**COMENTÁRIOS**

/\* *espaço para comentários* e*m mais de uma linha* \*/

// *espaço para comentários em uma única linha, inclusive após comandos*

**BIBLIOTECAS**

#include <stdio.h> *biblioteca de funções de entrada e saída*

#include <stdlib.h> *bibliotecas de funções mais utilizadas*

#include <alloc.h> *funções para gerenciamento de memória.*

#include <float.h> *funções para tratar números de ponto flutuante.*

#include <math.h> *funções matemáticas.*

#include <stddef.h> *vários tipos de dados e macro substituições.*

#include <string.h> *rotinas p/ manipular strings e memória.*

#include <conio.h> *rotinas de entrada e saída, com tabela de cores no vídeo.*

**ENTRADA E SAÍDA DE DADOS**

*observação: ponto e vírgula para separar comandos “ ; “*

scanf(“%x”, &variável); *entrada de dados*

scanf(“%[^\n], &string); *entrada de strings (+ de 1 caracter)*

resp = getche(); *captura um caracter e guarda em resp*

printf(“mensagem: “); *impressão de mensagem*

printf(“mensagem: %x.yf”, variável); *impressão de mensagem e variável*

printf(“mensagem: %s”, string); impressão de mensagem e strings

&  *Operador Endereço (da variável)*

x.y *casas à esquerda e direita da vírgula*

*observação: caracteres são definidos com aspas simples: variável – ‘A’*

*observação: strings são definidas com aspas duplas: variavel = “texto qualquer”*

**PAUSA ATÉ DIGITAÇÃO DE ALGO**

system(“pause”); *aguarda a digitação de uma tecla*

system(“cls”)  *limpa a tela*

getch(); *aguarda a digitação de uma tecla*

return 0; *termina a função main()*

**TIPOS DE DADOS**

int *inteiros de -32.768 até 32.767*

unsigned int *inteiros positivos de 0 até 65.535*

long *inteiros de -2.147.483.648 até 2.147.483.647*

float *reais de 3.4 E-38 até 3.4 E+38*

double *reais de 1.7 E-308 até 1.7 E+308*

char *caracteres com 1 posição*

char\*  *caracteres com até 255 posições*

char variavel[tamanho] *caracteres com até 255 posições (string.h)*

*lógicos: 0 é falso, 1 é verdadeiro*

**DECLARAÇÃO DE TIPO DE VARIÁVEIS**

Iniciar pelo **tipo**,

seguido do nome das variáveis separadas por **vírgula**,

terminar com **ponto e vírgula**

**CÓDIGOS DE FORMATAÇÃO PARA A FUNÇÃO SCANF()**

%c *Permite que seja efetuada a leitura de apenas um caractere*

%d *Permite que seja efetuada a leitura de números inteiros decimais*

%e *Permite que seja efetuada a leitura de números em notação científica*

%f *Permite que seja efetuada a leitura de números reais (ponto flutuante)*

%l *Permite que seja efetuada a leitura de um número inteiro longo*

%o *Permite que seja efetuada a leitura de números octais*

%s *Permite que seja efetuada a leitura de uma série de caracteres*

%u *Permite que seja efetuada a leitura de um número decimal sem sinal*

%x *Permite que seja efetuada a leitura de um número hexadecimal*

%s *Permite que seja efetuada a leitura de strings sem espaço*

%[^\n] *Permite que seja efetuada a leitura de strings com espaço*

**CÓDIGOS DE FORMATAÇÃO PARA A FUNÇÃO PRINTF()**

%c *Permite que seja efetuada a escrita de apenas um caractere*

%d *Permite que seja efetuada a escrita de números inteiros decimais*

%e *Permite que seja efetuada a escrita de números em notação científica*

%f *Permite que seja efetuada a escrita de números reais (ponto flutuante)*

%g *Permite que seja efetuada a escrita de %e ou %f noformato mais curto*

%o *Permite que seja efetuada a escrita de números octais*

%s *Permite que seja efetuada a escrita de uma série de caracteres*

%u *Permite que seja efetuada a escrita de um número decimal sem sinal*

%x *Permite que seja efetuada a escrita de um número hexadecimal*

**CÓDIGOS ESPECIAIS**

\n *Cria uma linha nova a partir do ponto que é indicado*

\t *Cria um espaço de tabulação do ponto que é indicado*

\b  *Executa um retrocesso de espaço do ponto que é indicado*

\” *Apresenta o símbolo de aspas no ponto que é indicado*

\\ *Apresenta o símbolo de barra no ponto que é indicado*

\f *Adiciona um salto de página de formulário (impressora)*

\0 *Gera um nulo*

**COMANDO SE (if - then - else)**

**if (condição)**

{

**&&** e

**||** ou

Sequência de comandos A ;

}

**else**

{

Sequência de comandos B ;

}

**COMANDO SE (if else)** (variação quando há somente um comando em cada opção)

**If (condição)**

Comando se o primeiro if for verdadeiro ;

**else if**

Comando se o primeiro if for falso e o segundo verdadeiro ;

**else if**

Comando se o segundo if for falso e o terceiro verdadeiro ;

**else if**

Comando se o terceiro if for falso e o quarto verdadeiro ;

...

**else**

Comando se o último if for falso ;

**COMANDO FOR (variável de inicialização; condição; incremento)**

**for(i=1;i<=10;i++) // repete o laço 10 vezes com incremento de 1 em i**

**{**

**Comandos a serem repetidos;**

**}**

**COMANDO WHILE (Condição)**

**While (condição)**

**{**

**Comandos a serem repetidos;**

**}**

**COMANDO DO WHILE (Condição)**

**do**

**{**

**Comandos a serem repetidos**

**} while (condição) ;**

**COMANDO SWITCH(variável)**

switch (variável)

{

case constante1:

Instruções;

break;

case constante2:

Instruções;

break;

default

Instruções;

}

**ESTRUTURAS DE ACUMULAÇÃO**

i++ i = i + 1

i+= j i = i + j

<https://pt.wikipedia.org/wiki/Operadores_em_C_e_C%2B%2B>

# Operadores em C e C++

Origem: Wikipédia, a enciclopédia livre.

C++ também contém os operadores para conversão de tipos de dados const\_cast, static\_cast, dynamic\_cast e reinterpret\_cast, que não estão listados na tabela.

## Índice

* [1 Precedência de operadores](https://pt.wikipedia.org/wiki/Operadores_em_C_e_C%2B%2B#Preced.C3.AAncia_de_operadores)
* [2 Tabela](https://pt.wikipedia.org/wiki/Operadores_em_C_e_C%2B%2B#Tabela)
  + [2.1 Operadores aritméticos](https://pt.wikipedia.org/wiki/Operadores_em_C_e_C%2B%2B#Operadores_aritm.C3.A9ticos)
  + [2.2 Operadores comparativos](https://pt.wikipedia.org/wiki/Operadores_em_C_e_C%2B%2B#Operadores_comparativos)
  + [2.3 Operadores de deslocamento de bits](https://pt.wikipedia.org/wiki/Operadores_em_C_e_C%2B%2B#Operadores_de_deslocamento_de_bits)
  + [2.4 Operadores lógicos sobre bits](https://pt.wikipedia.org/wiki/Operadores_em_C_e_C%2B%2B#Operadores_l.C3.B3gicos_sobre_bits)
  + [2.5 Outros operadores](https://pt.wikipedia.org/wiki/Operadores_em_C_e_C%2B%2B#Outros_operadores)
* [3 Sinônimos em C++](https://pt.wikipedia.org/wiki/Operadores_em_C_e_C%2B%2B#Sin.C3.B4nimos_em_C.2B.2B)

## Precedência de operadores

A seguir é listada a ordem de precedência e associatividade dos operadores. Elementos na mesma linha são calculados com a mesma precedência, de acordo com a direção dada.

A sintaxe das expressões é especificada por uma [gramática livre de contexto](https://pt.wikipedia.org/wiki/Gram%C3%A1ticas_livres_de_contexto), a tabela a seguir é inferida pela gramática.

|  |  |  |
| --- | --- | --- |
| **Operadores** | **Descrição** | [**Associatividade**](https://pt.wikipedia.org/wiki/Associatividade) |
| :: | Resolução de escopo | esquerda para direita |
| ++ --  () [] . -> | Incremento e decremento pós-fixo Parênteses (chamada de função)  Elemento de arranjo Seleção de elemento por identificador Seleção de elemento por ponteiro | esquerda para direita |
| ++ --  + - ! ~ (*tipo*) \* & sizeof new [] delete [] | Incremento e decremento prefixo  Adição e subtração unária *Não* lógico e complemento Conversão de tipo de dado Desreferência Referência (endereço de elemento) tamanho de elemento Alocação dinâmica de memória Desalocação dinâmica de memória | direita para esquerda |
| .\* ->\* | Ponteiro para membro | esquerda para direita |
| \* / % | Multiplicação, divisão, e módulo (resto) |
| + - | Adição e subtração |
| << >> | Deslocamento de bits à esquerda e à direita |
| < <=  > >= | “menor que” e “menor ou igual que”  “maior que” e “maior ou igual que” |
| == != | “Igual a” e “diferente de ” |
| & | *E* para bits |
| ^ | *Ou exclusivo* para bits |
| | | *Ou* para bits |
| && | *E* lógico |
| || | *Ou* lógico |
| *c*?*t*:*f* | [Condição ternária](https://pt.wikipedia.org/wiki/Opera%C3%A7%C3%A3o_tern%C3%A1ria) | direita para esquerda |
| =  += -= \*= /= %= <<= >>= &= ^= |= | Atribuição  Atribuição por adição ou subtração Atribuição por multiplicação, divisão ou módulo (resto) Atribuição por deslocalmento de bits Atribuição por operações lógicas |
| throw | Lançamento de [exceção](https://pt.wikipedia.org/wiki/Exce%C3%A7%C3%A3o) | --- |
| , | Vírgula | esquerda para direita |

## Tabela

|  |
| --- |
| Operadores aritméticos |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **Operador** | **Sintaxe** | **É sobrecarregável?** | **Presente em** [**C**](https://pt.wikipedia.org/wiki/Linguagem_de_programa%C3%A7%C3%A3o_C)**?** |
| Adição unária | **+**a | Sim | Sim |
| Adição | a **+** b | Sim | Sim |
| Incremento pré-fixado | **++**a | Sim | Sim |
| Incremento pós-fixado | a**++** | Sim | Sim |
| Atribuição por adição | a **+=** b | Sim | Sim |
| Subtração unária | **-**a | Sim | Sim |
| Subtração | a **-** b | Sim | Sim |
| Decremento pré-fixado | **--**a | Sim | Sim |
| Decremento pós-fixado | a**--** | Sim | Sim |
| Atribuição por subtração | a **-=** b | Sim | Sim |
| Multiplicação | a **\*** b | Sim | Sim |
| Atribuição por multiplicação | a **\*=** b | Sim | Sim |
| Divisão | a **/** b | Sim | Sim |
| Atribuição por divisão | a **/=** b | Sim | Sim |
| Módulo (resto) | a **%** b | Sim | Sim |
| Atribuição por módulo (resto) | a **%=** b | Sim | Sim |
| Operadores comparativos | | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **Operador** | **Sintaxe** | **É sobrecarregável?** | **Presente em** [**C**](https://pt.wikipedia.org/wiki/Linguagem_de_programa%C3%A7%C3%A3o_C)**?** |
| Menor que | a **<** b | Sim | Sim |
| Menor ou igual que | a **<=** b | Sim | Sim |
| Maior que | a **>** b | Sim | Sim |
| Maior ou igual que | a **>=** b | Sim | Sim |
| Diferente de | a **!=** b | Sim | Sim |
| Igual a | a **==** b | Sim | Sim |
| *Não* lógico | **!**a | Sim | Sim |
| *E* lógico | a **&&** b | Sim | Sim |
| *Ou* lógico | a **||** b | Sim | Sim |
| confirm("lala") Operadores de deslocamento de bits | | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **Operador** | **Sintaxe** | **É sobrecarregável?** | **Presente em** [**C**](https://pt.wikipedia.org/wiki/Linguagem_de_programa%C3%A7%C3%A3o_C)**?** |
| Deslocamento à esquerda | a **<<** b | Sim | Sim |
| Atribução de deslocamento à esquerda | a **<<=** b | Sim | Sim |
| Deslocamento à direita | a **>>** b | Sim | Sim |
| Atribuição de deslocamento à direita | a **>>=** b | Sim | Sim |
| Operadores lógicos sobre bits | | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **Operador** | **Sintaxe** | **É sobrecarregável?** | **Presente em** [**C**](https://pt.wikipedia.org/wiki/Linguagem_de_programa%C3%A7%C3%A3o_C)**?** |
| Complemento | **~**a | Sim | Sim |
| *E* | a **&** b | Sim | Sim |
| Atribuição por *e* | a **&=** b | Sim | Sim |
| *Ou* | a **|** b | Sim | Sim |
| Atribuição por *ou* | a **|=** b | Sim | Sim |
| *Ou exclusivo* | a **^** b | Sim | Sim |
| Atribuição por *ou exclusivo* | a **^=** b | Sim | Sim |
| Outros operadores | | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **Operador** | **Sintaxe** | **É sobrecarregável?** | **Presente em** [**C**](https://pt.wikipedia.org/wiki/Linguagem_de_programa%C3%A7%C3%A3o_C)**?** |
| Atribuição | a **=** b | Sim | Sim |
| Chamada de função | a**()** | Sim | Sim |
| Elemento de arranjo | a**[]** | Sim | Sim |
| De referência | **\***a | Sim | Sim |
| Referência | **&**a | Sim | Sim |
| Membro de ponteiro | a**->**b | Sim | Sim |
| Membro de identificador | a**.**b | Não | Sim |
| De referência de membro de identificador | a**.\***b | Não | Não |
| De-referência de membro de ponteiro | a**->\***b | Sim | Não |
| Conversão de tipo de dados | (*tipo*) a | Sim | Sim |
| Vírgula | a **,** b | Sim | Sim |
| Condição ternária | a **?** b **:** c | Não | Sim |
| Resolução de escopo | a **::** b | Não | Não |
| Tamanho de | **sizeof** a | Não | Sim |
| Identificador de tipo | **typeid** ([*tipo de dado*](https://pt.wikipedia.org/wiki/Tipo_de_dado)) | Não | Não |

## Sinônimos em C++

C++ define [palavras-chave](https://pt.wikipedia.org/wiki/Palavra-chave) que atuam como apelidos para alguns operadores: and (and), bitand (and), and\_eq (and=), or (||), bitor (|), or\_eq (|=), xor (^), xor\_eq (^=), not (!), not\_eq (!=), compl (~). Eles são processados pelo [analisador sintático](https://pt.wikipedia.org/wiki/Analisador_sint%C3%A1tico) da mesma forma que seus equivalentes.

Já C fornece na biblioteca padrão o cabeçalho iso646.h, que define esses símbolos através de macros.

[Categorias](https://pt.wikipedia.org/wiki/Especial:Categorias):

* [Linguagem de programação C](https://pt.wikipedia.org/wiki/Categoria:Linguagem_de_programa%C3%A7%C3%A3o_C)
* [C++](https://pt.wikipedia.org/wiki/Categoria:C%2B%2B)