

Programa 1: Faça uma função que recebe por parâmetro o raio de uma esfera e calcula o seu volume ($v = 4/3 * \pi * R^3$).

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int volume(double r)
```

```
{
```

```
    double v;
```

```
    double PI = 3.14159;
```

```
    v = (4 / 3.0) * PI * pow(r,3);
```

```
    cout << v << endl;
```

```
}
```

```
int main()
```

```
{
```

```
    double raio;
```

```
    cin >> raio;
```

```
    volume(raio);
```

```
    return 0;
```

```
}
```

Programa 2: Escreva uma função que recebe as 3 notas de um aluno por parâmetro e uma letra. Se a letra for A, a função calcula a média aritmética das notas do aluno, se for P, a sua média ponderada (pesos: 5, 3 e 2) e se for H, a sua média harmônica. A média calculada também deve retornar por parâmetro.

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int media(double a, double b, double c, char f)
```

```
{  
    if(f == 'A')  
    {  
        double ma;  
        ma = (a + b + c) / 3;  
        cout << ma << endl;  
    }  
    if(f == 'P')  
    {  
        double mp;  
        mp = ((a * 5) + (b * 3) + (c * 2)) / 10;  
        cout << mp << endl;  
    }  
    if(f == 'H')  
    {  
        double mh;  
        mh = (3 / ((1 / a) + (1 / b) + (1 / c)));  
        cout << mh << endl;  
    }  
}
```

```
int main()
```

```
{
```

```
double a, b, c;  
char f;  
cin >> f;  
cin >> a >> b >> c;  
media(a,b,c,f);  
return 0;  
}
```

Programa 3: Faça uma função que recebe por parâmetro um valor inteiro e positivo e retorna o valor lógico Verdadeiro caso seja primo e Falso caso contrário.

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int prime(int a)
```

```
{
```

```
    int aux = 0;
```

```
    for(int i = 1; i < a / 2 + 1; i++)
```

```
    {
```

```
        if (a % i == 0)
```

```
        {
```

```
            aux += i;
```

```
        }
```

```
    }
```

```
    if (aux == 1)
```

```
    {
```

```
        cout << "Verdadeiro" << endl;
```

```
    }
```

```
    else
```

```
    {
```

```
        cout << "Falso" << endl;
```

```
    }
```

```
}
```

```
int main()
```

```
{
```

```
    int a;
```

```
    cin >> a;
```

```
    prime(a);
```

```
return 0;
```

```
}
```

Programa 4: Faça uma função que recebe por parâmetro os valores necessários para o cálculo da fórmula de bhaskara e retorna, por referência, as raízes R1 e R2, caso seja possível calcular.

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
void baskhara(double x, double y, double z, double &R1, double &R2)
```

```
{  
    double delta = sqrt(y * y - 4 * x * z);  
    R1 = (-y + delta) / (2 * x);  
    R2 = (-y - delta) / (2 * x);  
}
```

```
int main()
```

```
{  
    double a, b, c, R1, R2;  
    cin >> a >> b >> c;  
    if((b * b - 4 * a * c < 0) || (a == 0))  
    {  
        cout << "Impossivel calcular" << endl;  
    }  
    else  
    {  
        baskhara(a, b, c, R1, R2);  
        cout << "R1 = " << R1 << endl;  
        cout << "R2 = " << R2 << endl;  
    }  
    return 0;  
}
```

Programa 5: Faça uma função que recebe por parâmetro o tempo de duração de uma fábrica expressa em segundos e retorna também por parâmetro esse tempo em horas, minutos e segundos.

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int duracao(int d)
```

```
{  
    int h, m, s;  
    h = d / 3600;  
    d = d % 3600;  
    m = d / 60;  
    s = d % 60;  
    cout << h << ":" << m << ":" << s << endl;  
}
```

```
int main()
```

```
{  
    int d;  
    cin >> d;  
    duracao(d);  
    return 0;  
}
```

Programa 6: Faça uma função que recebe um valor inteiro e verifica se o valor é positivo ou negativo. A função deve retornar um valor booleano.

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
bool type(int a)
```

```
{
```

```
    if(a >= 0)
```

```
    {
```

```
        return cout << "True" << endl;
```

```
    }
```

```
    return cout << "False" << endl;
```

```
}
```

```
int main()
```

```
{
```

```
    int a;
```

```
    cin >> a;
```

```
    type(a);
```

```
    return 0;
```

```
}
```


Programa 7: Faça uma função que recebe um valor inteiro e verifica se o valor é par ou ímpar. A função deve retornar um valor booleano.

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
bool type(int a)
```

```
{
```

```
    if(a >= 0)
```

```
    {
```

```
        return cout << "True" << endl;
```

```
    }
```

```
    return cout << "False" << endl;
```

```
}
```

```
int main()
```

```
{
```

```
    int a;
```

```
    cin >> a;
```

```
    type(a);
```

```
    return 0;
```

```
}
```

Programa 9: Escreva uma função que recebe, por parâmetro, dois valores X e Z e calcula e retorna X^Z . (Sem utilizar funções ou operadores de potência prontos)

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int potencia(int num, int exp)
```

```
{
```

```
    int total = 1;
```

```
    for(int i = 1; i <= exp; i++)
```

```
    {
```

```
        total = total * num;
```

```
    }
```

```
    return total;
```

```
}
```

```
int main()
```

```
{
```

```
    int num, exp, result;
```

```
    cin >> num >> exp;
```

```
    result = potencia(num, exp);
```

```
    cout << result << endl;
```

```
    return 0;
```

```
}
```

Programa 10: Faça uma função que recebe 3 valores e retorna o menor deles para ser mostrado no programa principal.

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int menor(int a, int b, int c)
```

```
{  
    if(a < b and a < c)  
    {  
        return a;  
    }  
    if(b < a and b < c)  
    {  
        return b;  
    }  
    if(c < a and c < b)  
    {  
        return c;  
    }  
}
```

```
int main()
```

```
{  
    int a, b, c;  
    cin >> a >> b >> c;  
    menor(a, b, c);  
    cout << "O menor numero eh: " << menor(a, b, c) << endl;  
    return 0;  
}
```

Programa 11: Faça uma função que calcula o fatorial de um número passado a ela e retorna o resultado para o programa principal. Mostre o fatorial calculado no programa principal.

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int fatorial(int n)
```

```
{
```

```
    if(n == 0)
```

```
    {
```

```
        return 1;
```

```
    }
```

```
    return n * fatorial(n - 1);
```

```
}
```

```
int main()
```

```
{
```

```
    int n;
```

```
    cin >> n;
```

```
    cout << fatorial(n) << endl;
```

```
    return 0;
```

```
}
```

Programa 12: Crie uma função denominada “divisores”. A seguir. Leia um valor no programa principal e chame a função para mostrar quantos divisores o valor lido possui. Nesse caso os divisores podem ser apresentados dentro da função chamada.

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int divisores(int num)
```

```
{
```

```
    int divisores = 1;
```

```
    for (int i = 1; i < num; i++)
```

```
    {
```

```
        if (num % i == 0)
```

```
        {
```

```
            divisores++;
```

```
        }
```

```
    }
```

```
    return divisores;
```

```
}
```

```
int main()
```

```
{
```

```
    int num;
```

```
    cin >> num;
```

```
    cout << divisores(num) << endl;
```

```
    return 0;
```

```
}
```

Programa 13: Faça uma função que recebe um caractere (minúsculo). A seguir, dentro da função, mostre todos os caracteres, do caractere lido até o caractere 'z'. Exemplo:

Digite um caractere: s

s t u v w x y z

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
int caractere(char a)
{
    for(int i = a; i <= 'z'; i++)
    {
        cout << (char)i << ' ';
    }
    cout << endl;
}
```

```
int main()
{
    char a;
    cin >> a;
    caractere(a);
    return 0;
}
```