

Precedence	Operator	Description	Associativity
1	++ --	Suffix/postfix increment and decrement	Left-to-right
	()	Function call	
	[]	Array subscripting	
	.	Structure and union member access	
	->	Structure and union member access through pointer	
	(type){list}	Compound literal(C99)	
2	++ --	Prefix increment and decrement	Right-to-left
	+ -	Unary plus and minus	
	! ~	Logical NOT and bitwise NOT	
	(type)	Type cast	
	*	Indirection (dereference)	
	&	Address-of	
	sizeof _Alignof	Size-of Alignment requirement(C11)	
3	* / %	Multiplication, division, and remainder	Left-to-right
4	+ -	Addition and subtraction	
5	<< >>	Bitwise left shift and right shift	
6	< <=	For relational operators < and ≤ respectively	
	> >=	For relational operators > and ≥ respectively	
7	== !=	For relational = and ≠ respectively	
8	&	Bitwise AND	
9	^	Bitwise XOR (exclusive or)	
10		Bitwise OR (inclusive or)	
11	&&	Logical AND	
12		Logical OR	
13 ^[note 1]	?:	Ternary conditional ^[note 2]	
14	=	Simple assignment	
	+= -=	Assignment by sum and difference	
	*= /= %=	Assignment by product, quotient, and remainder	
	<<= >>=	Assignment by bitwise left shift and right shift	
	&= ^= =	Assignment by bitwise AND, XOR, and OR	
15	,	Comma	Left-to-right

Type	Storage size	Value range
char	1 byte	-128 to 127 or 0 to 255
unsigned char	1 byte	0 to 255
signed char	1 byte	-128 to 127
int	2 or 4 bytes	-32,768 to 32,767 or -2,147,483,648 to 2,147,483,647
unsigned int	2 or 4 bytes	0 to 65,535 or 0 to 4,294,967,295
short	2 bytes	-32,768 to 32,767
unsigned short	2 bytes	0 to 65,535
long	4 bytes	-2,147,483,648 to 2,147,483,647
unsigned long	4 bytes	0 to 4,294,967,295

Type character	Print format
d	integer number printed in decimal (preceded by a minus sign if the number is negative)
U	Unsigned integer printed in decimal
f	floating point number (printed in the form <i>ddd.dddddd</i>)
E	floating point number (printed in scientific notation: <i>d.dddEddd</i>)
g	floating point number (printed either as f or E, depending on value and precision)
x	integer number printed in hexadecimal with lower case letters
X	integer number printed in hexadecimal with upper case letters
c	character

Escape sequence	Action
\n	prints a newline
\b	prints a backspace (backs up one character)
\t	prints a tab character
\\	prints a backslash
\"	prints a double quote

```
int main()
{
    printf("Strings:\n");
    const char* s = "Hello";
    printf("\t.%10s.\n\t.%-10s.\n\t.%"s.\n", s, s, 10, s);
    printf("Characters:\t%c %%\n", 65);
    printf("Integers\n");    printf("Decimal:\t%i %d %.6i %i %.0i %+i %u\n", 1, 2,
    3, 0, 0, 4, -1);
    printf("Hexadecimal:\t%x %x %X %#x\n", 5, 10, 10, 6);
    printf("Octal:\t%o %#o %#o\n", 10, 10, 4);
    printf("Floating point\n");
    printf("Rounding:\t%f %.0f %.32f\n", 1.5, 1.5, 1.3);
    printf("Padding:\t%05.2f %.2f %5.2f\n", 1.5, 1.5, 1.5);
    printf("Scientific:\t%E %e\n", 1.5, 1.5);
    printf("Hexadecimal:\t%a %A\n", 1.5, 1.5);
}
```

```
Strings:
    .      Hello.
    .Hello .
    .      Hello.
Characters:   A %
Integers
Decimal:     1 2 000003 0 +4 4294967295
Hexadecimal: 5 a A 0x6
Octal:      12 012 04
Floating point
Rounding:    1.500000 2 1.300000000000000000004440892098500626
Padding:     01.50 1.50 1.50
Scientific:  1.500000E+00 1.500000e+00
Hexadecimal: 0x1.8p+0 0X1.8P+0
```