

symbol	nazwa	przykład	opis przykładu
Operatory arytmetyczne			
*	mnożenie	a = a * 2;	wartość zmiennej a zostanie zwiększona dwukrotnie
/	dzielenie	a = a / b;	w zmiennej a znajdzie się wynik dzielenia a przez b
%	reszta z dzielenia	a = 50 % 2;	w a znajdzie się 0 - reszta z dzielenia 50 przez 2
+	dodawanie	a = a + 3;	wartość zmiennej a zostanie zwiększona o 3
-	odejmowanie	a = b - c;	a przypisana zostanie wartość różnicy między b i c
++	inkrementacja	a++;	zwiększenie a o 1 po użyciu (postinkrementacja)
--	dekrementacja	--a;	zmniejszenie a przed użyciem (predekrementacja)
Operatory bitowe			
	lub bit po bicie (OR)	a = b 0xf0;	a = wartość b z ustawionymi najwyższymi bitami
&	i bit po bicie (AND)	a = b & 0xf0;	a = wartość b z wyzerowanymi najwyższymi bitami
~	negacja poszczególnych bitów (NOT)	a = ~a;	wartość a zostanie zanegowana bit po bicie
^	logiczna suma modulo 2 (XOR)	a = a ^ 0x01;	najmłodszy bit w a zmieni wartość na przeciwną
<<	przesunięcie bitów w lewo	a = 1 << 7;	w a znajdzie się liczba 128 (1 na siódmej pozycji)
>>	przesunięcie bitów w prawo	a = b >> 1;	w a znajdzie się b przesunięte o 1 pozycję w prawo
Operatory porównania			
==	czy równe	if(a == 0) ...	instrukcja zostanie wykonana gdy a równa się 0
!=	czy różne	while(a != 0)...	pętla wykonywana dopuki a różne od 0
>	czy większe	if(a > b)...	instrukcja zostanie wykonana gdy a > b
>=	czy większe lub równe	if(a >= 3)...	jeśli a większe lub równe 3
<	czy mniejsze	while(a < 10)...	pętla wykonywana dopuki a mniejsze od 10
<=	czy mniejsze lub równe	while(a <= 10)...	pętla wykonywana dopuki a mniejsze lub równe 10
Operatory logiczne			
&&	i	if(1 < a && a < 5)...	jeśli a w przedziale otwartym między 1 a 5
	lub	if(a < 1 5 < a)...	jeśli a mniejsze od 1 lub większe od 5
!	nie	if(!(1 < a && a < 5))...	jeśli a nie w przedziale od 1 do 5
Pozostałe			
sizeof(a)	podaje rozmiar w bajtach zmiennej a	a = sizeof(z);	w a znajdzie się rozmiar z
&	podaje adres do zmiennej (wskaźnik)	ptr = &a;	wskaźnikowi ptr zostanie przypisany adres a
*	dostęp do pamięci o podanym adresie	(*ptr) = 10;	przypisanie zmiennej wskazywanej przez ptr wart.10
?:	Operator warunkowy	z = (a > b) ? a : b;	w z znajdzie się większa z wartości
(typ)	Rzutowanie - przekształcenie typu	(char)a	zmienna a zostanie potraktowana jak char

Operatory przypisania

Oprócz opisanych operatorów w C mamy jeszcze sporo operatorów przypisania. Najprostszy z nich = powoduje wpisanie do danej zmiennej wartości znajdującej się zaraz za nią. Operator przypisania może mieć także postać:

a (op) = b;

gdzie (op) oznacza dowolny operator arytmetyczny lub bitowy. Taki zapis jest równoważny z zapisaniem:

a = a (op) b;

Operatory przypisania są wykonywane od prawej strony do lewej. Oznacza to, że zapis

a = b = c = 5;

Spowoduje najpierw wpisanie wartości 5 do zmiennej c, potem c do b, dalej b do a. Ostatecznie we wszystkich zmiennych będzie wartość 5.

Kolejność działań

```
( ) [ ] -> .
! ~ ++ -- * & (typ) sizeof()
* / %
+ -
<< >>
< <= > >=
== !=
&
^
|
&&
||
?:
= += -= /= ...
```

jednoargumentowe * & (operatory adresu) mają priorytet wyższy niż dwuargumentowy operator mnożenia i iloczynu bitowego