

Expresiones en Lenguaje C

		Sintaxis (orden)		Restricción y Semántica (sentido)			
Operador	Descripción	Precedencia	Asociación	Secuencia	Promoción	Notas	Evaluación
(exp)	Expresión con paréntesis	Alta (18)	No aplica	Garantizada	lvalue		
id	Identificador				Terminal		
'char'	Carácter				Terminal		
"string"	Cadena literal				Terminal		
()	Llamada a función	17	Izquierda a derecha	nota_1(nota_2)		1, 2	Tipo función
[]	Elemento de arreglo			apuntador[entero]		3	Tipo arreglo
->	Miembro apuntador estructura			(nota_4)->(nota_7)		4, 7	Tipo apuntador "struct"
.	Miembro estructura			(nota_5)->(nota_7)		5, 7	Tipo apuntador "union"
++	Post-incremento	16	Derecha a izquierda	aritmético++		8	Operando SIN incrementar
--	Post-decremento			apuntador++		9	Operando SIN incrementar
++	Pre-incremento			aritmético--		8	Operando SIN decrementar
--	Pre-decremento			apuntador--		9	Operando SIN decrementar
+	Adición	13	Izquierda a derecha	++aritmético		8	Operando incrementado
-	Substracción			++apuntador		9	Operando incrementado
<<	Desplazamiento izquierda			--aritmético		8	Operando decrementado
>>	Desplazamiento derecha			--apuntador		9	Operando decrementado
<	Menor que	11	Izquierda a derecha	-aritmético			aritmético
<=	Menor que o igual a			~entero			entero
>	Mayor que			!escalar			"int" (0,1)
>=	Mayor que o igual a			(*)()			designador de función
==	Igual	10	Izquierda a derecha	*objeto			lvalue (objeto)
!=	No igual			*apuntador a tipo			"tipo"
&	AND a nivel de "bit"					10	Dirección de operando
^	XOR a nivel de "bit"					11, 12	entero
	OR a nivel de "bit"	9	Izquierda a derecha			13	tipo
&&	AND Lógico						aritmético
	OR Lógico						aritmético
exp-cte	Expresión constante						entero
?:	Condicional	3	Derecha a izquierda				entero
=	Asignación						entero
*							entero
/							entero
%		6	Izquierda a derecha				entero
+=							entero
-=							entero
<<=							entero
>>=		5	Izquierda a derecha				entero
&=							entero
^=							entero
=							entero
,	Coma	1	Izquierda a derecha				entero
							entero
							entero
							entero

exp

Baja

Notas:

- Apuntador a función que retorna "void" o retorna tipo objeto que no sea arreglo
- Número de argumentos igual al de parámetros
- Tipo objeto
- Apuntador a "struct" calificada o no calificada
- Apuntador a "union" calificada o no calificada
- Tipo "struct" o "union" calificadas o no calificadas
- Nombre a miembro de tipo
- Real calificado o no calificado
- Apuntador con "lvalue" modificable
- Designador de función

Resultado de operador [] ó *

- "lvalue" que designa objeto que no es campo de "bit" y no está declarado con "register"
- No se debe aplicar a expresión con tipo función o tipo incompleto
- No se debe aplicar a nombre en paréntesis de tipo o expresión que designa miembro de campo de "bit"
- Entero: Tamaño de operando en "bytes"
- A menos que el nombre de tipo sea "void", éste debe especificar tipos calificados o no calificados y debe tener tipo escalar
- Apuntadores a miembros de estructuras declaradas posteriormente se comparan "mayor que" las declaradas previamente.
- Apuntadores a elementos de arreglos con mayor índice se comparan "mayor que" los de menor índice.
- Apuntadores a miembros de los mismos objetos de la unión se comparan "igual a".
- Ambos operandos aritméticos, ambos operandos misma estructura o unión, mismo tipo "void" o apuntadores.
- Uno es apuntador y el otro es apuntador a constante "NULL", uno es apuntador y otro apunta a otro de tipo "void".

Versión a 6 de Marzo de 2023
Por Ing. Juan Carlos Giraldo, Ph.D.

Tipos de Operadores

	Primario
	Designador
	Aritmético
	Relacional
	Bit
	Lógico
	Asignación
	De flujo

Separación Operadores

	Distinta Categ. Sintáctica
	Misma Categ. Sintáctica