

Bitwise:

int a = 10, 1010

int b = 5, 0101

1. AND '&':

a = 1010

b = 0101

a & b = 0000

=

2. OR '|':

a = 1010

b = 0101

a | b = 1111

= 15

3. XOR '∧':

$$a = 1010$$

$$b = 0101$$

$$a \wedge b = 1111$$

$$= 15$$

4. NOT '¬':

$$10 = 1010$$

$$\neg 10 = \begin{array}{r} 1+ \\ 1011 \end{array} = 11$$

↓
denn ist die Zahl negativ

$$: -11$$

$$-10 = 1010 = 0101$$

$$\neg -10 = 0101$$

$$\begin{array}{r} 11+ \\ 0101 \end{array}$$

$$\begin{array}{l} 0110 \\ \rightarrow 1001 = 9 \end{array}$$

5. Left shift 'll'

=> Verschiebe die Zahl, um n position nach links:

Beispiel:

$$10 = 1010$$

$$10 \ll 2$$

$$= 101000$$

$$= 40$$

6. Right shift 'rr':

=> Verschiebe die Zahl um n position nach rechts und bewahrt das Vorzeichen

- wenn die Zahl positiv ist, werden 'nullen' von links eingefügt.

- wenn die Zahl negativ ist, werden 'einsen' von links eingefügt.

Beispiel

$$9 \gg 2$$

$$9 = 1001$$

$$0010 = 2$$

$$-9 \gg 2 = -3$$

$$-9 = 1001 = -0110$$

$$\begin{array}{r} 0110 \\ \hline 1+ \end{array}$$

$$0110 \gg 2$$

$$\begin{array}{r} 1101 \\ \hline 0010 \end{array}$$

Anstatt '0' => '1'

$$\begin{array}{r} 1+ \\ 0011 \\ \hline = -3 \end{array}$$

7. Zero fill Right Shift '777':

- verschiebt die Zahl nach rechts und füllt immer mit Nullen von links ein

$$10 = 10 \text{ } \cancel{10}$$

$$10 \ggg 2$$

$$0010 = 2$$

Zusammenfassung:

- \ll : verschiebt die Zahl nach links

$$\text{so } 10 \ll 2 = 40 \text{ und } -10 \ll 2 = -40$$

- \gg : verschiebt die Zahl nach rechts und betrachtet das Vorzeichen der Zahl.

so $10 \gg 2$: verschiebe die Zahl nach rechts und fülle 2 **nullen** von links

$-10 \gg 2$: verschiebe die Zahl nach rechts und fülle mit **einsen** von links ein

- \ggg : verschiebe die Zahl nach rechts und fülle immer mit **nullen** von links ein, damit die Zahl positiv bleibt.