

# Operadores aritméticos

[emanoelim@utfpr.edu.br](mailto:emanoelim@utfpr.edu.br)

# Operadores aritméticos

- Um operador é um símbolo que opera sobre um valor ou variável, gerando um resultado.
- O símbolo +, por exemplo, serve para realizar uma adição entre dois operandos.
- Na linguagem C existe uma grande quantidade de operadores.
- Primeiramente, serão estudados os operadores aritméticos.

# Operadores aritméticos

- Operadores aritméticos:

Operador	Ação
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Resto



# Operadores aritméticos

- Exemplo:

```
#include <stdio.h>
main() {
    int a = 9, b = 4, c;
    c = a + b;
    printf("a + b = %d \n", c);
    c = a - b;
    printf("a - b = %d \n", c);
    c = a * b;
    printf("a * b = %d \n", c);
    c = a / b;
    printf("a / b = %d \n", c);
    c = a % b;
    printf("Resto da divisão de a por b = %d \n", c);
}
```

# Operadores aritméticos

- Operador de resto (%):
  - Só pode ser aplicado a valores do tipo int;

- Exemplo:

$$2018 \% 3 = 2$$

pois: 
$$\begin{array}{r} 2018 \overline{) 3} \\ -18 \phantom{00} \underline{672} \\ 21 \\ -21 \\ 8 \\ -6 \\ 2 \end{array}$$

**2 Resto**

# Operadores aritméticos

- Operador de resto (%):
  - Se o dividendo for menor que o divisor, a divisão sempre terá resultado 0 e o resto sempre será o próprio dividendo. Ex.:

$$10 \% 30 = 10$$

$$1 \% 4 = 1$$

$$3 \% 15 = 3$$



# Operadores aritméticos

- O operador de resto é muito útil para verificar se um número é par ou ímpar, pois:
  - O resto de um número par dividido por 2 é sempre igual a 0.
  - O resto de um número ímpar dividido por 2 é sempre igual a 1.

# Operadores aritméticos

- Precedência dos operadores:
  - $*$ ,  $/$  e  $\%$  tem precedência sobre  $+$  e  $-$
  - Esta ordem pode ser mudada usando parênteses:  
 $(x + y) * 2$
  - As operações são feitas da esquerda para a direita:  
 $x = 5 - 2 + 8$   
Primeiro será feito  $5 - 2$  e ao resultado disso será adicionado 8;



# Operadores aritméticos

- Operadores de incremento e decremento:

Operador	Ação
++	Incremento
--	Decremento

# Operadores aritméticos

- Incremento:

```
#include <stdio.h>

main() {
    int a = 0;
    a++;
    printf("Novo valor de a: %d", a);
}
```

O novo valor de  
“a” será 1

**a++** equivale a: **a = a + 1**

# Operadores aritméticos

- Decremento:

```
#include <stdio.h>

main() {
    int a = 1;
    a--;
    printf("Novo valor de a: %d", a);
}
```

O novo valor de  
“a” será 0

**a--** equivale a: **a = a - 1**



# Operadores aritméticos

- Incrementando outros valores:

```
#include <stdio.h>

main() {
    int a = 1;
    a+=3;
    printf("Novo valor de a: %d", a);
}
```

O novo valor de  
“a” será 4

**a+=3** equivale a: **a = a + 3**

# Operadores aritméticos

- Decrementando outros valores:

```
#include <stdio.h>

main() {
    int a = 5;
    a-=3;
    printf("Novo valor de a: %d", a);
}
```

O novo valor de  
“a” será 2

**a-=3** equivale a: **a = a - 3**

# Operadores aritméticos

- O operador = pode ser usado com +, -, /, \* e %:

Operador	Exemplo	Equivale a
+=	a += b	a = a + b
-=	a -= b	a = a - b
*=	a *= b	a = a * b
/=	a /= b	a = a / b
%=	a %= b	a = a % b



# Operadores aritméticos

- Conversão de tipos:
  - Se tentarmos dividir dois números inteiros, o resultado sempre será um número inteiro. Por exemplo:

```
...  
int x = 10;  
int y = 3;  
float z = x / y;  
printf("A resposta da divisão é: %.2f\n", z);  
...
```

Apesar de declarar a variável que armazena a resposta como float, a resposta mostrada será 3.00, pois será considerada apenas a parte inteira da divisão.

# Operadores aritméticos

- Conversão de tipos:
  - Se quisermos mostrar o resultado desta divisão considerando a parte fracionária, será necessário converter o divisor ou o dividendo para float.
  - Uma conversão de tipo é feita da seguinte forma:

`(novo_tipo)variável;`

# Operadores aritméticos

- O exemplo anterior ficaria:

```
...  
int x = 10;  
int y = 3;  
float z = x / (float)y; // ou (float)x / y;  
printf("A resposta da divisão é: %.2f\n", z);  
...
```

Agora a resposta mostrada é  
3.33



# Biblioteca math.h

- Outras operações como potência, raiz quadrada, seno, cosseno e etc, podem ser feitas utilizando funções da biblioteca math.h.

```
#include <stdio.h>
#include <math.h> // inclusão da biblioteca

int main(void) {
    // algumas funções interessantes da math.h
    printf("A raiz de 25 é: %.2f \n", sqrt(25));
    printf("2^3 é: %.0f \n", pow(2, 3)); // precisamos passar a base e o expoente
    printf("O modulo de -290 é %.0f \n", fabs(-290));
    printf("Arredondando 2.40 para cima: %.2f \n", ceil(2.40));
    printf("Arredondando 2.40 para baixo: %.2f \n", floor(2.40));
}
```

# Biblioteca math.h

- Outros exemplos:

<https://www.geeksforgeeks.org/c-library-math-h-functions/>