiro positivo.
 - Crie um loop que percorra todos os números de 2 até a raiz quadrada do número informado.
 - Verifique se o número informado é divisível por algum dos números dentro desse intervalo.
 - Se encontrar um divisor, exiba uma mensagem informando que o número não é primo.
 - Caso contrário, exiba uma mensagem informando que o número é primo.
10. Escreva um programa em C++ que solicite ao usuário um número inteiro positivo e exiba todos os seus divisores. Detalhes:
- Solicite ao usuário para digitar um número inteiro positivo.
 - Crie um loop que percorra todos os números de 1 até o número informado.
 - Verifique se o número informado é divisível por cada número dentro desse intervalo.
 - Se encontrar um divisor, exiba-o na tela.
11. Crie um programa em C++ que solicite ao usuário um número inteiro positivo e verifique se ele é um número perfeito. Detalhes:
- Solicite ao usuário para digitar um número inteiro positivo.
 - Crie um loop que percorra todos os números de 1 até o número informado (excluindo ele mesmo).
 - Verifique se o número informado é divisível por cada número dentro desse intervalo.
 - Se encontrar um divisor, some-o a uma variável de soma.
 - No final, verifique se a soma dos divisores é igual ao número informado e exiba uma mensagem indicando se é um número perfeito ou não.
12. Faça um programa em C++ que solicite ao usuário um número inteiro positivo e exiba a sequência de Collatz para esse número. Detalhes:
- Solicite ao usuário para digitar um número inteiro positivo.

- Crie um loop que itere até que o número informado seja igual a 1.
- A cada iteração, verifique se o número é par ou ímpar.
- Se for par, divida-o por 2. Se for ímpar, multiplique-o por 3 e