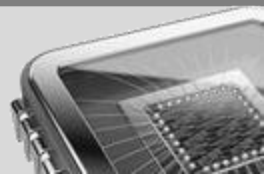
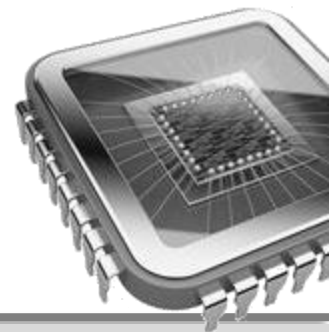


# Introdução à Programação

## Aula 08

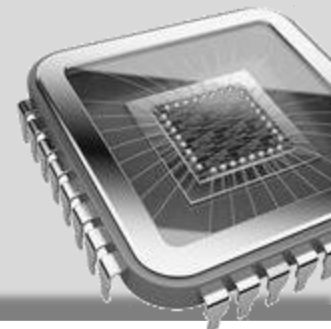
Prof. Max Santana Rolemberg Farias  
max.santana@univasf.edu.br  
Colegiado de Engenharia de Computação





# FUNÇÕES MATEMÁTICAS

# Funções Matemáticas



- As funções matemáticas são declaradas no cabeçalho *math.h*
  - Todas as funções da biblioteca *math.h* retornam um valor do tipo *double*.

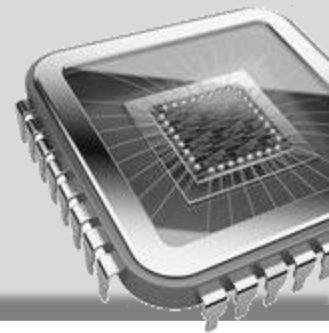
**VOCE**  
**?**  
**SABIA**

Em alguns ambientes de compilação a biblioteca não é automaticamente incorporada. Desse modo, o comando de compilação deve usar a opção *lm*, como ocorre no seguinte comando:

```
gcc prog.c -o prog -lm
```

# Funções Matemáticas

## Valor Absoluto

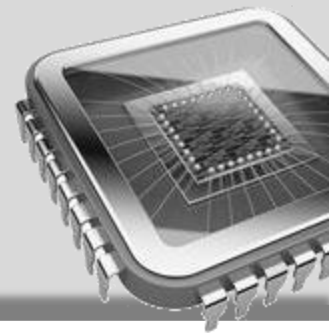


- As funções de valor absoluto da biblioteca `math.h` tem como argumento um número.
- A tabela apresenta as funções de valor absoluto.

Função	Descrição
<code>int abs(int x)</code>	Retorna o valor absoluto de um inteiro <code>x</code>
<code>float fabs(float x)</code>	Retorna o valor absoluto de um número real <code>x</code>

# Funções Matemáticas

## Valor Absoluto



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

int main(){
    int i;
    float j;

    i = -8;
    j = -3.7;

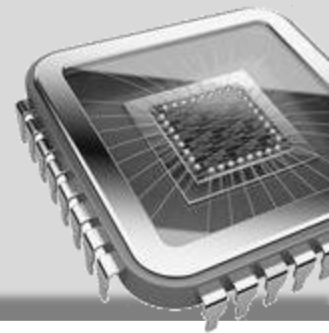
    printf("\n** Utilizando a biblioteca math.h **\n");
    printf("  Funções valor absoluto\n");

    printf(" Valor absoluto de %d é %d\n", i, abs(i));
    printf("Valo absoluto de %f é %f\n", j, fabs(j));

    return 0;
}
```

# Funções Matemáticas

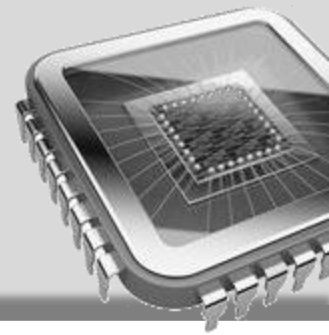
## Trigonométricas



- As funções trigonométricas da biblioteca math.h tem como argumento o ângulo em radianos.
- A tabela apresenta algumas das funções trigonométricas mais utilizadas

Função	Descrição
<code>double cos(double x)</code>	Retorna o cosseno de $x$
<code>double sin(double x)</code>	Retorna o seno de $x$
<code>double tan(double x)</code>	Retorna a tangente de $x$

# Funções Matemáticas Trigonométricas



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

#define PI 3.14

int main(){
    double x0 = 0, x30 = PI/6, x45 = PI/4, x60 = PI/3, x90 = PI/2;

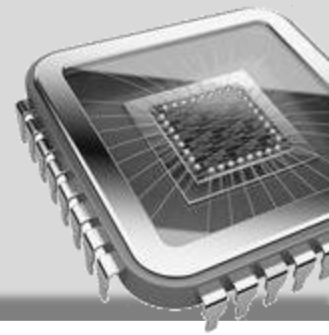
    printf("\n** Utilizando a biblioteca math.h **\n");
    printf("  Funções trigonométricas\n");

    printf(" \t%f rad \t%f rad\t%f rad\t%f rad\t%f rad\n", x0, x30, x45, x60, x90);
    printf("cos\t%f\t%f\t%f\t%f\t%f\n",cos(x0),cos(x30),cos(x45),cos(x60),cos(x90));
    printf("sin\t%f\t%f\t%f\t%f\t%f\n",sin(x0),sin(x30),sin(x45),sin(x60),sin(x90));
    printf("tan\t%f\t%f\t%f\t%f\t%f\n",tan(x0),tan(x30),tan(x45),tan(x60),tan(x90));

    return 0;
}
```

# Funções Matemáticas

## Exponenciais e Potencia

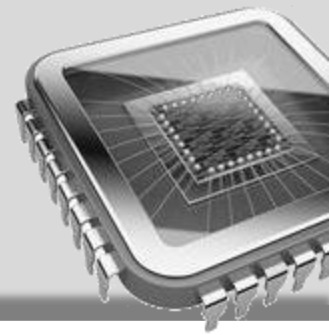


- A tabela apresenta algumas das funções exponenciais e de potencia mais utilizadas

Função	Descrição
<code>double exp(double x)</code>	Retorna o valor de e elevado à x, $e^x$
<code>double exp2(double x)</code>	Retorna o valor de 2 elevado à x, $2^x$
<code>double expm1(double x)</code>	Retorna a subtração da exponenciação x, $e^x - 1$
<code>double sqrt(double x)</code>	Retorna a raiz quadrada de x
<code>double cbrt(double x)</code>	Retorna a raiz cúbica de x
<code>double pow(double x, double y)</code>	Retorna ao resultado de x elevado a y

# Funções Matemáticas

## Exponenciais e Potencia



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

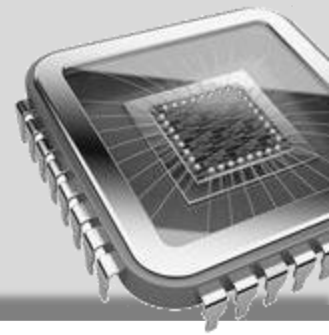
int main(){
    double x = 8.0;

    printf("\n** Utilizando a biblioteca math.h **\n");
    printf("  Funções Exponenciais e Potencia\n");

    printf("exp(%f)\t%f\n",x,exp(x));
    printf("exp2(%f)\t%f\n",x,exp2(x));
    printf("expm1(%f)\t%f\n",x,expm1(x));
    printf("sqrt(%f)\t%f\n",x,sqrt(x));
    printf("cbrt(%f)\t%f\n",x,cbrt(x));
    printf("pow(%f,2)\t%f\n",x,pow(x,2));
    return 0;
}
```

# Funções Matemáticas

## Arredondamento

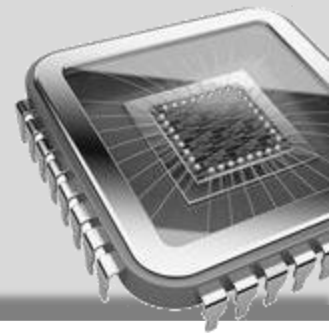


- A tabela apresenta algumas das funções de arredondamento mais utilizadas

Função	Descrição
<b>double</b> floor( <b>double</b> x)	Arredonda x para baixo
<b>double</b> ceil( <b>double</b> x)	Arredonda x para cima

# Funções Matemáticas

## Arredondamento



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

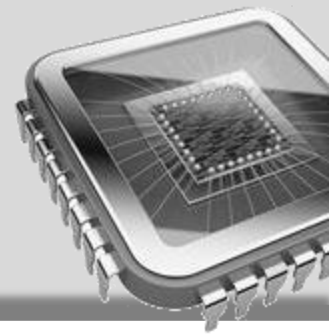
```
int main(){
    double x = 8.69;
```

```
    printf("\n** Utilizando a biblioteca math.h **\n");
    printf("    Funções Arredondamento\n");
```

```
    printf("floor(%f)\t%f\n",x,floor(x)); // arredondamento para baixo
    printf("ceil(%f)\t%f\n",x,ceil(x)); // arredondamento para cima
    return 0;
}
```

# Funções Matemáticas

## Logaritmo

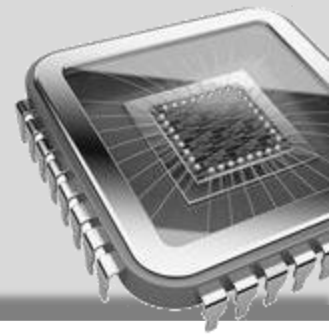


- As funções logaritmo da biblioteca math.h tem como argumento um número.
- A tabela apresenta as funções de logaritmo mais utilizadas.

Função	Descrição
<code>double log(double x)</code>	Retorna o logaritmo natural do número x
<code>double log10(double x)</code>	Retorna o logaritmo na base 10 do número x

# Funções Matemáticas

## Logaritmo



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

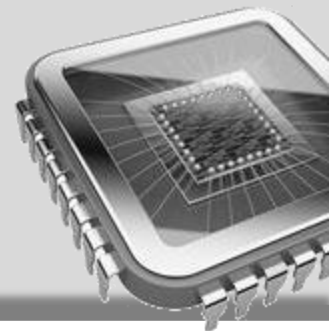
int main(){

    printf("\n** Utilizando a biblioteca math.h **\n");
    printf("  Funções logaritmo\n");

    printf("Logaritmo natural de 30 é %.1f\n", log(30));
    printf("Logaritmo na base 10 de 30 é %.3f\n", log10(30));

    return 0;
}
```

# Números Randômicos

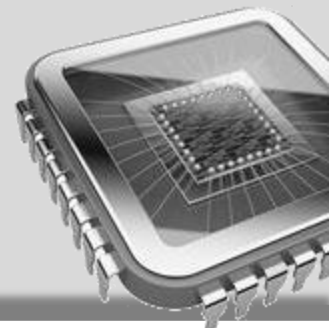


- A função `rand` da biblioteca `stdlib.h`, é comumente usada para gerar uma sequência de números inteiros pseudorrandômicos. A cada execução da função um novo número é obtido.



Para usar corretamente a função `rand`, temos que adicionar a biblioteca `time.h` e `stdlib.h`

# Números Randoômicos



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

```
int main(){
    int i;
```

```
    printf("função rand():
[0,%d]\n",RAND_MAX);
```

```
    for (i=1;i<=10;i++)
        printf("Número %d: %d\n",i, rand());
```

```
    return 0;
```

```
}
```

função rand(): [0,2147483647]

Número 1: 1804289383

Número 2: 846930886

Número 3: 1681692777

Número 4: 1714636915

Número 5: 1957747793

Número 6: 424238335

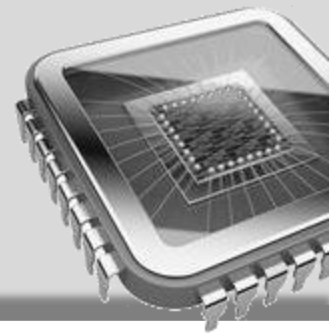
Número 7: 719885386

Número 8: 1649760492

Número 9: 596516649

Número 10: 1189641421

# Números Randômicos



Para fazer isso, basta adicionar antes da rand o seguinte comando:

```
srand((unsigned)time(NULL));
```

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

```
int main(){
    int i;
```

```
    printf("função rand(): [0,%d]\n",RAND_MAX);
    srand((unsigned)time(NULL));
    for (i=1;i<=10;i++)
        printf("Número %d: %d\n",i, rand());
```

```
    return 0;
}
```

função rand(): [0,2147483647]

Número 1: 1804289383

Número 2: 846930886

Número 3: 1681692777

Número 4: 1714636915

Número 5: 1957747793

Número 6: 424238335

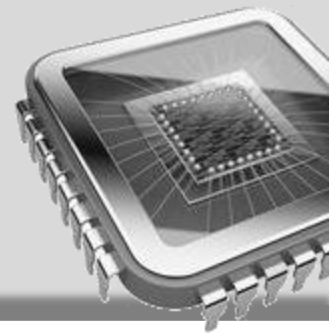
Número 7: 719885386

Número 8: 1649760492

Número 9: 596516649

Número 10: 1189641421

# Faixa de Número Randômicos



## Gerar números entre 0 e 9

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

int main(){
    int i;
    srand((unsigned)time(NULL));
    for (i=1;i<=10;i++)
        printf(“%d\n”, rand()%10);

    return 0;
}
```

## Gerar números entre 1 e 10

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

int main(){
    int i;
    srand((unsigned)time(NULL));
    for (i=1;i<=10;i++)
        printf(“%d\n”, 1+(rand()%10));

    return 0;
}
```