
20
anos
uniAvan

UNIAVAN - Centro Universitário Avantis
Curso: Sistemas de Informação
Disciplina: Algoritmos I

Introdução aos Algoritmos

Prof. Luiz Fernando M. Arruda, Me. Eng.

20 anos

Sumário

Operadores

Os operadores são ferramentas fundamentais para a programação de computadores. Eles geralmente são divididos em:

- Aritméticos: utilizados para efetuar operações matemáticas
- Relacionais: para comparar valores
- Lógicos: para combinar expressões booleanas
- Atribuição: atribuir valores
- Incremento e Decremento: utilizados para incrementar ou decrementar variáveis
- Bitwise para operações binárias: deslocamento de bit

Curiosidade

Existem outros operadores em programação que são exclusivos de cada linguagem. Por exemplo em java temos o operador **instanceof** para verificar a existência de um objeto é uma instância de uma classe especifica.

Operadores Aritméticos

Os operadores aritméticos são símbolos usados na matemática para representar as operações básicas de adição, subtração, multiplicação e divisão. Esses operadores são amplamente usados em linguagens de programação para realizar cálculos numéricos.
E segundo Deitel e Deitel (2011), grande parte maioria dos programas em C executa cálculos aritméticos.

- Adição ("+")
- Subtração ("−")
- Multiplicação ("∗")
- Divisão ("/")
- Potência ("∧")
- Módulo ou Resto da Divisão ("%")

Atenção

Para efetuar um conjunto de operações em que há uma sequência ou prioridade a ser respeitada, utiliza-se os '( )' parenteses, tal como estudado em matemática durante o ensino médio.

Operadores Aritméticos

Operação em C	Operador aritmético	Expressão algébrica	Expressão em C
Adição	+	$f + 7$	f + 7
Subtração	–	$p - c$	p – c
Multiplicação	*	bm	b * m
Divisão	/	x / y ou $\frac{x}{y}$ ou $x \div y$	x / y
Módulo ou resto da divisão entre 2 inteiros	%	$r \bmod s$	r % s

Figura: Operadores Aritméticos - (DEITEL; DEITEL, 2011, p. 29)

Operadores Aritméticos

Em C os operadores nas expressões aritméticas seguem uma sequência determinada pela regra de precedência de operadores, que geralmente são iguais às regras da álgebra:

- Primeiro efetua-se as operadores contidas dentro de pares de parênteses.
- Depois as operações de multiplicação, divisão e módulo.
- Por fim as operações de adição e de subtração.

Atenção

Sempre que houver mais de uma operação, seja ela com multiplicação, divisão ou qualquer outra, efetua-se a operação que surgir primeiro da esquerda para direita.

Operadores Aritméticos

Operador(es)	Operação(ões)	Ordem de avaliação (precedência)
( )	Parênteses	Avaliados em primeiro lugar. Se os parênteses forem aninhados, a expressão no par mais interno é a primeira a ser avaliada. Se houver vários pares de parênteses 'no mesmo nível' (ou seja, não aninhados), eles serão avaliados da esquerda para a direita.
* / %	Multiplicação Divisão Módulo	Avaliados em segundo lugar. Se houver vários, serão avaliados da esquerda para a direita.
+ –	Adição Subtração	Avaliados por último. Se houver vários, serão avaliados da esquerda para a direita.

Figura: Precedência dos Operadores Aritméticos - (DEITEL; DEITEL, 2011, p. 30)

Operadores Aritméticos

Etapa 1. `y = 2 * 5 * 5 + 3 * 5 + 7;` *(Multiplicação da esquerda)*
`2 * 5 é 10`

Etapa 2. `y = 10 * 5 + 3 * 5 + 7;` *(Multiplicação da esquerda)*
`10 * 5 é 50`

Etapa 3. `y = 50 + 3 * 5 + 7;` *(Multiplicação antes da adição)*
`3 * 5 é 15`

Etapa 4. `y = 50 + 15 + 7;` *(Adição da esquerda)*
`50 + 15 é 65`

Etapa 5. `y = 65 + 7;` *(Última adição)*
`65 + 7 é 72`

Etapa 6. `y = 72` *(Última operação coloca 72 em y)*

Figura: Exemplo de Operadores Aritméticos - (DEITEL; DEITEL, 2011, p. 31)

Operadores Relacionais

Em programação, são realizadas ações como cálculo quanto tomada de decisão, e para que se possa tomar uma decisão são utilizados os operadores relacionais (DEITEL; DEITEL, 2011). Assim, eles são utilizados para comparar valores e verificar se eles são iguais, diferentes, maiores, menores, maiores ou iguais e menores ou iguais.
Exemplos de operadores:

- Igual ("==")
- Diferente ("!=")
- Maior que (">")
- Maior ou igual (">=")
- Menor que ("<")
- Menor ou igual ("<=")

Atenção

Embora existente em todas as linguagem de programação, cada uma delas pode possuir uma sintaxe diferente.

Operadores Relacionais

Operador de igualdade ou relacional na álgebra	Operador de igualdade ou relacional em C	Exemplo de condição em C	Significado da condição em C
Operadores de igualdade			
=	==	x == y	x é igual a y
≠	!=	x != y	x não é igual a y
Operadores relacionais			
>	>	x > y	x é maior que y
<	<	x < y	x é menor que y
≥	>=	x >= y	x é maior ou igual a y
≤	<=	x <= y	x é menor ou igual a y

Figura: Operadores de igualdade e relacionais (DEITEL; DEITEL, 2011, p. 32)

Operadores Relacionais

Alguns erros comuns apontados por (DEITEL; DEITEL, 2011)

Atenção

Ocorrerá um erro de sintaxe se os dois símbolos em um dos operadores $==$, $!=$, $>=$ e $<=$ forem separados por espaços.

Atenção

Ocorrerá um erro de sintaxe se os dois símbolos em um dos operadores $!=$, $>=$ e $<=$ forem invertidos, como em $=!$, $=>$ e $=<$, respectivamente.

Operadores Lógicos

Os operadores lógicos são utilizados em operações mais complexas ao combinar condições simples (DEITEL; DEITEL, 2011). Desta forma em programação, os operadores lógicos são utilizados para combinar ou negar expressões booleanas.
Exemplos de operadores:

- AND lógico ("&&") onde retorna verdadeiro se e somente se as duas expressões que o cercam forem verdadeiras.
- OR lógico ("||") onde retorna verdadeiro se pelo menos uma das duas expressões que o cercam for verdadeira.
- NOT lógico ("!"), retorna o oposto da expressão que o segue

Atenção

Assim como os demais operadores, os operadores lógicos existem em outras linguagem mas podem surgir mudanças de sintaxes.

Operadores Lógicos

A	~A
F	V
V	F

Not ➔ ~ , !
Ou ➔ or, ||
E ➔ and &&

A	B	A e B	A ou B
F	F	F	F
F	V	F	V
V	F	F	V
V	V	V	V

Figura: Exemplo de Operadores Lógicos

Atenção

Sempre que houver mais de uma operação, efetua-se a operação NOT em seguida AND e por ultimo OR.

Operadores de Atribuição

Em geral, o operador de atribuição é usado para armazenar valores em variáveis, que podem ser posteriormente usadas em cálculos ou outros tipos de operações.

Operador de atribuição	Exemplo de expressão	Explicação	Atribui
Considere: **int** c = 3, d = 5, e = 4, f = 6, g = 12;			
+=	c += 7	c = c + 7	10 a c
-=	d -= 4	d = d - 4	1 a d
*=	e *= 5	e = e * 5	20 a e
/=	f /= 3	f = f / 3	2 a f
%=	g %= 9	g = g % 9	3 a g

Figura: (DEITEL; DEITEL, 2011, p. 65)

Operadores de Incremento e Decremento

O operador de incremento $++$ e o operador decremento $--$, geralmente são utilizados em laços de repetição onde se buscar percorrer um vetor ou uma matriz, deslocando-se uma casa a frente o atrás. Há exemplo se uma variável c é incrementada em 1, o operador de incremento $++$ pode ser usado no lugar das expressões $c = c + 1$ ou $c+ = 1$.

Operador	Exemplo de expressão	Explicação
++	++a	Incrementa a em 1, e então usa o novo valor de a na expressão em que a estiver.
++	a++	Usa o valor corrente de a na expressão em que a estiver, e então incrementa a em 1.
--	--b	Decrementa b em 1, e então usa o novo valor de b na expressão em que b estiver.
--	b--	Usa o valor corrente de b na expressão em que b estiver, e então decrementa b em 1.

Figura: (DEITEL; DEITEL, 2011, p. 65)

Operadores de Bitwise

Os operadores bitwise são usados em programação para manipular bits individuais em variáveis numéricas. Esses operadores permitem que você realize operações lógicas com valores binários, em vez de valores decimais. Eles são comumente usados em sistemas embarcados, redes e programação de baixo nível.

- AND bitwise ("$\&$"): retorna 1 no bit resultante se ambos os bits comparados forem 1.
- Or bitwise ("$|$"): retorna 1 no bit resultante se pelo menos um dos bits comparados for 1.
- XOR bitwise ("$\wedge$"): retorna 1 no bit resultante se apenas um dos bits comparados for 1.
- Deslocamento para a esquerda ("$<<$"): desloca os bits de uma variável para a esquerda. Por exemplo: $0001 << 2 = 0100$.
- Deslocamento para a direita ("$>>$"): desloca os bits de uma variável para a direita. Por exemplo: $0100 >> 2 = 0001$.

Vamos Praticar!?

Referencial Bibliográfico I

DEITEL, Harvey M; DEITEL, Paul J. **Como programar em C - 6ª Edição**. São Paulo: Pearson Prentice Hall, 2011.