

[< VOLTAR](#)

Operadores aritméticos, relacionais e lógicos

Apresentar os operadores aritméticos, relacionais e lógicos utilizados no Português Estruturado e na linguagem C.

NESTE TÓPICO

- > Introdução
- > Operadores de atribuição
- > Operadores aritméticos
- > Operadores aritméticos e atribuição **Marcar tópico**
- > Operadores Relacionais



Introdução

Os operadores são utilizados para representar expressões de cálculo, comparação, condição e atribuição. São meios pelos quais incrementamos, decrementamos, comparamos e avaliamos dados dentro do computador. temos os seguintes tipos de operadores: de atribuição, aritméticos, relacionais e lógicos.

Operadores de atribuição

A atribuição é a operação que permite armazenar um valor em uma variável. Para essa operação, será utilizado o símbolo `<-` (seta para esquerda), e a variável que receberá esse valor ou resultado de uma expressão deverá estar do lado esquerdo da seta. Dessa forma, pode-se ler a expressão `X <- 15` como armazenar o valor 15 na variável X ou ainda X recebe 15. Ainda é possível escrever `X <- X + 1`, que significa X recebe o valor de `X + 1`. Nesse caso, ao se entrar com um valor para a variável X, será obtido como resultado o valor original acrescido de 1. Se X for 15, o resultado exibido será 16.

Na linguagem C, o símbolo `<-`, que é utilizado no Português Estruturado, deverá ser substituído pelo símbolo `=`. Utilizando o mesmo exemplo dado acima, temos a expressão `X <- X + 1` em Português Estruturado, que na linguagem C corresponde à expressão `X = X + 1`.



Operadores aritméticos

Os operadores aritméticos são responsáveis pelo estabelecimento das operações matemáticas a serem realizadas em um computador. Tanto variáveis como constantes são utilizadas na elaboração dos cálculos matemáticos. São divididos em duas categorias: unários e binários.

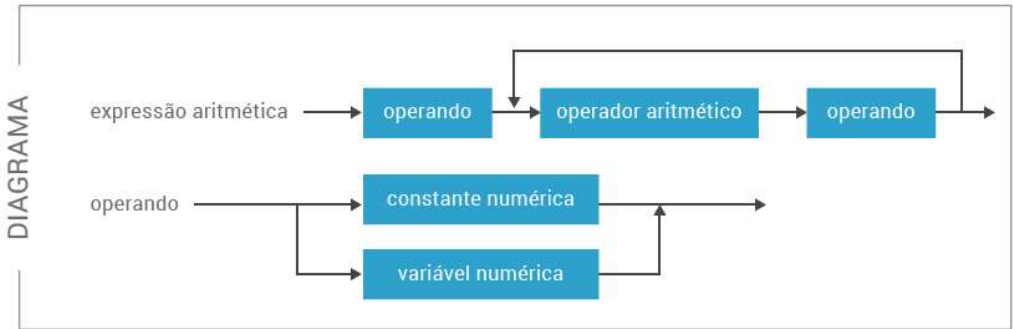
São unários os operadores que atuam na inversão do estado de um valor numérico. A operação de troca de sinal é unária, pois altera o sinal de um único operando. Por exemplo, se o valor da variável X é 10, a expressão $Y \leftarrow -X$, depois de executada, resulta em Y valendo -10.

Já os operadores binários são utilizados em operações matemáticas de divisão, multiplicação, adição e subtração. Em uma expressão aritmética, caso necessite alterar o nível de prioridade de um referido cálculo, ele deve ser definido por meio de parênteses.

Na Tabela a seguir são mostrados os operadores aritméticos, em Português Estruturado, mais utilizados.

Operador	Operação	Categoria	Resultado	Prioridade
+	Manutenção de sinal	Unário	-	1
-	Inversão de sinal	Unário	-	1
+	Adição	Binário	Inteiro ou Real	3
-	Subtração	Binário	Inteiro ou Real	3
*	Multiplicação	Binário	Inteiro ou Real	2
/	Divisão	Binário	Real	2
mod	Módulo (resto da divisão)	Binário	Inteiro	2

As expressões aritméticas são definidas pelo relacionamento existente entre variáveis e constantes numéricas com a utilização dos operadores aritméticos. Na Figura a seguir é possível observar um diagrama do esquema de uma expressão aritmética.



Esquema de uma expressão aritmética

Fonte: FORBELLONE, A. L. V.; EBERSPACHER, H. F. Lógica de Programação: a construção de algoritmos e estruturas de dados (3a. edição). São Paulo: Prentice Hall, 2005.

Como exemplo de expressão aritmética, a seguir é apresentada a expressão em Português estruturado e sua respectiva expressão em C.

```
1. /* Português Estruturado
2. X <- X + (Y * 2)
3.
4. /* Linguagem C
5. X = X + (Y * 2);
```

As expressões aritméticas, em computação, são escritas de uma forma um pouco diferente da forma conhecida em matemática. Observe os exemplos a seguir:

- Expressão matemática: $X = \{43 \cdot [55 : (30 + 2)]\}$
- Expressão Computacional: $X <- (43 * (55 / (30 + 2)))$

Na forma computacional, as chaves e colchetes são abolidos, utilizando-se em seu lugar apenas parênteses. É também substituído o sinal de (=) igual pelo sinal de (<-) implicado ou atribuído. O sinal implicado ou atribuído (<-) é utilizado para indicar que o valor de uma expressão aritmética está sendo armazenado em uma variável.

Um exemplo é fórmula para calcular a área de um triângulo, o qual é mostrado a seguir:

–Expressão matemática:

$$A = \frac{b \cdot h}{2}$$

–Expressão computacional:

$$A <- (b * h) / 2$$

A linguagem C oferece 6 operadores aritméticos binários (operam sobre dois operandos) e um operador aritmético unário (opera sobre um operando). Os operadores não se diferenciam dos mesmos operadores em Português Estruturado, exceto pelo operador *mod*, que na linguagem C utilizamos o símbolo %. Logo, os operadores aritméticos na linguagem C são os seguintes:



Operador	Descrição
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Módulo (resto da divisão inteira)

É adotada a mesma ordem de precedência (prioridades) dos operadores utilizados em Português Estruturado, sendo que para alterar a ordem de precedência, devemos utilizar parênteses.

No exemplo a seguir, o algoritmo (Português Estruturado e em C) calcula e mostra o resto da divisão de um número inteiro digitado pelo usuário.

```
1. algoritmo Calcula_Resto
2. var
3.     num, resto: inteiro
4. inicio
5.
6.     escreva "Digite um numero inteiro"
7.     leia num
8.     resto <- num MOD 2
9.     escreva "Resto da divisao: ", resto
10.
11. fim
```

```
1. #include <stdio.h>
2. #include <stdlib.h>
3.
4. main (){
5.
6.     int num, resto;
7.     printf ("\n Digite um numero inteiro:");
8.     scanf ("%d", &num);
9.     resto = num % 2;
10.    printf ("\n Resto da divisao: %d", resto);
11.    system ("PAUSE");
12.
13. }
```

Caso a variável lida pelo usuário seja 10, então, em ambos os casos, será exibida a seguinte mensagem: "Resto da divisao: 0".

Além das operações aritméticas tradicionais, na linguagem C existe o conceito de operadores de incremento (++) e decremento (--).

O operador de incremento (++) é responsável pelo incremento de uma unidade em seu operando. Este operador trabalha de dois modos:

1. O primeiro modo é chamado pré-fixado e o operador aparece antes do nome da variável. Como exemplo, temos a expressão `++n;`



2. O segundo é o modo pós-fixado em que o operador aparece seguindo o nome da variável. Como exemplo, temos a expressão $n++$;

Em ambos os casos, a variável é incrementada. Porém quando $++n$ é usado numa instrução, n é incrementada antes de seu valor ser usado, e quando $n++$ estiver numa instrução, n é incrementada depois de seu valor ser usado.

Os operadores de incremento e decremento tem precedência sobre todos os operadores aritméticos.

Como exemplo temos o seguinte: se as seguintes linhas de código a seguir forem executadas, o que será exibido?

```
1. n = 5;
2. x = n++;
3. printf ("x = %d n = %d", x, n);
```

A mensagem de saída será a seguinte: $x = 5$ $n = 6$.

Em outro exemplo temos o seguinte: se as seguintes linhas de código a seguir forem executadas, o que será exibido?

```
1. n = 5;
2. x = ++n;
3. printf ("x = %d n = %d", x, n);
```

A mensagem de saída será a seguinte: $x = 6$ $n = 6$.

Operadores aritméticos de atribuição

Muitas vezes, para simplificar uma expressão aritmética na linguagem C, utilizamos o conceito de operadores aritméticos de atribuição, que são os seguintes: $+=$, $-=$, $*=$, $/=$, $%=$. Cada um destes operadores é usado com um nome de variável a sua esquerda e uma expressão a sua direita. A operação consiste em atribuir um novo valor à variável que dependerá do operador e da expressão à direita.

Se x é uma variável, exp uma expressão e op um operador aritmético ($+$, $-$, $*$, $/$ ou $%$), então:

$x\ op=exp;$ equivale a $x = (x)\ op\ (exp);$

Na Tabela abaixo seguem alguns exemplos.

Expressão utilizando operador aritmético de atribuição	Expressão equivalente
$i += 2;$	$i = i + 2;$
$x *= y + 1;$	$x = x * (y + 1);$



Expressão utilizando operador aritmético de atribuição	Expressão equivalente
$t /= 2.5;$	$t = t / 2.5;$
$p \% = 5;$	$p = p \% 5;$
$d -= 3;$	$d = d - 3;$

Operadores Relacionais

Uma condição é o estabelecimento de uma relação lógica entre dois elementos (variável x variável ou variável x constante). Para definir uma relação lógica é necessário usar operadores relacionais. Uma relação lógica (comparação) retorna somente os valores *verdadeiro* ou *falso*. Os operadores relacionais possuem o mesmo nível de prioridade.

Na Tabela a seguir são mostrados os operadores relacionais em Português Estruturado e na linguagem C.

Operador relacional (Português Estruturado)	Operador relacional (linguagem C)	Função
=	==	Igual a
>	>	Maior que
<	<	Menor que
>=	>=	Maior ou igual a
<=	<=	Menor ou igual a
<>	!=	Diferente de

Por exemplo, analisando a relação numérica $A + B = C$, o resultado será verdade ou falsidade à medida que o valor da expressão aritmética $A + B$ seja igual ou diferente do conteúdo da variável C , respectivamente.

Observe os seguintes exemplos:

- $2 * 4 = 24 / 3 \quad 8 = 8$ (Verdadeiro)
- $15 \text{ MOD } 4 < 19 \text{ MOD } 6 \quad 3 < 1$ (Falso)
- $2 + 8 \text{ MOD } 7 \geq 3 * 6 - 15 \quad 2 + 1 \geq 18 - 15 \quad 3 \geq 3$ (Verdadeiro)



Operadores Lógicos

São utilizados para uma mesma tomada de decisão que necessite de mais de uma condição. Também são conhecidos como operadores booleanos. Utilizaremos três operadores básicos e eles estão descritos na Tabela a seguir, em Português Estruturado e na linguagem C.

Operador lógico (Português Estruturado)	Operador Lógico (linguagem C)	Função	Prioridade
.e.	&&	Conjunção	2
.ou.		Disjunção	3
.não.	!	Negação	1

Considere a expressão: $A = B < 8$.e. $C = 3$. Essa expressão combina duas expressões de comparação: $B < 8$ e $C = 3$. Se ambas resultarem em verdadeiro, o valor verdadeiro será atribuído à variável A. Em qualquer outra circunstância, o valor falso será atribuído a A. Devemos observar que, primeiramente, são avaliadas as expressões de comparação $B < 8$ e $C = 3$ e, posteriormente, os resultados dessas duas expressões são associados por meio do operador .e., obtendo-se o resultado final. Isso se deve ao fato de existir uma precedência entre os operadores relacionais e os lógicos. Os operadores relacionais são avaliados primeiro e, em seguida, os lógicos.

A tabela-verdade expressa o conjunto de possibilidades existentes para a combinação de variáveis ou expressões e operadores lógicos. Um exemplo de combinação entre variáveis é $A \geq 5$.e. $B \neq 10$, onde A e B são as variáveis, \geq e \neq são os operadores relacionais e .e. é o operador lógico. Desse modo, a tabela-verdade é utilizada para facilitar a análise da combinação dessas expressões ou variáveis, conforme pode ser observado na Tabela a seguir. Em resultados possíveis, o valor refere-se ao valor lógico verdadeiro e .f. ao valor lógico falso. Os resultados referentes à operação de negação são baseados na primeira coluna dos resultados possíveis.

Resultados possíveis		&& (.e.)	(.ou.)	! (.não.)
V	V	V	V	F
V	F	F	V	F
F	V	F	V	V
F	F	F	F	V



Pela observação da tabela-verdade, se $A = 10$ e $B = 15$, então a expressão $A \geq 5$.e. $B \neq 10$ resulta em V (verdadeiro).

Observe os seguintes exemplos, sendo os 2 primeiros em Português Estruturado e os outros 2 em linguagem C.

1. $5 < 3$.e. $10/2=5$ F .e. V resulta em F.
2. $5 < 3$.ou. $10/2=5$ F .ou. V resulta em V.
3. $!(5 < 3) \ \&\& \ 10/2 \neq 5$ V $\&\&$ F resulta em F.
4. $!(5 < 3) \ || \ 10/2 \neq 5$ V $||$ F resulta em V.

Quiz

Exercício Final

Operadores aritméticos, relacionais e lógicos

INICIAR ➤

Referências

FORBELLONE, A. L. V; EBERSPACHER, H. F. Lógica de Programação: a construção de algoritmos e estruturas de dados (3a. edição). São Paulo: Prentice Hall, 2005.

RISSETTI, G.; PUGA, S. Lógica de Programação e estruturas de dados, com aplicações em Java. São Paulo: Prentice Hall, 2003.

SOUZA, M. A. F. e outros. Algoritmos e lógica de programação: um texto introdutório para Engenharia (2a. edição). São Paulo: Cengage Learning, 2011.



Avalie este tópico



ANTERIOR

Introdução à linguagem C



Índice

Biblioteca
(https://www.uninove.br/conheca-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/) Portal Uninove
(http://www.uninove.br) Mapa do Site

Ajuda?
PRÓXIMO
(https://ava.uninove.br/ava/Desvio condicional simples)

Desvio condicional simples

© Todos os direitos reservados

