

**FCT/Unesp – Presidente Prudente**  
**Departamento de Matemática e Computação**

**ATP I**

**Funções Matemáticas**  
**Biblioteca math.h**

**Prof. Danilo Medeiros Eler**  
**danilo.eler@unesp.br**

# Funções Matemáticas

- Geralmente, durante o desenvolvimento de programas precisamos de recursos matemáticos para resolver determinados problemas
- Para tanto, podemos implementar algoritmos ou utilizar funções pré-definidas
  - Essas funções podem estar codificadas em bibliotecas

# Funções Matemáticas

- Uma biblioteca muito conhecida é a `math.h`
  - Possui funções matemáticas prontas para serem usadas nos programas desenvolvidos na linguagem C
- Para usar a biblioteca, primeiramente, precisamos incluí-la no programa
  - `#include <math.h>`

# Funções Matemáticas

- Uma referência completa das funções da biblioteca math.h pode ser encontrada em
  - <https://www.cplusplus.com/reference/cmath/>
- As funções seguem o seguinte padrão  
    tipo\_retorno nome\_funcao(parâmetros)
  - tipo\_retorno: tipo do dado retornado pela função
  - nome\_funcao: nome dado para a função
  - parâmetros: variáveis para receber valores para a função manipular

# Funções Matemáticas

- Exemplos:

`double sqrt(double x)`

- tipo\_retorno: double
- nome\_funcao: sqrt (square root) --- calcula a raiz quadrada
- parâmetros: uma variável double
- Exemplo de uso:

`double num, raiz;`

`num = 9;`

`raiz = sqrt(num); //a função retornará o valor 3`

# Funções Matemáticas

- Exemplos:

`double ceil(double x)`

- tipo\_retorno: double
- nome\_funcao: ceil (teto) --- arredonda o valor de 'x' para o próximo inteiro que não é menor do que 'x'
  - Arredondamento para cima

- parâmetros: uma variável double

- Exemplo de uso:

```
double num, novo_valor;
```

```
num = 9.5;
```

```
novo_valor = ceil(num); //a função retornará o valor 10
```

# Funções Matemáticas

- Exemplos:

`double floor(double x)`

- tipo\_retorno: double
- nome\_funcao: floor (chão) --- arredonda o valor de 'x' para o próximo inteiro que não é maior do que 'x'
  - Arredondamento para baixo
- parâmetros: uma variável double
- Exemplo de uso:

```
double num, novo_valor;
```

```
num = 9.5;
```

```
novo_valor = floor(num); //a função retornará o valor 9
```

# Funções Matemáticas

- Exemplos:

`double round(double x)`

- tipo\_retorno: double
- nome\_funcao: round --- arredonda o valor de 'x' para o inteiro mais próximo de 'x'
- parâmetros: uma variável double
- Exemplo de uso:

```
double num, novo_valor;
```

```
num = 9.6;
```

```
novo_valor = round(num); //a função retornará o valor 10
```



# Funções Matemáticas

- Exemplos:

`double trunc(double x)`

- tipo\_retorno: double
- nome\_funcao: round --- retorna o valor inteiro mais próximo que não é maior em magnitude do que 'x'
  - Remove a parte fracionária
- parâmetros: uma variável double
- Exemplo de uso:

```
double num, novo_valor;
```

```
num = 9.6;
```

```
novo_valor = trunc(num); //a função retornará o valor 9
```

# Funções Matemáticas

- Exemplos:

`double sin(double x)`

- tipo\_retorno: double
- nome\_funcao: sin (seno) --- retorna o seno para o ângulo 'x'
  - O valor de 'x' deve estar em radianos
- parâmetros: uma variável double
- Exemplo de uso:

`double angulo, seno, PI = 3.14159265;`

`angulo = 30*PI/180; //converte de graus para radianos`

`seno = sin(angulo); //a função retornará o valor 0.5`

# Funções Matemáticas

- Exemplos:

`double cos(double x)`

- tipo\_retorno: double
- nome\_funcao: cos (cosseno) --- retorna o cosseno para o ângulo 'x'

- O valor de 'x' deve estar em radianos

- parâmetros: uma variável double

- Exemplo de uso:

`double angulo, cosseno, PI = 3.14159265;`

`angulo = 60*PI/180; //converte de graus para radianos`

`cosseno = cos(angulo); //a função retornará o valor 0.5`

# Exercícios

- Faça um programa para ler um ângulo em graus e apresentar o seno e cosseno desse ângulo
- Faça um programa para ler um número e apresentar o resultado das funções: *ceil*, *floor*, *round* e *trunc*
- Faça um programa para ler um número e apresentar a raiz quadrada do número
- Faça um programa para ler a base e o expoente. Em seguida, deverá apresentar o resultado da potenciação. Para tanto, use a função *pow*, conforme segue:
  - `double pow (double base, double exponent);`

# Exercícios

- Faça um programa para ler os valores de  $a$ ,  $b$  e  $c$  de uma equação do 2º grau. Em seguida, ele deverá calcular as raízes da equação.

Lembrando que:

$$x = \frac{-b \pm \sqrt{\Delta}}{2a}$$

Onde

$$\Delta = B^2 - 4ac$$

E  $ax^2 + bx + c = 0$  representa uma equação de 2º grau.

A variável  $a$  tem que ser diferente de zero. Caso seja igual, imprima a mensagem “Não é equação de segundo grau”.

- Se  $\Delta < 0$ , não existe raiz real. Imprima a mensagem Não existe raiz.
- Se  $\Delta = 0$ , existe uma raiz real. Imprima a raiz e a mensagem Raiz única.
- Se  $\Delta \geq 0$ , imprima as duas raízes reais.

<https://programacaodescomplicada.wordpress.com/>

# Bibliografia Básica

1. ASCENCIO, A. F. G.; CAMPOS, E. A. V. C. **Fundamentos da programação de computadores**: algoritmos, pascal e C/C++. Pearson Prentice Hall, 2003. 355p.
2. KERNINGHAN, B. W.; Ritchie, D. M. **C: a Linguagem de Programação padrão ANSI**. Rio de Janeiro: Editora Campus, 1990. 289p.
3. KERNINGHAN, B. W.; Pike, R. **A Prática de Programação**. Rio de Janeiro: Editora Campus, 2000, 280p.
4. LOPES, A.; GARCIA, G. **Introdução à Programação**: 500 exercícios resolvidos. Rio de Janeiro: Editora campus, 2002. 469p.
5. MANZANO, J. A. N. G.; OLIVEIRA, J. F. **Algoritmos**: lógica para desenvolvimento de programação de computadores. 10ª ed. Editora Érica Ltda. 2000, 236p.
6. MIZRAHI, V. V. **Treinamento em Linguagem C – Curso Completo – Módulo 1**. São Paulo: MAKRON Books do Brasil Editora Ltda, 1990. 241p.
7. MEDINA, M.; FERTIG, C. **Algoritmos e programação**: teoria e prática. São Paulo: Novatec Editora. 384p. 2005.
7. PUGA, S.; RISSETTI, G. **Lógica de programação e estruturas de dados**: com aplicações em JAVA. São Paulo: Pearson Education do Brasil, 254p. 2004.
8. SCHILDT H. **C Completo e Total**. 3ª ed. São Paulo: MAKRON Books do Brasil editora Ltda. 1997. 827p.
9. XAVIER, G. F. C. **Lógica de Programação**. São Paulo: Editora SENAC. 1998. 378p.

# Bibliografia Complementar

1. BROOKSHEAR, J. G. **Ciência da computação**: uma visão abrangente. 5ª ed., Bookman Editora, 2000. 499p.
2. CORMEN, T.H., Leiserson, C.E., Rivest R.L., Stein, C. **Algoritmos**: teoria e Prática. Rio de janeiro: Editora Campus, 2002. 916p.
3. PLAUGER, P. L. **A Biblioteca Standard C**. Rio de Janeiro: Editora Campus, 1994. 614p.
4. PRATA, S. **C primer plus**, 4ª ed. SAMS Publishing, 2002. 931p.
5. OLIVEIRA, U. **Programando em C**, vol. I – fundamentos. Editora Ciência Moderna, 2008, 743p.