

# INF 01040

# Introdução à Programação

Expressões aritméticas avançado

# Um dia, um grupo, em uma pizzeria: o problema...

- 2 das nossas pizzas, disse o garçom, dá e sobra para 3 pessoas... como vocês estão em 21

*2 pizzas → 3 pessoas*

*Y pizzas → 21 pessoas*


- Ou seja

$$Y = \frac{2 * 21}{3} = \frac{42}{3} = 14 \text{ pizzas}$$



## Como escrever essa expressão em C?

# Resposta: usando operadores aritméticos segundo as regras sintáticas do C

Operador	Ação	Prioridade
-- ++	Decremento e Incremento	Maior
-	Subtração unária (inverte o sinal)	
* / %	Multiplicação, Divisão e Resto da divisão inteira	
- +	Subtração e Adição	
		Menor

Observação: em uma expressão entre operadores no mesmo nível de precedência, o cálculo é sempre feito a partir da esquerda para a direita.

# ... no caso da pizzeria

- A expressão:

$$Y = \frac{2 * 21}{3}$$

- Escrita em C, fica:

`Y = 2 * 21 / 3;`

- A expressão será avaliada pelo computador da esquerda para a direita já que os operadores tem a mesma prioridade:

*2 \* 21 = 42 e depois 42 / 3 = 14*



# E se os operadores forem de diferentes prioridades?

- Seja a expressão:

$$X = \frac{a+b}{2}$$

- Se definirmos que  $a = 5$  e  $b = 3$ , o valor calculado será  $X = 4$

## Implementando em C

### Certo

```
x = (a + b) / 2;
```


Os parênteses determinam que a expressão  $a + b$  deve ser avaliada antes da divisão por 2

### Errado

```
x = a + b / 2;
```

A ordem de precedência dos operadores faz com que  $b / 2$  seja avaliado antes, resultado em um valor incorreto para X

# Atualizando a tabela de operadores

Operador	Ação	Prioridade
( )	Obrigam a operação entre ( e ) a ser realizada primeiramente	Maior
-- ++	Decremento e Incremento	
-	Subtração unária (inverte o sinal)	
* / %	Multiplicação, Divisão e Resto da divisão inteira	
- +	Subtração e Adição	Menor

Observação: parênteses têm precedência maior que todos os outros operadores

A lista completa de precedência está disponível em:  
[http://en.cppreference.com/w/c/language/operator\\_precedence](http://en.cppreference.com/w/c/language/operator_precedence)

# Outros exemplos

Expressão aritmética em texto de álgebra	Em C	Análise
$\frac{(a+b+c+d)}{4}$	<code>(a+b+c+d) / 4 ;</code>	Parêntesis são obrigatórios para efetivar a soma dos 4 valores antes da divisão. Se não usados, d será dividido (/) por 4 e depois serão efetivadas a soma de d/4 com a, b, c.
$\frac{b^2}{4ac}$	<code>b*2/ (4*a*c) ;</code>	Todos os operadores devem ser explicitamente escritos, senão b2 é entendido como um identificador, enquanto que b*2 é uma expressão aritmética.
$(a+b)^2$	<code>(a+b) * (a+b) ;</code>	Não existe operador potência, então aplica-se o conceito, multiplicando-se 2 vezes.
$\frac{1+C}{A-B}$	<code>(1+C) / (A-B) ;</code>	Os identificadores devem ser transcritos para formato em minúsculo, ou caixa baixa. Observar a necessidade dos parêntesis.
$4 * \pi * r^3$	<code>4*3.1416*r*r*r ;</code>	A constante Pi é um caractere grego, sem significado para C. Neste caso, deve-se colocar o valor aproximado.

# Exemplos com operadores de incremento e decremento

Expressão	Ação	Equivale a:
<b>++x ;</b> ou <b>x++ ;</b>	Acrescenta 1 ao operando x. Isto é, se $x = 5$ , depois da expressão ++x ou x++, x passa a conter o valor 6.	$x = x + 1$ x recebe o novo valor
<b>--y ;</b> ou <b>y-- ;</b>	Subtrai 1 do operador y. Isto é, se $y = 8$ , então após a expressão --y ou y--, y passa a valer 7.	$y = y - 1$ y recebe novo valor



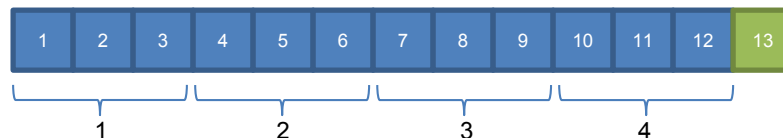
# Exemplo de uso de resto da divisão inteira

- Em uma divisão inteira em C a parte flutuante do resultado é truncada, isto é, em:

`x = 13 / 3;`

*X recebe o valor inteiro 4*

- O resto da divisão de 13 por 3 é igual a 1



- Ou seja:

`x = 13 % 3;`

*X recebe o valor inteiro 1*

# Quando os operadores básicos não são suficientes...

- A biblioteca **math.h** fornece algumas funções bastante úteis em C:

Função	Descrição	Exemplo em C
<code>sqrt(x)</code>	Calcula a raiz quadrada do número $x$	<code>y = sqrt(x) ;</code>
<code>pow(b, e)</code>	Eleva o número $b$ na potência $e$	<code>y = pow(x, 5) ;</code>
<code>sin(x)</code>	Calcula o seno de $x$	<code>y = sin(x) ;</code>
<code>cos(x)</code>	Calcula o cosseno de $x$	<code>y = cos(x) ;</code>
<code>log(x)</code>	Calcula o logaritmo natural de $x$	<code>y = log(x) ;</code>

A lista completa de funções disponíveis na biblioteca **math.h** pode ser encontrada em <http://www.cplusplus.com/reference/cmath/>