Rank	Operator in C	Description	Result	Associativity
A	()	Grouping	exp	N/A
B1	()	Function call	rexp	L-R
B2	[]	Subscript	lexp	L-R
B3	.	Structure member	lexp	L-R
B4	->	Structure pointer member	lexp	L-R
B5	++	Postfix increment	rexp	L-R
B6	--	Postfix decrement	rexp	L-R
C1	!	Logical negate	rexp	R-L
C2	~	One's complement	rexp	R-L
C3	+	Unary plus	rexp	R-L
C4	-	Unary minus	rexp	R-L
C5	++	Prefix increment	rexp	R-L
C6	--	Prefix decrement	rexp	R-L
C7	*	Indirection (dereference)	lexp	R-L
C8	&	Address of	rexp	R-L
C9	sizeof	Size in bytes	rexp	R-L
D	(type)	Type conversion (cast)	rexp	R-L
E1	*	Multiplication	rexp	L-R
E2	/	Division	rexp	L-R
E3	%	Integer remainder (modulo)	rexp	L-R
F1	+	Addition	rexp	L-R
F2	-	Subtraction	rexp	L-R
G1	<<	Left shift	rexp	L-R
G2	>>	Right shift	rexp	L-R
H1	>	Greater than	rexp	L-R
H2	>=	Greater than or equal	rexp	L-R
H3	<	Less than	rexp	L-R
H4	<=	Less than or equal	rexp	L-R
I1	==	Equal to	rexp	L-R
I2	!=	Not equal to	rexp	L-R
J	&	Bitwise AND	rexp	L-R
K	^	Bitwise exclusive OR	rexp	L-R
L	|	Bitwise inclusive OR	rexp	L-R
M	&&	Logical AND	rexp	L-R
N	||	Logical OR	rexp	L-R
O	?:	Conditional	rexp	N/A
P1	=	Assignment	rexp	R-L
P2	+=	Add to	rexp	R-L
P3	-=	Subtract from	rexp	R-L
P4	*=	Multiply by	rexp	R-L
P5	/=	Divide by	rexp	R-L
P6	%=	Modulo by	rexp	R-L
P7	<<=	Shift left by	rexp	R-L
P8	>>=	Shift right by	rexp	R-L
P9	&=	AND with	rexp	R-L
P10	^=	Exclusive OR with	rexp	R-L
P11	|=	Inclusive OR with	rexp	R-L
Q	,	Comma	rexp	L-R

Note: All operators within a section (between horizontal lines) have the same precedence and the associativity must be applied.