

Linguagem de Programação I - ESP201

Prof^a Ana Carolina Sokolonski

Bacharelado em Sistemas de Informação
Instituto Federal de Ciência e Tecnologia da Bahia
Campus de Feira de Santana

carolsoko@ifba.edu.br

June 3, 2025

Operações Matemáticas e Conversões de Tipos

- 1 Operações Matemáticas
 - Precedência de Operadores
 - Comparações e Testes
- 2 Conversões de Tipos
- 3 Exercícios
- 4 Biblioteca Math.h
- 5 Exercícios
- 6 Referências

Operações Matemáticas

Operadores Aritméticos

Operador	Ação
+	Adição
*	Multiplicação
/	Divisão
%	Resto da divisão inteira
-	Subtração (unário)
--	Decremento
++	Incremento

Operadores Relacionais e Lógicos

Operador	Ação
>	Maior que
>=	Maior ou igual que
<	Menor que
<=	Menor ou igual que
==	Igual a
!=	Diferente de
&&	Condição "E"
	Condição "OU"
!	Não

Operadores Bit-a-Bit

Operador	Ação
&	AND
	OR
^	XOR (OR exclusivo)
<<	Deslocamento para esquerda
>>	Deslocamento para direita

Precedência de Operadores

Precedência	Operador
Maior	() [] ->
	! ~ ++ -- - (tipo) * & sizeof
	* / %
	+ -
	<< >>
	< <= > >=
	== !=
	&
	^
	!
	&&
	!!
	?:
	= += -= *= /= etc.
Menor	,

Comparações e Testes

```
if (10 > 4 && !(10 < 9) || 3 <= 4)
```


Comparações e Testes

```
if (10 > 4 && !(10 < 9) || 3 <= 4)
```

Como seria avaliada esta instrução?

Comparações e Testes

```
if (10 > 4 && !(10 < 9) || 3 <= 4)
```

Como seria avaliada esta instrução?

Resultaria em **Verdadeiro**, pois **(10 é maior que 4) E (10 não é menor que 9) OU (3 é menor ou igual a 4)**

Comparações e Testes - Observações

Em C o resultado da comparação será **ZERO** se resultar em **FALSO** e **DIFERENTE DE ZERO** no caso de obtermos **VERDADEIRO** num teste qualquer.

```
#include <stdio.h>

void main ( ){
    int verdadeiro, falso;

    verdadeiro = (15 < 20);
    falso      = (15 == 20);
    printf("Verd.= %d,Falso= %d",
        verdadeiro, falso);
}
```

Saída: Verd.=1 Falso = 0

Comparações e Testes

Se $x = 1$ e $y = 2$, qual será a saída para o seguinte comando:

```
printf("%d == %d é %d \n", x, y, x == y);
```

Comparações e Testes

Se $x = 1$ e $y = 2$, qual será a saída para o seguinte comando:

```
printf("%d == %d é %d \n", x, y, x == y);
```

resultaria em: **1 == 2 0**, pois a expressão é falsa

Comparações e Testes - ATENÇÃO

Observemos antes de mais nada que $++x$ é diferente de $x++$!

Vamos avaliar o $++x$:

Se

```
x = 10;
```

```
y = ++x;
```

```
/* x=x+1;    y=x; */
```

então

```
x = 11 e
```

```
y = 11
```

Comparações e Testes - ATENÇÃO

Observemos antes de mais nada que $++x$ é diferente de $x++$!

Vamos avaliar o $x++$:

porém Se

```
x = 10;
```

```
y = x++;
```

```
/* y=x;    x=x+1 */
```

então

```
x = 11 e
```

```
y = 10
```

Comparações e Testes - ATENÇÃO

Exemplo:

```
1
2 #include <stdio.h>
3 void main(){
4     int cont = 0, loop;
5
6     loop = ++cont;
7     printf ("Loop=%d, Cont=%d\n", loop, cont);
8
9     loop = cont++;
10    printf("Loop=%d, Cont=%d\n", loop, cont);
11 }
```


Comparações e Testes - ATENÇÃO

Exemplo:

```
1
2 #include <stdio.h>
3 void main() {
4     int cont = 0, loop;
5
6     loop = ++cont;
7     printf("Loop=%d, Cont=%d\n", loop, cont);
8
9     loop = cont++;
10    printf("Loop=%d, Cont=%d\n", loop, cont);
11 }
```

< [Icons]

Loop=1, Cont=1
Loop=1, Cont=2

...Program finished with exit code 15
Press ENTER to exit console.

Conversões de Tipos

A linguagem C é uma linguagem **fortemente tipada**, ou seja, todas as variáveis da linguagem C têm um tipo específico (bem definido, que tem que ser declarado) e o tipo da variável é importante para a linguagem C, esse tipo interfere em suas operações e em como a linguagem trata as suas variáveis [Sebesta 2010].

Conversões de Tipos

Quando forem misturadas variáveis de diferentes tipos, o compilador C converterá os operandos para o tipo de operando maior (**em tempo de compilação**) de acordo com as regras descritas a seguir, não podendo alterar os tipos das variáveis depois de concluída a compilação (**em tempo de execução**) [Schildt e Mayer 1997].

Conversões de Tipos - REGRAS

- 1 Todo **char** e **short int** é convertido para **int**.
- 2 Todo **float** é convertido para **double**.
- 3 Para os demais pares de operandos valem as seguintes regras em sequência:
 - 1 Se um operando for **long double**, o outro também o será.
 - 2 Se um operando for **double**, o outro também o será.
 - 3 Se um operando for **long**, o outro também o será.
 - 4 Se um operando for **unsigned**, o outro também o será.
- 4 Observe que a operação à direita é realizada antes da atribuição, assim: **float x = 10/3** , terá resultado **inteiro**.

Conversões de Tipos - CAST

- Este operador tem como função forçar para que uma expressão seja de um determinado tipo. Sua sintaxe é:
(tipo de dado) expressão
- Assim, o problema anterior, onde:
float x = 10/3, seria resolvido se fizéssemos:
float x = (float) 10/3
- O tipo de dado pode ser qualquer um dos definidos pelo padrão ANSI. Um exemplo da utilização deste operador pode ser visto a seguir:

CAST

```
2  #include <stdio.h>
3
4  void main ( ){
5      int x,y;
6      float divi, divisao;
7
8      printf ("Digite dois numeros inteiros\n");
9      scanf("%d\n%d", &x, &y);
10
11     divisao = x/y;
12     divi = (float) x/y;
13
14     printf ("x =%d\ny =%d\nDivisão inteira = %f\nDivisão = %f\n",x,y, divisao, divi);
15 }
```

CAST

```
2  #include <stdio.h>
3
4  void main ( ){
5      int x,y;
6      float divi, divisao;
7
8      printf ("Digite dois numeros inteiros\n");
9      scanf ("%d\n%d", &x, &y);
10
11     divisao = x/y;
12     divi = (float) x/y;
13
14     printf ("x =%d\n y =%d\n Divisão inteira = %f\n Divisão = %f\n", x,y, divisao, divi);
15 }
```

< input

```
< Digite dois numeros inteiros
34
8
x =34
y =8
Divisão inteira = 4.000000
Divisão = 4.250000
```

Exercícios

- 1 Escreva um programa para ler o raio de um círculo, calcular e escrever a sua área.
- 2 Um motorista de táxi deseja calcular o rendimento de seu carro na praça. Sabendo-se que o preço do combustível é de R\$1,90, escreva um programa para ler: a marcação do odômetro (Km) no início do dia, a marcação (Km) no final do dia, o número de litros de combustível gasto e o valor total (R\$) recebido dos passageiros. Faça um algoritmo que calcule e mostre a média do consumo em Km/L e o lucro líquido do dia.
- 3 Faça um algoritmo que leia a quantidade de pêras compradas em uma mercearia e o valor da unidade. O algoritmo deve informar o total a ser pago pelo cliente.

Funções da Biblioteca Math.h

- 1 `sqrt(x)`; - Raiz quadrada de x
- 2 `pow(x,y)`; - x elevado a y
- 3 `acos(x)`; - cosseno de x (x é um ângulo)
- 4 `asin(x)`; - seno de x (x é um ângulo)
- 5 `atan(x)`; - tangente de x (x é um ângulo)
- 6 `log(x)`; - retorna $\ln(x)$
- 7 `ceil(x)`; - retorna o valor de x arredondado para cima
- 8 `floor(x)`; - retorna o valor de x arredondado para baixo

Funções da Biblioteca Math.h

```
1  #include <stdio.h>
2  #include <math.h>
3  #include <stdlib.h>
4
5  void main (){
6
7      double x = 8.62; |
8
9      printf("Valor aproximado para baixo de %f é %f\n",x, floor(x));
10     printf("Valor aproximado para cima de %f é %f\n", x, ceil(x));
11
12     printf("Raiz quadrada de %f é %f\n",x,sqrt(x));
13     printf("%.2lf ao quadrado é %.2f \n",x,pow(x,2));
14
15     printf("Valor de seno de %.2f = %.2f \n",x,sin(x));
16     printf("Valor de cosseno de %.2f = %.2f \n",x,cos(x));
17     printf("Valor de tangente de %.2f = %.2f \n",x,tan(x));
18
19     printf("Logaritmo natural de %.2f = %.2f \n",x,log(x));
20     printf("Logaritmo de %.2f na base 10 = %.2f \n",x,log10(x));
21     printf("Exponencial de %.2f = %e \n",x,exp(x));
22
23     printf("O valor aproximado de pi é %e \n",M_PI);
24     printf("O valor aproximado de pi/2 é %e \n",M_PI_2);
25
26     printf("O módulo de -3.2 é %f \n",fabs(-3.2));
27     printf("O módulo de -3 é %d \n",abs(-3));
28 }
```

Exercícios

- 1 Faça um algoritmo que calcula o delta e as duas raízes de uma equação a partir da leitura dos coeficientes a , b e c da fórmula de Bháskara, $\text{delta} = b * b - 4 * a * c$;
 $x_1 = (-b - \text{sqrt}(\text{delta})) / (2 * a)$; e
 $x_2 = (-b + \text{sqrt}(\text{delta})) / (2 * a)$;
- 2 O coração humano bate em média uma vez por segundo. Desenvolver um algoritmo para calcular e escrever quantas vezes o coração de uma pessoa baterá se viver X anos.
- 3 Muitos países estão passando a utilizar o sistema métrico. Fazer um algoritmo para executar as seguintes conversões: Ler uma temperatura em Fahrenheit e imprimir o equivalente em Celsius $C = (5F - 160) / 9$. Ler uma quantidade de chuva dada em polegadas e imprimir o equivalente em milímetros ($1'' = 25,4 \text{ mm}$).



Exercícios

- 1 Desenvolver um algoritmo para ler os comprimentos dos três lados de um triângulo (L1, L2 e L3) e calcular a área do triângulo. Pesquisar na internet a fórmula da área de um triângulo, transformar para a linguagem C.
- 2 Desenvolver um algoritmo para ler o nome, idade e peso de três pessoas. Calcular e imprimir idade e o peso médio das três pessoas.
- 3 Dados dois pontos A e B, cujas coordenadas A(x1, y1) e B(x2, y2) serão informadas via teclado, desenvolver um algoritmo que calcule a distância entre A e B.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Referências

Referências

-  SCHILDT, H.; MAYER, R. *C completo e total*. [S.l.]: Pearson University, 1997. ISBN 9788534605953.
-  SEBESTA, R. *Conceitos de Linguagens de Programação*. [S.l.]: Bookman, 2010. ISBN 9788536301716.