

Expresiones Aritméticas

Andrés Arcia
Departamento de Computación
Escuela de Ingeniería de Sistemas
Facultad de Ingeniería
Universidad de Los Andes

La Operación Asignación

variable = expresión;

La sentencia de asignación es un modo de darle valores a una variable.

La expresión se evalúa y el resultado es asignado a la variable.

La sentencia de asignación modifica el contenido de una variable.

variable = valor constante;
variable = variable;
variable = expresión;

x = 3;
x = y;
x = (y + 1)/2;

La Operación Asignación

Ejemplos

$a = b;$

$\text{suma} = 60;$

¿Cuál será el valor que tomará la variable D tras la ejecución de las siguientes instrucciones ?

$A = 12;$

$B = A + 5;$

$C = B - 2;$

$D = C + (A + B);$



Expresiones

Puede ser:

Una variable

Una constante

Una combinación de operadores y operandos.



Expresiones

Operandos: constantes, variables u otras expresiones.

Operadores: Símbolo que indica al compilador que ejecute determinadas operaciones con los elementos sobre los que opera (aritméticos, lógicos, de cadena y relacionales).

Expresiones

Tipos de expresiones

Aritméticas: Los operandos que intervienen en ella son numéricos, el resultado es numérico y los operadores son aritméticos.

Lógicas: Su resultado es **CIERTO** ó **FALSO**. Se construyen mediante los operadores relacionales y lógicos.

Operadores Aritméticos

Operación	Operador	Expresión algebraica	Operador en C++	Expresión en C++
Suma	+	$A + 3$	+	$A + 3$
Resta	-	$B - C$	-	$B - C$
Multiplicación	x	$X \times Y$	*	$X * Y$
División	/	X/Y ó $\frac{X}{Y}$	/	X/Y
Resto de la División			%	$R \% S$
Incremento		$I + 1$	++	$I++$ ó $++I$
Decremento		$I - 1$	--	$I--$ ó $--I$

Operadores Aritméticos

Incremento (++)

Para $n = 2$, la ejecución de la sentencia

```
i = n++;           // post-incremento
```

asigna el valor 2 a i y después incrementa el valor de n a 3.

```
i = 2
```

```
n = 3
```

La ejecución de la sentencia

```
i = ++n;          // pre-incremento
```

incrementa el valor de n a 4 y después asigna este valor a i .

```
n = 4
```

```
i = 4
```


Operadores Aritméticos

Decremento (--)

Para $n = 5$, la ejecución de la sentencia

$i = n--;$ // post-decremento

asigna el valor 5 a i y después decrementa el valor de n a 4.

$i = 5$

$n = 4$

La ejecución de la sentencia

$i = --n;$ // pre-decremento

decrementa el valor de n a 3 y después asigna este valor a i .

$n = 3$

$i = 3$

Operadores Aritméticos

Incremento/ decremento

- Siempre que pueda, no utilice ++a; o --a; aunque son operaciones válidas.
- Puede utilizar incrementos/ decrementos en una expresión, por ejemplo
x = 5 + y++;
Equivalente a
x = 5 + y;
y++;
- Se utilizan sólo con variables enteras.

Operadores Aritméticos

- Los operadores +, -, *, /, se pueden usar con operandos (constante, variable o expresión) tanto enteros como reales.

- La división de enteros da como resultado un entero.

$$17/5 = 3$$

- El operador *módulo* (%) sólo puede ser utilizado con operandos enteros. La expresión $a\%b$ da como resultado el resto de la división entera de dos números a y b.

$$17\%5 = 2$$

Operadores Aritméticos

Ejemplos

$a + b * c$

$(\text{votos/electores}) * 100$

$a * x * x + b * x + c$

$(\text{alto} - \text{bajo}) / 2$

$\text{horas} * 3600 + \text{minutos} * 60 + \text{segundos}$

$\text{numero} \% 2$

Operadores Aritméticos

Los operandos de una expresión aritmética deben satisfacer las siguientes reglas:

- ☆ Si ambos operandos son de tipo entero entonces el resultado es de tipo entero.
- ☆ Si uno de los operandos o ambos operandos son de tipo real entonces el resultado es de tipo real.
- ☆ Si la expresión evaluada da como resultado un valor de tipo entero y la variable a la cual se le va a asignar este valor es de tipo real, entonces el resultado será convertido automáticamente al tipo real antes de realizar la asignación.
- ☆ Si la expresión evaluada da como resultado un valor de tipo real y la variable a la cual se le va a asignar este valor es de tipo entero, entonces el resultado será convertido automáticamente al tipo entero antes de realizar la asignación.
- ☆ En general los tipos de datos son promovidos, desde un tipo mas angosto hacia uno más amplio. Ej: de int a float.
- ☆ Cuando hay conversiones de datos más grandes a datos más pequeños, ocurre un truncamiento de los bits mas significativos

Operadores Aritméticos

- Si alguno de los dos operadores es long double, convierta el otro a long double.
- Si no, si cualquier operador es double, convierta el otro a double.
- Si no, si cualquier operador es float, convierta el otro a float.
- Si no, convierta char y short a int.
- Luego, si algun operador el long, convierta el otro a long.

Precedencia de los Operadores Aritméticos en

() Subexpresiones encerradas entre paréntesis se evalúan primero (mayor nivel de precedencia)

++ --

* / %

+ - (menor nivel de precedencia)

La evaluación de los operadores con la misma prioridad se realiza de izquierda a derecha.

Expresiones Aritméticas: Ejemplo 1

Evaluar la siguiente expresión aritmética:

$$(a + b + c + d + e) \% 5$$

$$R1 = a + b$$

$$R2 = R1 + c$$

$$R3 = R2 + d$$

$$R4 = R3 + e$$

$$R5 = R4 \% 5 \text{ (RESULTADO)}$$

Expresiones Aritméticas: Ejemplo 1

Evaluar la expresión aritmética

$$(a + b + c + d + e) \% 5$$

para los valores:

$$a = 3, b = 2, c = 1, d = 8, e = 4$$

$$R1 = 3 + 2 = 5$$

$$R2 = 5 + 1 = 6$$

$$R3 = 6 + 8 = 14$$

$$R4 = 14 + 4 = 18$$

$$R5 = 18 \% 5 = 3$$

Expresiones Aritméticas:

Ejemplo 2

Evaluar la siguiente expresión aritmética:

$$a \% 2 / b - c * a * y + 18 / c - 3 * h$$

$$R1 = a \% 2$$

$$R2 = R1 / b$$

$$R3 = c * a$$

$$R4 = R3 * y$$

$$R5 = 18 / c$$

$$R6 = 3 * h$$

$$R7 = R2 - R4$$

$$R8 = R7 + R5$$

$$R9 = R8 - R6 \text{ (Resultado)}$$

Expresiones Aritméticas: Ejemplo 2

Evaluar la expresión aritmética

$$a \% 2 / b - c * a * y + 18 / c - 3 * h$$

para los valores:

$$a = 3, b = 2, c = 1, h = 8, y = 4$$

$$R1 = 3 \% 2 = 1$$

$$R2 = 1 / 2 = 0$$

$$R3 = 1 * 3 = 3$$

$$R4 = 3 * 4 = 12$$

$$R5 = 18 / 1 = 18$$

$$R6 = 3 * 8 = 24$$

$$R7 = 0 - 12 = -12$$

$$R8 = -12 + 18 = 6$$

$$R9 = 6 - 24 = -18$$

Expresiones Aritméticas:

Ejemplo 3

Evaluar la siguiente expresión aritmética:

$$a * (b + c) + c * (d + e)$$

$$R1 = b + c$$

$$R2 = d + e$$

$$R3 = a * R1$$

$$R4 = c * R2$$

$$R5 = R3 + R4 \text{ (Resultado)}$$

Expresiones Aritméticas:

Ejemplo 3

Evaluar la siguiente expresión aritmética

$$a * (b + c) + c * (d + e)$$

para los valores:

$$a = 3, b = 2, c = 1, d = 8, e = 4$$

$$R1 = 2 + 1 = 3$$

$$R2 = 8 + 4 = 12$$

$$R3 = 3 * 3 = 9$$

$$R4 = 1 * 12 = 12$$

$$R5 = 9 + 12 = 21$$

Expresiones Aritméticas:

Ejemplo 4

Evaluar la siguiente expresión aritmética:

$$(a * (b + c)) - 2 * a + (4 * d - f)$$

$$R1 = b + c$$

$$R2 = a * R1$$

$$R3 = 4 * d$$

$$R4 = R3 - f$$

$$R5 = 2 * a$$

$$R6 = R2 - R5$$

$$R7 = R6 + R4 \text{ (Resultado)}$$

Expresiones Aritméticas: Ejemplo 4

Evaluar la expresión aritmética

$$(a * (b + c)) - 2 * a + (4 * d - f)$$

para los valores:

$$a = 3, b = 2, c = 1, d = 8, f = 4$$

$$R1 = 2 + 1 = 3$$

$$R2 = 3 * 3 = 9$$

$$R3 = 4 * 8 = 32$$

$$R4 = 32 - 4 = 28$$

$$R5 = 2 * 3 = 6$$

$$R6 = 9 - 6 = 3$$

$$R7 = 3 + 28 = 31$$

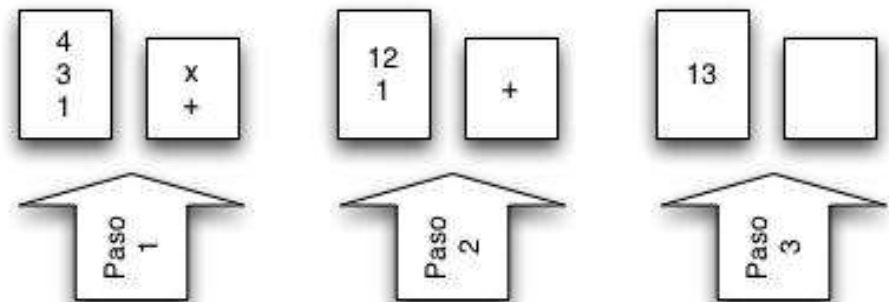
Evaluación de Expresiones Aritmeticas

Las expresiones aritmeticas pueden ser evaluadas de forma estrictamente secuencial:

Por ejemplo, la expresion:

$1+3*4$ se evalua poniendo los operandos en de un lado y los operadores de otro lado. Esta separación ocurre de forma secuencial.

Para la expresión anterior sería:

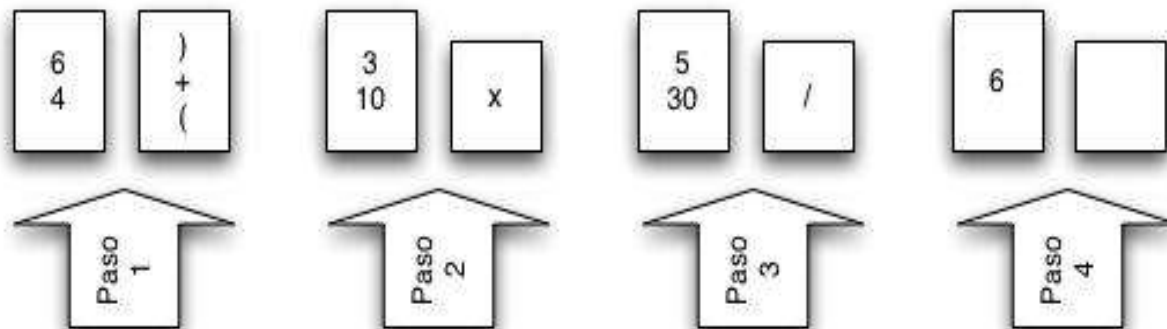


Evaluación de Expresiones Aritmeticas

Para evaluar una expresión debe tenerse en cuenta que:

- Los operadores tienen prioridad. De mayor a menor son: \times / $+$ -
- Las operaciones con operadores de igual prioridad se evalúan de izquierda a derecha.
- Al final de la operación se efectúan el resto de operaciones pendientes hasta que el recipiente de operadores quede vacío.

Ejemplo: $(4+6)*3/5$



Evaluación de Expresiones Aritmeticas

En el ejemplo anterior se utilizaron parentesis, que ocupan la mayor prioridad en la evaluación de una expresión aritmetica.

Cuando un parentesis que cierra aparece, se ejecutan todas las operaciones hasta encontrar un parentesis que abre.