

## Aula 02 - Operadores

### Operadores aritméticos

Operador	Exemplo	Forma contraída	Descrição
+	<code>a =(a + 4);</code>	<code>a+=4;</code>	Realiza a soma par a par de elementos que podem ser números constantes e/ou variáveis
-	<code>b = b - 4;</code>	<code>b-=4;</code>	Realiza a subtração par a par de elementos que podem ser números constantes e/ou variáveis
*	<code>a = a*b*4;</code>	<code>b*=3;</code>	Realiza a multiplicação par a par de elementos que podem ser números constantes e/ou variáveis
/	<code>a = a/(b+4);</code>	<code>a/=10;</code>	Realiza a divisão par a par de elementos que podem ser números constantes e/ou variáveis
%	<code>a = a%3;</code>	<code>a%=3;</code>	Retorna o resto da divisão entre números inteiros. No caso do exemplo, supondo que o valor de a seja 10, o valor de a, ao final da operação, seria 1, pois $10/3 = 3$ e resta 1.

### Incremento e decremento (++ e --)

Tais operadores podem ser pré-fixados ou pós-fixados. Seu efeito é diferente dependendo do contexto.

## Funções matemáticas

gcc codigo.c -o saida -lm

Função	Descrição
<code>double <b>ceil</b>(double x);</code> <code>float <b>ceilf</b>(float x);</code> <code>long double <b>ceil</b>(long double x);</code>	Arredonda um número sempre para o próximo inteiro maior.
<code>double <b>floor</b>(double x);</code> <code>float <b>floorf</b>(float x);</code> <code>long double <b>floorl</b>(long double x);</code>	Arredonda um número real para baixo, independente do valor da parte decimal
<code>double <b>round</b>(double x);</code> <code>float <b>roundf</b>(float x);</code> <code>long double <b>roundl</b>(long double x)</code>	Arredonda um número real de forma usual, ou seja se a parte decimal for maior que 0.5, o número será arredondado para cima.
<code>double <b>sin</b>(double x);</code> <code>float <b>sinf</b>(float x);</code> <code>long double <b>sinl</b>(long double x);</code>	Calcula o seno de X radianos
<code>double <b>cos</b>(double x);</code> <code>float <b>cosf</b>(float x);</code> <code>long double <b>cosl</b>(long double x);</code>	Calcula o cosseno de X radianos.
<code>double <b>tan</b>(double x);</code> <code>float <b>tanf</b>(float x);</code> <code>long double <b>tanl</b>(long double x);</code>	Calcula a tangente de X
<code>double <b>pow</b>(double x, double y);</code> <code>float <b>powf</b>(float x, float y);</code> <code>long double <b>powl</b>(long double x, long double y);</code>	Calcula $x^y$
<code>double <b>exp</b>(double x);</code> <code>float <b>expf</b>(float x);</code> <code>long double <b>expl</b>(long double x);</code>	Calcula $e^x$ , sendo e a base natural dos logaritmos
<code>double <b>log</b>(double x);</code> <code>float <b>logf</b>(float x);</code>	Calcula $\log(x)$ , logaritmo

<code>long double <b>logl</b>(long double x);</code>	
<code>double <b>log10</b>(double x); float <b>log10f</b>(float x); long double <b>log10l</b>(long double x);</code>	Calcula log(x) na base 10
<code>double <b>modf</b>(double x, double *y); float <b>modff</b>(float x, float* y); long double <b>modfl</b>(long double x, long double *y)</code>	Retorna o valor fracionário de um número e em y escrito a parte inteira.
<code>double <b>sqrt</b>(double x); float <b>sqrtof</b>(float x); long double <b>sqrtl</b>(long double x);</code>	Retorna a raiz quadrada do número x.
<code>double <b>fabs</b>(double x); float <b>fabsf</b>(float x); long double <b>fabsl</b>(long double x);</code>	Retorna o valor absoluto do número x.

## Operadores relacionais

Operador	Efeito
<	Menor
<=	Menor ou igual
>	Maior
>=	Maior ou igual
==	Igual
!=	Diferente

Entendimento para com o computador: "igual a 0" é falso e "diferente de 0" é verdadeiro.

## Operadores lógicos

Operador	Efeito
&&	"E" lógico
	"Ou" lógico
!	"Não" ou inversor de valor booleano

Apresentar as tabelas-verdade.

## Operadores lógicos bit a bit

Representação dos dados. Binário, octal e hexadecimal.

Sistema posicional - Inicie da base decimal

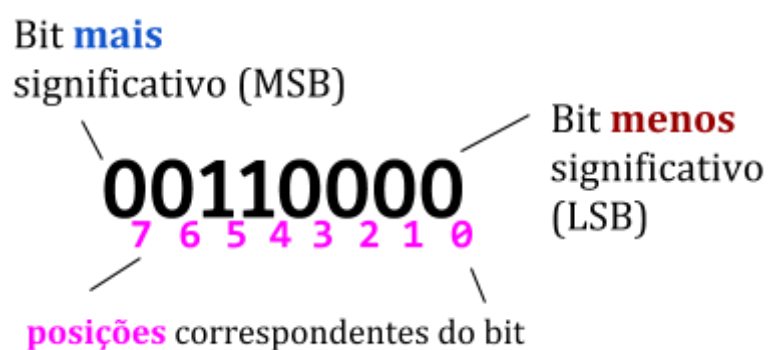


Figura 1 - Representação de um byte como um inteiro de 8 bits.

Operador	Significado	Exemplo de uso
----------	-------------	----------------

~	complemento bit a bit	~x
&	AND bit a bit	x & 64
	OR bit a bit	y = x   32
^	XOR bit a bit	z = x^y
<<	deslocamento de bits à esquerda	x << 2
>>	deslocamento de bits à esquerda	x >> 3

Tabela 1 - operadores bit a bit da linguagem C.

OBS: Os operadores bit a bit só podem ser usados nos tipos **char**, **int** e **long** (BACKES, 2012).