

Linguagem de Programação I

Aula 4 – Operadores, Conversões de Tipos e Biblioteca Math.h

Prof Ana Carolina Sokolonski

Bacharelado em Sistemas de Informação

Instituto Federal da Bahia – Campus Feira de Santana

2026



Operadores

Comparações e Testes

Conversões de Tipos

Biblioteca Math.h

Exercícios

Referências

Operadores

Aritméticos, Relacionais, Lógicos e Bit-a-Bit

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

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

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

Expressões lógicas em C

▪ Expressão

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

▪ Parte 1

10 > 4

VERDADEIRO

▪ Parte 2

!(10 < 9)

VERDADEIRO

▪ Parte 3

3 <= 4

VERDADEIRO

▪ Resultado

VERDADEIRO && VERDADEIRO || VERDADEIRO ⇒ VERDADEIRO

■ Importante

Em C, o resultado de uma comparação é:

0 ⇒ **FALSO**

qualquer valor ≠ 0 ⇒ **VERDADEIRO**

■ Exemplo

Se `x = 1` e `y = 2`:

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

Saída: `1 == 2 é 0`

(expressão falsa ⇒ resultado 0)

```
#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

▪ Atenção: ++x \neq x++

Ambos incrementam x em 1, mas **em momentos distintos**.

▪ ++x — Pré-incremento

Incrementa **antes** de usar o valor na expressão.

```
Se
x = 10;
y = ++x;
/* x=x+1; y=x; */
então
x = 11 e
y = 11
```

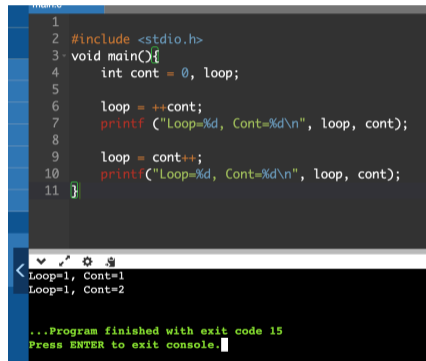
▪ x++ — Pós-incremento

Usa o valor na expressão **antes** de incrementar.

```
porém Se
x = 10;
y = x++;
/* y=x; x=x+1 */
então
x = 11 e
y = 10
```

Exemplo Prático: ++x vs x++

```
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 }
```



```
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 }
```

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

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

Conversões de Tipos

Type Casting em C

▪ C é Fortemente Tipada

Todas as variáveis têm um tipo específico que deve ser declarado. O tipo interfere nas operações e na forma como a linguagem trata os dados. [Sebesta 2010]

▪ Conversão Automática (Implícita)

Quando operandos de tipos diferentes são misturados, o compilador converte automaticamente **em tempo de compilação** para o tipo “maior” (mais abrangente). [Schildt e Mayer 1997]



Regras gerais

- 1 `char` e `short int` → `int`
- 2 `float` → `double`

Armadilha comum!

A operação à **direita** da atribuição é realizada *antes* da atribuição:

```
float x = 10/3;
```

Resultado: **3.0** (inteiro!),
pois `10` e `3` são `int`.

Solução: usar **cast** explícito.

Para pares de operandos

- 1 Um é `long double` → outro vira `long double`
- 2 Um é `double` → outro vira `double`
- 3 Um é `long` → outro vira `long`
- 4 Um é `unsigned` → outro vira `unsigned`

▪ Sintaxe

Força uma expressão a ser tratada como um tipo específico:

(tipo) expressão

▪ Corrigindo a divisão

```
float x = 10/3;    // 3.0
```

```
float x = (float)10/3; // 3.33
```

```
1 #include <stdio.h>
2
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 }
```

Input

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

Biblioteca Math.h

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

▪ Funções básicas

- `sqrt(x)` — raiz quadrada de x
- `pow(x,y)` — x^y
- `ceil(x)` — arredonda para cima
- `floor(x)` — arredonda para baixo
- `log(x)` — logaritmo natural $\ln(x)$

▪ Funções trigonométricas

- `cos(x)` — cosseno (x em radianos)
- `sin(x)` — seno (x em radianos)
- `tan(x)` — tangente (x em radianos)
- `acos(x)`, `asin(x)`, `atan(x)`

▪ Compilação com GCC

Ao usar `math.h`, lembre de linkar:

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

Exemplo de Uso da 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

Pratique o que aprendeu!

- 1 Escreva um programa que leia o **raio** de um círculo, calcule e exiba a sua **área**.
($A = \pi r^2$)
- 2 Um motorista de táxi deseja calcular o rendimento do dia. Leia: marcação do odômetro no **início** e no **final** do dia (km), litros de **combustível** gastos (preço R\$ 1,90/L) e valor **recebido dos passageiros**. Calcule e exiba a média de consumo (km/L) e o **lucro líquido**.
- 3 Leia a **quantidade** e o **valor unitário** de pêras compradas numa mercearia. Exiba o **total a pagar**.

- ① Leia os coeficientes a , b e c de uma equação do 2º grau. Calcule Δ e as raízes x_1 e x_2 pela **Fórmula de Bhaskara**:

$$\Delta = b^2 - 4ac \quad x_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a}$$



- ② O coração humano bate **1 vez por segundo**. Leia quantos anos uma pessoa viveu e calcule o **total de batimentos**.

- ③ Conversões de unidades:

- Leia uma temperatura em **Fahrenheit** e converta para **Celsius**: $C = (5F - 160)/9$
- Leia uma quantidade de chuva em **polegadas** e converta para **mm**: $1'' = 25,4 \text{ mm}$

- 1 Leia os três lados de um triângulo (L_1, L_2, L_3) . Pesquise a **fórmula de Heron** e calcule a **área** do triângulo em C.
- 2 Leia o **nome**, **idade** e **peso** de três pessoas. Calcule e exiba a **média** de idade e de peso.
- 3 Dados dois pontos $A(x_1, y_1)$ e $B(x_2, y_2)$, calcule a **distância** entre eles:

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

-  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.