

### **Hipotenusa de um Triângulo Retângulo:**

Crie um programa que recebe os dois lados menores de um triângulo retângulo e uma função retorna o valor da hipotenusa.

### **Existência de um Triângulo:**

Crie um programa que recebe os três lados de um triângulo e passa esses valores para uma função que verifica se esse triângulo existe ou não. Para que um triângulo exista, cada lado deve ser maior que o módulo da subtração dos outros dois lados e menor que a soma dos outros dois lados.

### **Quadrado de Hashtags:**

Faça um programa que peça um número inteiro positivo 'n' ao usuário e imprima um quadrado de lado 'n' preenchido com hashtags. Por exemplo, para n=4, o resultado seria:

####

####

####

####

### **Maior e Menor Número:**

Programa um software que recebe três números e cria duas funções: uma que retorna o

maior número e outra que retorna o menor número.

### **Número Perfeito:**

Um número é dito ser perfeito quando ele é igual à soma de seus divisores. Por exemplo, o seis é perfeito, pois:  $6=1+2+3$

.

Crie um programa que pede um número ao usuário e diga se ele é perfeito ou não.

### **Número Invertido:**

Crie um software que recebe um número do usuário, passa esse valor para uma função e ela retorna o número escrito ao inverso. Por exemplo, se o usuário der o valor 1234, a função deve retornar 4321. Dica: primeiro, crie uma função que conta quantos dígitos tem um número.

### **Lançamento de Moedas:**

Faça um programa para lançar uma moeda. Quando chamamos uma função, ela deve retornar “cara” ou “coroa”. Em outra função, faça ‘n’ lançamentos de moedas (onde ‘n’ é o valor que o usuário quiser) e mostre a porcentagem de vezes que deu cara e coroa. O que tende a acontecer se você jogar a moeda 10, 100, 1000 ou um milhão de vezes?

**Dado Aleatório:**

Crie um dado em C++. A função deve sortear um número aleatório de 1 até 6. Agora, faça com que o dado seja lançado 100 vezes, mil vezes e 1 milhão de vezes. Armazene o valor que ele forneceu em cada lançamento e mostre quantas vezes cada número foi sorteado. Compare os resultados com a estatística.

**Jogo de Par ou Ímpar:**

Escolha 0 para par ou 1 para ímpar. Em seguida, forneça um número. Crie um programa que determine se a soma do número escolhido com um número aleatório é par ou ímpar.

**Fatorial:**

Crie um programa que recebe um número inteiro positivo 'n' e calcula o fatorial desse número. O fatorial de 'n' é o produto de todos os números inteiros positivos de 1 até 'n'. Por exemplo,  $5! = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 120$

.

**Soma dos Dígitos:**

Escreva uma função que recebe um número inteiro e retorna a soma de seus dígitos. Por exemplo, se o número for 123, a função deve retornar 6 ( $1 + 2 + 3$ ).

**Média Aritmética:**

Crie um programa que pede ao usuário para digitar uma sequência de números e, em seguida, calcula a média aritmética desses números.

#### **Verificação de Palíndromo:**

Faça um programa que verifica se uma palavra ou frase é um palíndromo. Um palíndromo é uma palavra ou frase que se lê da mesma forma de trás para frente. Por exemplo, “arara” é um palíndromo.

#### **Contagem de Vogais:**

Escreva uma função que recebe uma string e conta quantas vogais (a, e, i, o, u) ela contém.

#### **Conversão de Temperatura:**

Crie um programa que converte uma temperatura em graus Celsius para Fahrenheit. A fórmula de conversão é:  $F = 5C + 32$

.

#### **Números Primos:**

Faça um programa que verifica se um número é primo. Um número é primo se ele não possui divisores além de 1 e ele mesmo.

### **Soma dos Números Pares:**

Escreva uma função que recebe um número inteiro positivo 'n' e retorna a soma de todos os números pares de 1 até 'n'.

### **Sequência de Fibonacci:**

Crie um programa que gera os primeiros 'n' números da sequência de Fibonacci. A sequência começa com 0 e 1, e cada número subsequente é a soma dos dois anteriores.

### **Cálculo de Juros Compostos:**

Faça um programa que calcula o montante final de um investimento com juros compostos. O usuário deve fornecer o capital inicial, a taxa de juros anual e o período de investimento em anos.

### **Validação de CPF:**

Implemente um programa que verifica se um número de CPF é válido. O CPF é composto por 11 dígitos e possui um algoritmo de validação específico.

### **Conversão de Bases Numéricas:**

Crie um programa que converte um número decimal para binário, octal ou hexadecimal, conforme a escolha do usuário.

#### **Cálculo de Média Ponderada:**

Escreva um programa que pede ao usuário para digitar as notas e os pesos de várias disciplinas e, em seguida, calcula a média ponderada.

#### **Ordenação de Números:**

Implemente um algoritmo de ordenação (por exemplo, Bubble Sort, Selection Sort ou Quick Sort) para ordenar um vetor de números inteiros.

#### **Cálculo de Potência:**

Crie uma função que recebe a base e o expoente e calcula a potência. Por exemplo,  $2^3=8$

.

#### **Contagem de Caracteres:**

Escreva um programa que conta quantos caracteres (incluindo espaços) há em uma string.

**Cálculo de Média Harmônica:**

Calcule a média harmônica de um conjunto de números. A fórmula é:  $H = \frac{1}{\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_n}}$

.

**Verificação de Anagramas:**

Faça um programa que verifica se duas palavras são anagramas (ou seja, possuem as mesmas letras, mas em ordem diferente).

**Conversão de Horas para Minutos:**

Escreva uma função que recebe um valor em horas e retorna o equivalente em minutos.