



Algoritmos e Programação II

Prof.^a Noeli A. Pimentel Vaz
Prof. Joilson dos Reis Brito

FUNÇÃO

Função

Quando estudamos expressões aritméticas em C, foram apresentadas algumas funções matemáticas:

pow(base,expoente)

sqrt(numeroInteiro)

Utilizamos também funções para trabalhar com caracteres e strings:

isdigit(caracter)

strlen(string)

Função

Vamos analisar as partes de uma função:

pow(base,expoente)

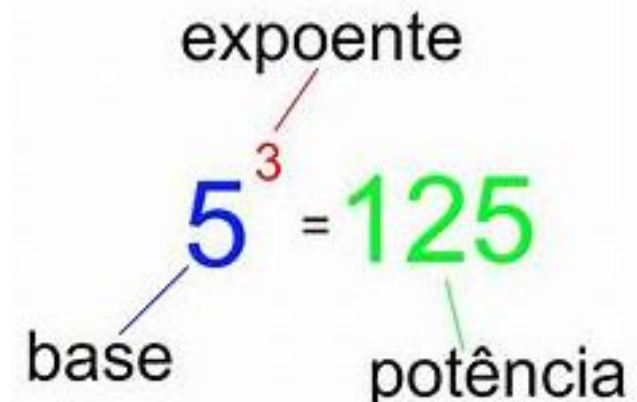


Nome: **pow**

Parâmetros passados para a função: **base e expoente**

Retorno da função: O valor que a função calcula e retorna para o programa. O retorno da função não está escrito quando chamamos uma função mas ele acontecerá.

O retorno da função **pow(5,3)** será 125.

A diagram showing the expression 5^3 = 125. The number 5 is blue and labeled 'base' with a blue line. The number 3 is red and labeled 'expoente' with a red line. The number 125 is green and labeled 'potência' with a green line. The equals sign is black.
$$\begin{array}{c} \text{expoente} \\ 5^3 = 125 \\ \text{base} \qquad \text{potência} \end{array}$$

Função

Além de utilizar funções prontas você pode criar suas próprias funções no C.



Funções são usadas para criar pequenos pedaços de códigos separados do programa principal.

Em C/C++, tudo, na verdade, é uma função.

int main() é uma função. A função principal do programa.

Função

Qual a vantagem de utilizar funções?



Funções são importantes porque **elas retornam valores**, ajudam a fragmentar o código em partes menores - mais fáceis de lidar - e ainda por cima podem ser utilizadas mais de uma vez no mesmo programa, poupando preciosos minutos de programação e inúmeras linhas de código.

Função

Uma função é um bloco de instruções que é executado quando é chamada de algum outro ponto do programa. Seu formato é o seguinte:

tipo nome (argumento1, argumento2, ...)

```
{  
    conteúdo  
}
```

onde:

- *tipo* é o tipo de dados que a função retorna(int, float,char).
- *nome* é o nome pelo qual será possível chamar a função.
- *argumentos* (podem ser especificados quantos você quiser). Cada argumento consiste em um tipo de dados seguido pelo seu identificador, como na declaração de uma variável (por exemplo, int x) e que funciona dentro da função como qualquer outra variável. Eles permitem a passagem de parâmetros para a função quando é chamada. Os parâmetros diferentes são separados por vírgulas.
- *conteúdo* é o corpo da função. Pode ser uma única instrução ou um bloco de instruções. No último caso, precisa ser delimitado por chaves {}.

Função

Para chamar uma função digitamos o nome da função e entre parênteses colocamos seus argumentos.

```
sqrt(9);
```

```
system("pause");
```

```
strlen(Nome);
```

Se a função não tem argumentos, simplesmente não colocamos nada entre os parênteses.

Exemplo 1

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

\\ As funções devem ser definidas antes do int main(), onde começa o programa.

```
int area(int LadoA, int LadoB)
{
    int AreaFigura;
    AreaFigura = LadoA*LadoB;
    return AreaFigura;
}
```

Exemplo 1

\\ Aqui começa o programa que irá chamar as funções definidas

```
int main()
{
    int Lado1, Lado2;
    setlocale(LC_ALL, "Portuguese");
    printf("Informe lado 1:");scanf("%i", &Lado1);
    printf("Informe lado 2:");scanf("%i", &Lado2);
    printf( "Área: %i\n", area(Lado1,Lado2));

    system("pause");
    return 0;
}
```

Exemplo 2

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

```
int fatorial(int NumeroInteiro)
{
    int l,Fatorial;
    Fatorial = 1;
    for (l=1;l<=NumeroInteiro;l++)
        Fatorial = Fatorial * l;
    return Fatorial;
}
```

Exemplo 2

\\ Aqui começa o programa principal que irá chamar as funções definidas

```
int main()
{
    setlocale(LC_ALL, "Portuguese");
    int NumeroParaFatorial;
    printf("\nInforme o número para fatorial:");
    scanf("%i", &NumeroParaFatorial);
    printf("O fatorial é: %i\n", fatorial(NumeroParaFatorial));
    system("pause");
    return 0;
}
```

Exemplo 3

```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
float salarioliquido(float Salario)
{
    float Inss,Liquido;
    Inss = 0.12;
    Liquido = Salario - (Salario * Inss);
    if(Salario >= 3000)
        Liquido = Liquido - (0.27*Liquido);
    else
        if((Salario >1500) && (Salario < 3000))
            Liquido = Liquido - (0.15*Liquido);
    return Liquido;
}
```

Exemplo

\\ Aqui começa o programa principal que irá chamar as funções definidas

```
int main()
{
    setlocale(LC_ALL, "Portuguese");
    float SalarioBruto;
    printf("\nInforme o salário bruto:");scanf("%f", &SalarioBruto);
    printf("O salário liquido é: %.2f\n", salarioliquido(SalarioBruto));
    system("pause");
    return 0;
}
```