

Programação em C

Operadores matemáticos e lógicos

Agostinho Brito

2021

O que são operadores

Operadores são símbolos que informam ao compilador que operação lógica ou matemática se deseja realizar. Eles podem ser:

- Operadores aritméticos;
- Operadores relacionais;
- Operadores lógicos;
- Operadores bit-a-bit;
- Operadores de atribuição;
- Outros operadores.

Operadores aritméticos

- São usados para realizar operações matemáticas entre valores.

Operador	Descrição	Exemplo
+	Adiciona duas quantidades	A + B
-	Subtrai o primeiro do segundo valor	A - B
*	Multiplica duas quantidades	A * B
/	Divide o primeiro valor pelo segundo	A / B
%	Resto da divisão do primeiro valor pelo segundo	A % B
++	Incrementa o valor inteiro de uma unidade	A++, ++A
--	Decrementa o valor inteiro de uma unidade	A--, --A

Operadores aritméticos

- São usados para realizar operações matemáticas entre valores.

Operador	Descrição	Exemplo
+	Adiciona duas quantidades	A + B
-	Subtrai o primeiro do segundo valor	A - B
*	Multiplica duas quantidades	A * B
/	Divide o primeiro valor pelo segundo	A / B
%	Resto da divisão do primeiro valor pelo segundo	A % B
++	Incrementa o valor inteiro de uma unidade	A++, ++A
--	Decrementa o valor inteiro de uma unidade	A--, --A



Praticando o uso de operadores...

Operadores Relacionais

- São usados para realizar comparações entre valores. Caso o resultado do teste seja afirmativo, a expressão retorna verdadeiro, caso contrário retorna falso.

Operador	Descrição	Exemplo
==	Verifica se os dois operandos são iguais	A == B
!=	Verifica se os dois operandos são diferentes	A != B
>	Verifica se o primeiro operando é maior que o segundo	A > B
<	Verifica se o primeiro operando é menor que o segundo	A < B
>=	Verifica se o primeiro operando é maior ou igual ao segundo	A >= B
<=	Verifica se o primeiro operando é menor ou igual ao segundo	A <= B

- São usados para realizar comparações entre valores. Caso o resultado do teste seja afirmativo, a expressão retorna verdadeiro, caso contrário retorna falso.

Operador	Descrição	Exemplo
==	Verifica se os dois operandos são iguais	A == B
!=	Verifica se os dois operandos são diferentes	A != B
>	Verifica se o primeiro operando é maior que o segundo	A > B
<	Verifica se o primeiro operando é menor que o segundo	A < B
>=	Verifica se o primeiro operando é maior ou igual ao segundo	A >= B
<=	Verifica se o primeiro operando é menor ou igual ao segundo	A <= B



Praticando o uso de operadores...

Operadores Lógicos

- São usados para realizar operações lógicas entre valores.
- Normalmente participam para compor testes condicionais em estruturas de tomada de decisão.

Operador	Descrição	Exemplo
& &	Operador lógico AND. Se ambos os operandos forem diferentes de ZERO, o resultado é verdadeiro	A & & B
	Operador lógico OU. Se um dos operandos for diferente de ZERO, o resultado é verdadeiro	A B
!	Operador lógico NOT. Se a condição do operando for VERDADEIRO, ele retorna falso. Se for FALSO, retorna VERDADEIRO	!A

- São usados para realizar operações lógicas entre valores.
- Normalmente participam para compor testes condicionais em estruturas de tomada de decisão.

Operador	Descrição	Exemplo
& &	Operador lógico AND. Se ambos os operandos forem diferentes de ZERO, o resultado é verdadeiro	A & & B
	Operador lógico OU. Se um dos operandos for diferente de ZERO, o resultado é verdadeiro	A B
!	Operador lógico NOT. Se a condição do operando for VERDADEIRO, ele retorna falso. Se for FALSO, retorna VERDADEIRO	!A



Praticando o uso de operadores...

- Os operadores bit-a-bit realizam operações lógicas com os bits dos números inteiros que participam da operação.
- É importante que se preste bastante atenção aos símbolos, pois podem ser confundidos com aqueles utilizados para os operadores lógicos.
- São três os símbolos $\&$, $|$, \wedge que equivalem às operações bit-a-bit de AND, OU e XOR (OU EXCLUSIVO), conforme a tabela seguinte:

p	q	p&q	p q	p ^ q
0	0	0	0	0
0	1	0	1	1
1	1	1	1	0
1	0	0	1	1

- Outros símbolos como \sim , $<<$ e $>>$ também realizam operações, fazendo complemento de um, e deslocamentos à esquerda e à direita do número inteiro, respectivamente.

Operador	Descrição	Exemplo
&	AND binário	A & B
	OR binário	A B
^	XOR binário	A ^ B
~	Complemento de 1	~A
<<	Deslocamento à esquerda	A << B
>>	Deslocamento à direita	A >> B

- Assumindo, por exemplo, que duas variáveis `unsigned char` tenham valores numéricos $A = 115$ e $B = 37$, como ficam os resultados dos operadores bit-a-bit?

Operador	Descrição	Exemplo
&	AND binário	A & B
	OR binário	A B
^	XOR binário	A ^ B
~	Complemento de 1	~A
<<	Deslocamento à esquerda	A << B
>>	Deslocamento à direita	A >> B

- Assumindo, por exemplo, que duas variáveis `unsigned char` tenham valores numéricos `A = 115` e `B = 37`, como ficam os resultados dos operadores bit-a-bit?



Praticando o uso de operadores...

- Diversos operadores existem em C para acelerar a tarefa de codificar a atribuição de valores.

Operador	Descrição	Exemplo
=	Atribuição simples	A = B
+=	Adiciona e atribui	A += B
-=	Subtrai e atribui	A -= B
*=	Multiplica e atribui	A *= B
/=	Divide e atribui	A /= B
%=	Calcula o resto e atribui	A %= B
<<=	Desloca à esquerda e atribui	A <<= B
>>=	Desloca à direita e atribui	A >>= B
&=	Calcula AND bit-a-bit e atribui	A &= B
=	Calcula OU bit-a-bit e atribui	A = B
^=	Calcula XOR bit-a-bit e atribui	A ^= B

- Diversos operadores existem em C para acelerar a tarefa de codificar a atribuição de valores.

Operador	Descrição	Exemplo
=	Atribuição simples	A = B
+=	Adiciona e atribui	A += B
-=	Subtrai e atribui	A -= B
*=	Multiplica e atribui	A *= B
/=	Divide e atribui	A /= B
%=	Calcula o resto e atribui	A %= B
<<=	Desloca à esquerda e atribui	A <<= B
>>=	Desloca à direita e atribui	A >>= B
&=	Calcula AND bit-a-bit e atribui	A &= B
=	Calcula OU bit-a-bit e atribui	A = B
^=	Calcula XOR bit-a-bit e atribui	A ^= B



Praticando o uso de operadores...

Outros operadores

Operador	Descrição	Exemplo
<code>sizeof()</code>	Retorna o tamanho de uma variável (em bytes)	<code>sizeof(A)</code>
<code>&</code>	Retorna o endereço de uma variável	<code>&A</code>
<code>*</code>	Retorna o conteúdo de um endereço	<code>*A</code>
<code>?:</code>	Expressão condicional	<code>(A>B) ? X : Y</code>

Operador	Descrição	Exemplo
<code>sizeof()</code>	Retorna o tamanho de uma variável (em bytes)	<code>sizeof(A)</code>
<code>&</code>	Retorna o endereço de uma variável	<code>&A</code>
<code>*</code>	Retorna o conteúdo de um endereço	<code>*A</code>
<code>?:</code>	Expressão condicional	<code>(A>B) ? X : Y</code>



Praticando o uso de operadores...

Precedência de operadores

Tipo	Operador	Associatividade
Pós-fixado	() [] -> . ++ --	Esquerda para a direita
Unário	+ - ! ~ ++ -- (type) * & sizeof	Direita para a esquerda
Multiplicativo	* / %	Esquerda para a direita
Aditivos	+ -	Esquerda para a direita
Deslocamento	« »	Esquerda para a direita
Relacionais	< <= > >=	Esquerda para a direita
Igualdade	== !=	Esquerda para a direita
AND bit-a-bit	&	Esquerda para a direita
XOR bit-a-bit	^	Esquerda para a direita
OR bit-a-bit		Esquerda para a direita
AND lógico	&&	Esquerda para a direita
OR lógico		Esquerda para a direita
Condicionais	?:	Direita para a esquerda
Atribuição	= += -= *= /= %= >>= <<= &= ^= =	Direita para a esquerda
Vírgula	,	Esquerda para a direita

Precedência de cima para baixo

Precedência de operadores

- A precedência dos operadores é importante para determinar como as partes de uma expressão são avaliadas para montar o resultado final.

```
x = 4 + 5 * 3; // x=19
```

```
x = 4 % 2 + 2; // x=2
```

```
x = 4 | 2 + 2; // x=4
```

- Como conseguir mudar a ordem das operações? Com o uso de parêntesis .
- É importante prestar atenção à ordem de precedência, pois muitos operadores são usados raramente, mas podem eventualmente figurar em alguma expressão.

- A precedência dos operadores é importante para determinar como as partes de uma expressão são avaliadas para montar o resultado final.

```
x = 4 + 5 * 3; // x=19
```

```
x = 4 % 2 + 2; // x=2
```

```
x = 4 | 2 + 2; // x=4
```

- Como conseguir mudar a ordem das operações? Com o uso de parêntesis .
- É importante prestar atenção à ordem de precedência, pois muitos operadores são usados raramente, mas podem eventualmente figurar em alguma expressão.



Praticando precedência...



Obrigado