

Flávio Augusto de Freitas
Introdução à Programação em Linguagem C/C++

<http://flavioaf.blogspot.com>

C/C++

Tutorial 5 (usando Dev-C++ versão 4.9.9.2)



2011

1 INTRODUÇÃO

Esta série de tutoriais foi escrita usando o **Microsoft Windows 7 Ultimate** e o **Bloodshed Dev-C++** versão 4.9.9.2, que pode ser baixada em <http://www.bloodshed.net>. Se alguém quiser adquirir mais conhecimentos e quiser aprofundar no assunto, visite <http://www.cplusplus.com/>.

Qualquer exemplo de programa nesta série de tutoriais, implica em um novo projeto, dentro de uma nova pasta. Se não souber como criar uma nova pasta, volte e refaça o tutorial 1.

2 OPERAÇÕES MATEMÁTICAS

O cabeçalho `<math.h>` contém protótipos de algumas funções na área de matemática. Na versão de 1990 do padrão ISO, somente a versão `double` das funções foram especificadas; na versão de 1999 foram adicionadas as versões `float` e `long double`.

As funções podem ser agrupadas nas seguintes categorias:

- Funções Trigonométricas
- Funções Hiperbólicas
- Funções Exponencial e Logaritmo
- Funções `pow` e `sqrt`
- Funções de Arredondamento para Números Inteiros, Valores Absolutos e Resto da Divisão

2.1 FUNÇÕES TRIGONOMÉTRICAS

2.1.1 AS FUNÇÕES ACOS E ASIN

A função `acos` retorna o arco-cosseno dos seus argumentos em radianos, e a função `asin` retorna o arco-seno dos seus argumentos em radianos. Todas as funções esperam por argumentos que estejam no intervalo $[-1, +1]$. O arco-cosseno retorna valores no intervalo $[0, \pi]$; o arco-seno retorna valores no intervalo $[-\pi/2, +\pi/2]$.

```
#include <math.h>
float asinf(float x); /* C99 */
float acosf(float x); /* C99 */
double asin(double x);
double acos(double x);
long double asinl(long double x); /* C99 */
long double acosl(long double x); /* C99 */
```

Exemplo:

```
#include <cstdlib>
#include <iostream>
#include <math.h>

using namespace std;

int main(int argc, char *argv[])
{
    double num = 0.5;

    printf("asin(%f) = %f rad\n", num, asin(num));

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

2.1.2 AS FUNÇÕES ATAN E ATAN2

As funções `atan` retornam o arco-tangente dos argumentos em radianos, e a função `atan2` retorna o arco-tangente de y/x em radianos. As funções `atan` retornam o valor no intervalo $[-\pi/2, +\pi/2]$ (a razão pelo que $\pm\pi/2$ está incluído no intervalo é porque os valores decimais pode representar o infinito, e $\text{atan}(\pm\infty) = \pm\pi/2$); as funções `atan2` retornam o valor no intervalo $[-\pi, +\pi]$. Para a função `atan2`, um "domain error" pode ocorrer se os dois argumentos forem zero.

```
#include <math.h>
float atanf(float x); /* C99 */
float atan2f(float y, float x); /* C99 */
double atan(double x);
double atan2(double y, double x);
long double atanl(long double x); /* C99 */
long double atan2l(long double y, long double x); /* C99 */
```

Exemplo:

```
#include <cstdlib>
#include <iostream>
#include <math.h>

using namespace std;

int main(int argc, char *argv[])
{
    double num = 0.17;

    printf("atan(%f) = %f rad\n", num, atan(num));

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

2.2.3 AS FUNÇÕES COS, SIN E TAN

As funções cos, sin, e tan retornam o cosseno, seno, e tangente do argumento, expresso em radianos.

```
#include <math.h>
float cosf(float x); /* C99 */
float sinf(float x); /* C99 */
float tanf(float x); /* C99 */
double cos(double x);
double sin(double x);
double tan(double x);
long double cosl(long double x); /* C99 */
long double sinl(long double x); /* C99 */
long double tanl(long double x); /* C99 */
```

Exemplo:

```
#include <cstdlib>
#include <iostream>
#include <math.h>

using namespace std;

int main(int argc, char *argv[])
{
    double num = 30.0; // 30 graus

    // acos(-1) = 3.141592... ou pi
    num = num / 180.0 * acos(-1);
    num = tan(num);
    printf("atan(%f rad) = %f graus\n",
        num, atan(num) * 180.0 / acos(-1));

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

2.2 FUNÇÕES HIPERBÓLICAS

As funções cosh, sinh e tanh computam o cosseno hiperbólico, o seno hiperbólico e a tangente hiperbólica respectivamente.

```
#include <math.h>
float coshf(float x); /* C99 */
float sinh(float x); /* C99 */
float tanhf(float x); /* C99 */
double cosh(double x);
double sinh(double x);
double tanh(double x);
long double coshl(long double x); /* C99 */
long double sinhl(long double x); /* C99 */
long double tanhl(long double x); /* C99 */
```

Exemplo:

```
#include <cstdlib>
#include <iostream>
#include <math.h>

using namespace std;

int main(int argc, char *argv[])
{
    double param, result;

    param = log(2.0);
    result = sinh(param);
    printf("sinh(%lf) = %lf\n", param, result);

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

2.3 FUNÇÕES EXPONENCIAL E LOGARITMO

2.3.1 A FUNÇÃO EXP

As funções exp computam a função exponencial de x (e^x). Um "range error" ocorre se o valor de x é muito grande.

```
#include <math.h>
float expf(float x); /* C99 */
double exp(double x);
long double expl(long double x); /* C99 */
```

Exemplo:

```
#include <cstdlib>
#include <iostream>
#include <math.h>

using namespace std;

int main(int argc, char *argv[])
{
    double param, result;

    param = 5.0;
    result = exp(param);
    printf(" %lf\n", param);
    printf("e      = %lf\n", result);

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

2.3.2 AS FUNÇÕES FREXP, LDEXP E MODF

As funções frexp dividem um número real numa fração normalizada e um número inteiro múltiplo de 2. As funções guardam o número inteiro no objeto apontado por ex.

As funções frexp retornam o valor x de forma que x tem o valor $[1/2, 1)$ ou zero, e $value$ é igual a x vezes 2 elevado a $*ex$. Se $value$ for zero, as duas partes do resultado serão zero.

As funções `ldexp` multiplicam um número real por um número inteiro múltiplo de 2 e retornam o resultado. Um "range error" pode ocorrer.

As funções `modf` divide o argumento `value` entre uma parte inteira e uma fração, cada uma tem o mesmo sinal do argumento. As funções guardam a parte inteira no objeto apontado por `*iptr` e retornam a fração.

```
#include <math.h>
float frexpf(float value, int *ex); /* C99 */
double frexp(double value, int *ex);
long double frexpl(long double value, int *ex); /* C99 */
float ldexpf(float x, int ex); /* C99 */
double ldexp(double x, int ex);
long double ldexpl(long double x, int ex); /* C99 */
float modff(float value, float *iptr); /* C99 */
double modf(double value, double *iptr);
long double modfl(long double value, long double *iptr); /* C99 */
```

2.3.3 AS FUNÇÕES LOG E LOG10

As funções `log` computam o logaritmo natural do argumento e retornam o resultado. Um "domain error" ocorre se o argumento for negativo. Um "range error" pode ocorrer se o argumento for zero.

As funções `log10` computam o logaritmo comum (base-10) do argumento e retornam o resultado. Um "domain error" ocorre se o argumento for negativo. Um "range error" ocorre se o argumento for zero.

```
#include <math.h>
float logf(float x); /* C99 */
double log(double x);
long double logl(long double x); /* C99 */
float log10f(float x); /* C99 */
double log10(double x);
long double log10l(long double x); /* C99 */
```

Exemplo:

```
#include <cstdlib>
#include <iostream>
#include <math.h>

using namespace std;

int main(int argc, char *argv[])
{
    double num, lg, ln;

    num = 2.0;

    ln = log(num);
    lg = log10(num);
    printf("ln(%lf) = %f\n", num, ln);
    printf("log(%lf) = %f\n", num, lg);
    printf("se ln(n) = %f, entao n = %f\n", ln, exp(ln));
    printf("se log(n) = %f, entao n = %f\n", lg, exp(lg*log(10)));

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

2.4 FUNÇÕES POW E SQRT

2.4.1 A FUNÇÃO POW

A função `pow` computa x elevado a y e retornam o resultado. Um "domain error" ocorre se x for negativo e y não for um número inteiro. Um "domain error" ocorre se o resultado não puder ser representado quando x é zero e y é menor ou igual a zero. Um "range error" pode ocorrer.

```
#include <math.h>
float powf(float x, float y); /* C99 */
double pow(double x, double y);
long double powl(long double x, long double y); /* C99 */
```

Exemplo:

```
#include <cstdlib>
#include <iostream>
#include <math.h>

using namespace std;

int main(int argc, char *argv[])
{
    double b, e;

    b = 2.0;
    e = 5.0;

    printf("%lf^%lf = %lf\n", b, e, pow(b, e));

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

2.4.2 A FUNÇÃO SQRT

As funções `sqrt` computam a raiz positiva de x e retornam o resultado. Um "domain error" ocorre se o argumento for negativo.

```
#include <math.h>
float sqrtf(float x); /* C99 */
double sqrt(double x);
long double sqrtl(long double x); /* C99 */
```

Exemplo:

```
#include <cstdlib>
#include <iostream>
#include <math.h>

using namespace std;

int main(int argc, char *argv[])
{
    double num;

    num = 2.0;

    printf("raiz de %lf = %lf\n", num, sqrt(num));

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

2.5 FUNÇÕES DE ARREDONDAMENTO PARA NÚMEROS INTEIROS, VALORES ABSOLUTOS E RESTO DA DIVISÃO

2.5.1 AS FUNÇÕES CEIL E FLOOR

As funções `ceil` computam o menor número inteiro que não seja menor que x e retornam o resultado; as funções `floor` computam o maior número inteiro que não seja maior que x e retornam o resultado.

```
#include <math.h>
float ceilf(float x); /* C99 */
double ceil(double x);
long double ceill(long double x); /* C99 */
float floorf(float x); /* C99 */
double floor(double x);
long double floorl(long double x); /* C99 */
```

Exemplo:

```
#include <cstdlib>
#include <iostream>
#include <math.h>
```

```
using namespace std;
```

```
int main(int argc, char *argv[])
{
    double num;

    num = 2.57456;

    printf("ceil de %lf = %lf\n", num, ceil(num));
    printf("floor de %lf = %lf\n", num, floor(num));

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

2.5.2 AS FUNÇÕES FABS

As funções `fabs` computam o valor absoluto do número real x e retornam o resultado.

```
#include <math.h>
float fabsf(float x); /* C99 */
double fabs(double x);
long double fabsl(long double x); /* C99 */
```

2.5.3 AS FUNÇÕES FMOD

As funções `fmod` computam o resto de x/y e retornam o valor $x - i * y$, pra algum número inteiro i onde, se y for um número diferente de zero, o resultado tem o mesmo sinal de x e magnitude menor que a magnitude de y . Se y for zero, dependendo da implementação da função, ocorrerá um "domain error" ou a função `fmod` retornará zero.

```
#include <math.h>
float fmodf(float x, float y); /* C99 */
double fmod(double x, double y);
long double fmodl(long double x, long double y); /* C99 */
```

Exemplo:

```
#include <cstdlib>
#include <iostream>
#include <math.h>
```

```
using namespace std;
```

```
int main(int argc, char *argv[])
{
    printf("fmod de 5.3/2 = %lf\n", fmod(5.3,2));
    printf("fmod de 18.5/4.2 = %lf\n", fmod(18.5,4.2));

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

Resolvendo uma equação do 2º grau

Achar as soluções para $y(0) = x^2 - 6x + 7 \Rightarrow$

$$x = \frac{6 \pm \sqrt{(-6)^2 - 4 \times 1 \times 7}}{2 \times 1} \Rightarrow x = 4,414 \text{ e } x = 1,586$$

```
#include <cstdlib>
#include <iostream>
#include <math.h>
```

```
using namespace std;
```

```
int main(int argc, char *argv[])
{
    double a, b, c, delta, x1, x2;

    a = 1.0; b = -6.0; c = 7.0;
    // experimente outros valores de a, b e c

    delta = pow(b, 2) - 4*a*c;
    x1 = (-b + sqrt(delta))/(2*a);
    x2 = (-b - sqrt(delta))/(2*a);
    printf("%fx^2 + (%f)x + (%f) = 0\n", a, b, c);
    printf("x1 = %f\n", x1);
    printf("x2 = %f\n", x2);

    system("PAUSE");
    return EXIT_SUCCESS;
}
```

3 EXERCÍCIOS PROPOSTOS

- Faça um programa para exibir uma tabela dos senos dos ângulos de 0 a 90 graus, de 15 em 15 graus.
- Faça um programa para exibir uma tabela dos cossenos dos ângulos de 0 a 90 graus, de 15 em 15 graus.
- Faça um programa para exibir uma tabela dos tangentes dos ângulos de 0 a 90 graus, de 15 em 15 graus.

- d) Faça um programa para exibir uma tabela dos senos, cossenos e tangentes dos ângulos de 0 a 90 graus, de 15 em 15 graus.
- e) Faça um programa para exibir uma tabela dos senos hiperbólicos dos ângulos de 0 a 90 graus, de 15 em 15 graus.
- f) Faça um programa para exibir uma tabela dos cossenos hiperbólicos dos ângulos de 0 a 90 graus, de 15 em 15 graus.
- g) Faça um programa para exibir uma tabela dos tangentes hiperbólicos dos ângulos de 0 a 90 graus, de 15 em 15 graus.
- h) Faça um programa para exibir uma tabela dos senos, cossenos e tangentes hiperbólicos dos ângulos de 0 a 90 graus, de 15 em 15 graus.
- i) Crie um programa que leia um número e calcule seus logaritmos natural e decimal.
- j) Faça um programa que, dado um logaritmo natural, exiba o antilogaritmo. Ou seja, dado $\ln n$, exiba n .
- k) Construa um programa que exiba as raízes quadradas dos números de 1 a 100, de 1 em 1.
- l) Monte um programa que mostre os quadrados, cubos, raízes quadradas e raízes cúbicas dos números de 1 a 20, de 1 em 1.

4 TERMINAMOS

Terminamos por aqui. Saia do Dev-C++ e corra para o próximo tutorial.