

**UNIVERSIDADE FEDERAL DE OURO PRETO  
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS  
DEPARTAMENTO DE COMPUTAÇÃO  
CIC107 – PROGRAMAÇÃO DE COMPUTADORES I**

**9ª Aula Prática  
Introdução a Funções**

Uma função é um bloco de instruções que é executado quando ela é chamada de algum outro ponto do programa.

**Exemplo 1:** Função que calcula a área de um retângulo.

```
#include <iostream>
using namespace std;

float area(float b, float h){
    float a;
    a = b * h;
    return a;
}

main(){
    float resp;

    resp = area(10, 15);

    cout<<"A area do retangulo e: "<<resp;
    system("pause>null");
}

<<saída>>
A area do retangulo e: 150
```

Neste exemplo, ao chamar a função **area** a variável **b** recebe o valor 10 e a variável **h** recebe o valor 15. Em seguida a função realiza os calculos e retorna o valor 150 que é atribuído à variável **resp**.

```
float    area(float b, float h)
↓150
resp = area(    10,    15);
```

O formato geral para definição de uma função em C/C++ é o seguinte:

```
tipo_de_valor_de_retorno nome_da_função (lista_de_parâmetros)
{
    declarações e instruções
}
```

Onde:

- *tipo\_de\_valor\_de\_retorno* é o **tipo** do resultado (valor) retornado pela função (Exemplo: **int, float, char, bool**). O *tip\_de\_valor\_de\_retorno* **void** indica que a função não retorna um valor;

- *nome\_da\_função* é qualquer identificador válido e corresponde ao nome pelo qual será possível chamar a função;
- *lista\_de\_parâmetros* é uma lista separada por vírgulas na qual a função determina o **tipo** e o **nome** dos seus parâmetros, por exemplo, **int x**. Cada parâmetro funciona dentro da função como qualquer outra variável.

As funções permitem ao programador modularizar um programa, dividindo as tarefas a serem realizadas em diferentes blocos.

Quando a instrução

**return expressão;**

é executada, a "expressão" é avaliada e o seu resultado é retornado para o ponto do programa onde a função foi chamada. Assim, a execução de uma instrução "return" provoca a finalização da execução da função e o controle do programa volta para o local de onde a função foi chamada.

Uma função pode ser declarada com o *tipo\_de\_valor\_de\_retorno* "void", o que indica que nenhum valor será retornado. Nesse caso é comum designar a função como "procedimento" e a sua finalização é indicada pela instrução

**return;**

**Exemplo 2:** Programa que calcula a área e o perímetro de um retângulo.

```
#include <iostream>
using namespace std;

float area(float b, float h){
    float a;
    a = b * h;
    return a;
}

float perimetro(float b, float h){
    float p;
    p = 2*b + 2*h;
    return p;
}

main(){
    float base, altura, are, per;

    cout<<"Digite o valor da base: ";
    cin>>base;

    cout<<"Digite o valor da altura: ";
    cin>>altura;

    are = area(base, altura);
    per = perimetro(base, altura);

    cout<<"A area do retangulo e: "<<are<<endl;
    cout<<"O perimetro do retangulo e: "<<per<<endl;

    system("pause>null");
}
```

## Exercícios

1. Escreva uma função **potencia(base, expoente)** que retorna o valor de  $\text{base}^{\text{expoente}}$ . Por exemplo,  $\text{potencia}(3, 4) = 3 * 3 * 3 * 3$ . Assuma que **expoente** seja um valor inteiro maior que ou igual a zero e **base** seja um valor inteiro. (*Obs: neste exercício não use a biblioteca matemática*).

Incorpore esta função a um programa que lê os valores da **base** e **expoente** digitados pelo usuário (enquanto o valor do **expoente** não for negativo) e exibe o valor de  $\text{base}^{\text{expoente}}$ .

2. Desenvolva um programa que lê o peso (kg) e a altura (m) de uma pessoa (enquanto peso diferente de zero) e imprime o seu Índice de Massa Corporal (IMC). A fórmula para cálculo do

$$\text{IMC é: } \text{IMC} = \frac{\text{peso}}{(\text{altura})^2}$$

*Observação:* Crie uma função que recebe como parâmetros o peso e a altura e retorna como resultado o valor do IMC.

3. Normalmente o valor do Índice de Massa Corporal (IMC) é utilizado para indicar se uma pessoa está ou não acima do peso. Crie uma função, no programa anterior, que recebe como parâmetro o valor do IMC e imprime uma mensagem que infere sobre o peso da pessoa, conforme tabela a seguir.

IMC	Mensagem
$IMC < 18,5$	Abaixo do peso
$18,5 \leq IMC \leq 25$	Peso normal
$25 < IMC \leq 30$	Acima do peso
$IMC > 30$	Obeso

4. Escreva uma função que exibe um quadrado de asteriscos cujo lado é especificado por um parâmetro inteiro. Por exemplo, se o parâmetro for 4, a função exibe

```
****
****
****
****
```

Incorpore esta função a um programa que lê um valor inteiro digitado pelo usuário (enquanto este valor for diferente de zero) e exibe o quadrado de asteriscos correspondente.

5. Dizemos que um número inteiro é um número perfeito se a soma de seus fatores, incluindo 1 (mas não o próprio número), é igual ao número. Por exemplo, 6 é um número perfeito porque  $6 = 1 + 2 + 3$ . Escreva uma função que determina se seu parâmetro é um número perfeito. Use esta função em um programa que determina e imprime todos os números perfeitos entre 1 e 1000. Para cada número perfeito encontrado o programa também deverá imprimir os seus fatores.

*Sugestão:* Crie uma função que recebe como parâmetro um número perfeito e imprime os seus fatores.

6. A série de Fibonacci: 0, 1, 1, 2, 3, 5, 8, 13, 21, ... começa com os termos 0 e 1 e tem a propriedade de que cada termo subsequente é igual a soma dos seus dois precedentes imediatos. Escreva uma função **fibonacci( n )** que calcula o n-ésimo número de Fibonacci.