

//1- Crie uma função que recebe como parâmetro um número inteiro e devolve o seu dobro.

```
#include <stdio.h>

int dobro(int num);

void main()
{
    int numero = 0;

    printf("Digite um número: ");
    scanf("%d", &numero);

    printf("O dobro de %d eh %d\n", numero, dobro(numero));
}

int dobro(int num) {
    return num * 2;
}
```

//2- Faça uma função que receba a data atual (dia, mês e ano em inteiro) e exiba-a na tela no formato textual por extenso. Ex.: Data: 18/11/2022, Imprimir: 18 de novembro de 2022.

```
#include <stdio.h>

void formatDate(int dia, int mes, int ano);

void main()
{
    int dia;
    int mes;
    int ano;

    scanf("%d/%d/%d", &dia, &mes, &ano);
    formatDate(dia, mes, ano);
}

void formatDate(int dia, int mes, int ano)
{
    char* months[] = {"janeiro", "fevereiro", "março", "abril", "maio", "junho", "julho", "agosto",
"setembro", "outubro", "novembro", "dezembro"};

    printf("%d de %s de %d\n", dia, mes[mes - 1], ano);
}
```

//3Faça uma função para verificar se um número é positivo ou negativo. Sendo que o valor de retorno será 1 se positivo, -1 se negativo e 0 se for igual a 0.

```
#include <stdio.h>
```

```
int checkNumber(int number);
```

```
void main()
```

```
{  
    int number = 0;  
  
    scanf("%d", &number);  
    printf("%d\n", checkNumber(number));  
}
```

```
int checkNumber(int number)
```

```
{  
    if (number > 0)  
        return 1;  
    else if (number < 0)  
        return -1;  
    else
```

```
        return 0;  
}
```

//4 - Faça uma função para verificar se um número é um quadrado perfeito. Um quadrado perfeito é um número inteiro não negativo que pode ser expresso como o quadrado de outro número inteiro. Ex.: 1, 4, 9...

```
#include <stdio.h>
```

```
int checkPerfectSquare(int number);
```

```
void main()
```

```
{  
    int number = 0;  
  
    scanf("%d", &number);  
  
    if (checkPerfectSquare(number)) printf("É um quadrado perfeito\n");  
    else printf("Não é um quadrado perfeito\n");  
}
```

```

int checkPerfectSquare(int number)
{
    int i = 1;

    while (i * i < number)
        i++;

    if (i * i == number)
        return 1;
    else

    return 0;
}

```

//5 - Faça uma função e um programa de teste para o cálculo do volume de uma esfera. Sendo que o raio é passado por parâmetro.

```

#include <stdio.h>

float sphereVolume(float radius);

void main()
{
    float radius;

    scanf("%f", &radius);
    printf("%.2f\n", sphereVolume(radius));
}

float sphereVolume(float radius)
{
    return (4.0 / 3.0) * 3.14159 * radius * radius * radius;
}

```

//6 - Faça uma função que receba 3 números inteiros como parâmetro, representando horas, minutos e segundos, e os converta em segundos.

```

#include <stdio.h>

int convertToSeconds(int hours, int minutes, int seconds);

void main()

```

```

{
    int hours, minutes, seconds;

    scanf("%d:%d:%d", &hours, &minutes, &seconds);
    printf("%d\n", convertToSeconds(hours, minutes, seconds));
}

int convertToSeconds(int hours, int minutes, int seconds)
{
    return hours * 3600 + minutes * 60 + seconds;
}

```

//7- Faça uma função que receba uma temperatura em graus Celsius e retorne-a convertida em graus Fahrenheit. A fórmula de conversão é: $F = C * (9,0/5,0) + 32,0$, sendo F a temperatura em Fahrenheit e C a temperatura em Celsius.

```

#include <stdio.h>

float convertToFahrenheit(float celsius);

void main()
{
    float celsius;

    scanf("%f", &celsius);
    printf("%.2f\n", convertToFahrenheit(celsius));
}

float convertToFahrenheit(float celsius)
{
    return celsius * (9.0 / 5.0) + 32.0;
}

```

//8 - Sejam a e b os catetos de um triângulo, onde a hipotenusa é obtida pela equação: $\text{hipotenusa} = \sqrt{a^2 + b^2}$. Faça uma função que receba os valores de a e b e calcule o valor da hipotenusa através da equação.

```

#include <stdio.h>
#include <math.h>

float calculateHypotenuse(float a, float b);

```

```

void main()
{
    float a, b;

    scanf("%f %f", &a, &b);
    printf("%.2f\n", calculateHypotenuse(a, b));
}

```

```

float calculateHypotenuse(float a, float b)
{
    return sqrt(a * a + b * b);
}

```

//9 - Faça uma função que receba a altura e o raio de um cilindro circular e retorne o volume do cilindro. O volume de um cilindro circular é calculado por meio da seguinte fórmula: $V = \pi \cdot raio^2 \cdot altura$, onde $\pi = 3.141592$.

```

#include <stdio.h>

float cylinderVolume(float radius, float height);

void main()
{
    float radius, height;

    scanf("%f %f", &radius, &height);
    printf("%.2f\n", cylinderVolume(radius, height));
}

float cylinderVolume(float radius, float height)
{
    return 3.141592 * radius * radius * height;
}

```

//10 - Faça uma função que receba dois números e retorne qual deles é o maior.

```

#include <stdio.h>

int max(int a, int b);

void main()
{

```

```

int a, b;

scanf("%d %d", &a, &b);
printf("%d\n", max(a, b));
}

int max(int a, int b)
{
    if (a > b)
        return a;
    else
        return b;
}

```

//11 - Elabore uma função que receba três notas de um aluno como parâmetros e uma letra. Se a letra for A, a função deverá calcular a média aritmética das notas do aluno; se for P, deverá calcular a média ponderada, com pesos 5, 3 e 2.

```

#include <stdio.h>

float calculateAverage(float grade1, float grade2, float grade3, char type);

void main()
{
    float grade1, grade2, grade3;
    char type;

    scanf("%f %f %f %c", &grade1, &grade2, &grade3, &type);
    printf("%.2f\n", calculateAverage(grade1, grade2, grade3, type));
}

float calculateAverage(float grade1, float grade2, float grade3, char type)
{
    if (type == 'A')
    {
        return (grade1 + grade2 + grade3) / 3.0;
    }
    else if (type == 'P')
    {
        return (grade1 * 5.0 + grade2 * 3.0 + grade3 * 2.0) / 10.0;
    }
    else
    {
        return 0.0;
    }
}

```

```
}
```

//12 - Escreva uma função que receba um número inteiro maior do que zero e retorne a soma de todos os seus algarismos. Por exemplo, ao número 251 corresponderá o valor 8 ($2 + 5 + 1$). Se o número lido não for maior do que zero, o programa terminará com a mensagem "Número inválido".

```
#include <stdio.h>
```

```
int sumDigits(int number);
```

```
void main()
```

```
{
```

```
    int number;
```

```
    scanf("%d", &number);
```

```
    if (number > 0)
```

```
        printf("%d\n", sumDigits(number));
```

```
    else
```

```
        printf("Número inválido\n");
```

```
}
```

```
int sumDigits(int number)
```

```
{
```

```
    int sum = 0;
```

```
    while (number > 0)
```

```
    {
```

```
        sum += number % 10;
```

```
        number /= 10;
```

```
    }
```

```
    return sum;
```

```
}
```

//13- Faça uma função que receba dois valores numéricos e um símbolo. Este símbolo representará a operação que se deseja efetuar com os números. Se o símbolo for '+' deverá ser realizada uma adição, se for '-' uma subtração, se for '*' uma multiplicação e se for '/' será efetuada uma divisão

```
#include <stdio.h>
```

```
float calculate(float a, float b, char symbol);
```

```

void main()
{
    float a, b;
    char symbol;

    scanf("%f %f %c", &a, &b, &symbol);
    printf("%.2f\n", calculate(a, b, symbol));
}

```

```

float calculate(float a, float b, char symbol)
{
    if (symbol == '+')
        return a + b;

    else if (symbol == '-')
        return a - b;

    else if (symbol == '*')
        return a * b;

    else if (symbol == '/')
        return a / b;

    else

        return 3;
}
}

```

//14 - Faça uma função que receba a distância em Km e a quantidade de litros de gasolina consumidos por um carro em um percurso, calcule o consumo em Km/l e escreva uma mensagem de acordo com a tabela.

```
#include <stdio.h>
```

```
void calculateConsumption(float distance, float liters);
```

```

void main()
{
    float distance, liters;

    scanf("%f %f", &distance, &liters);
    calculateConsumption(distance, liters);
}

```

```
void calculateConsumption(float distance, float liters)
```



```

{
    float consumption = distance / liters;

    if (consumption < 8)
    {
        printf("Venda o carro!\n");
    }
    else if (consumption >= 8 && consumption <= 14)
    {
        printf("Econômico!\n");
    }
    else
    {
        printf("Super econômico!\n");
    }
}

```

//15 - Crie um programa que receba três valores (obrigatoriamente maiores que zero), representando as medidas dos três lados de um triângulo. Elabore funções para:

a) Determinar se eles lados formam um triângulo, sabendo que:

- O comprimento de cada lado de um triângulo é menor do que a soma dos outros dois lados.

b) Determinar e mostrar o tipo de triângulo, caso as medidas formem um triângulo. Sendo que:

- Chama-se equilátero o triângulo que tem três lados iguais;
- Denominam-se isósceles o triângulo que tem o comprimento de dois lados iguais;
- Recebe o nome de escaleno o triângulo que tem os três lados diferentes.

```
#include <stdio.h>
```

```
int isTriangle(float a, float b, float c);
```

```
void triangleType(float a, float b, float c);
```

```
void main()
```

```

{
    float a, b, c;

    scanf("%f %f %f", &a, &b, &c);
    if (isTriangle(a, b, c)) triangleType(a, b, c);
    else printf("Nao e um triangulo\n");
}

```

```
int isTriangle(float a, float b, float c)
```

```

{
    if (a < b + c && b < a + c && c < a + b) return 1;
}

```

```

    else return 0;
}

void triangleType(float a, float b, float c)
{
    if (a == b && b == c) printf("Equilatero\n");
    else if (a == b || b == c || a == c) printf("Isosceles\n");
    else printf("Escaleno\n");
}

```

//16 - Faça uma função chamada `desenha_linha`. Ele deve desenhar uma linha na tela usando uma sequência de símbolos de igual (Ex.: =====). A função recebe por parâmetro quantos sinais de igual serão mostrados.

```

#include <stdio.h>

void drawLine(int number);

void main()
{
    int number;

    scanf("%d", &number);
    drawLine(number);
}

void drawLine(int number)
{
    int i;

    for (i = 0; i < number; i++)
        printf("=");
    printf("\n");
}

```

//17 - Faça uma função que receba dois números inteiros positivos por parâmetro e retorne a soma dos N números inteiros existentes entre eles.

```

#include <stdio.h>

int sumBetween(int a, int b);

void main()
{

```

```

int a, b;

scanf("%d %d", &a, &b);
printf("%d\n", sumBetween(a, b));
}

int sumBetween(int a, int b)
{
    int sum = 0;

    if (a > b) {
        int aux = a;
        a = b;
        b = aux;
    }

    for (int i = a + 1; i < b; i++)
        sum += i;

    return sum;
}

```

//18 - Faça uma função que receba por parâmetro dois valores inteiros x e z. Calcule e retorne o resultado de x^z para o programa principal. Atenção, não utilize nenhuma função pronta de exponenciação.

```

#include <stdio.h>

int power(int x, int z);

void main()
{
    int x, z;

    scanf("%d %d", &x, &z);
    printf("%d\n", power(x, z));
}

int power(int x, int z)
{
    int result = 1;

    for (int i = 0; i < z; i++)
        result *= x;

    return result;
}

```

```
}
```

//19 - Faça uma função que receba um número inteiro positivo n e calcule o seu fatorial, n!.

```
#include <stdio.h>
```

```
int factorial(int n);
```

```
void main()
```

```
{
```

```
    int n;
```

```
    scanf("%d", &n);
```

```
    printf("%d\n", factorial(n));
```

```
}
```

```
int factorial(int n)
```

```
{
```

```
    int i, result = 1;
```

```
    for (i = 1; i <= n; i++)
```

```
        result *= i;
```

```
    return result;
```

```
}
```

//21 - Crie uma função que receba como parâmetro um valor inteiro e gere como saída n linhas com pontos de exclamação, conforme o exemplo abaixo (para n = 5):

```
!
```

```
!!
```

```
!!!
```

```
!!!!
```

```
!!!!!
```

```
#include <stdio.h>
```

```
void printExclamation(int n);
```

```
void main()
```

```
{
```

```
    int n;
```

```
    scanf("%d", &n);
```

```

    printExclamation(n);
}

void printExclamation(int n)
{
    int i, j;

    for (i = 1; i <= n; i++)
    {
        for (j = 1; j <= i; j++)
            printf("!");
        printf("\n");
    }
}

```

//23 - Escreva uma função que gera um triângulo lateral de altura $2 \cdot n - 1$ e n largura. Por exemplo, a saída para $n = 4$ seria:

```

*
**
***
****
***
**
*
*

```

```

#include <stdio.h>

```

```

void printTriangle(int n);

```

```

void main()

```

```

{
    int n;

    scanf("%d", &n);
    printTriangle(n);
}

```

```

void printTriangle(int n)

```

```

{
    int i, j;

    for (i = 1; i <= 2 * n - 1; i++)
    {
        if (i <= n)

```

```

    {
        for (j = 1; j <= i; j++)
            printf("*");
    }
    else
    {
        for (j = 1; j <= 2 * n - i; j++)
            printf("*");
    }
    printf("\n");
}
}

```

//24 - Escreva uma função que gera um triângulo de altura e lados n e base $2 \cdot n - 1$. Por exemplo, a saída para n = 6 seria:

```

    *
   ***
  *****
 *****
*****
*****
*/

```

```

#include <stdio.h>

```

```

void printTriangle(int n);

```

```

void main()

```

```

{
    int n;

    scanf("%d", &n);
    printTriangle(n);
}

```

```

void printTriangle(int n)

```

```

{
    int i, j;

    for (i = 1; i <= n; i++)
    {
        for (j = 1; j <= n - i; j++)
            printf(" ");
        for (j = 1; j <= 2 * i - 1; j++)
            printf("*");
        printf("\n");
    }
}

```

}
}