

```
// Einige ausgewählte Operatoren von C++
```

```
#include <iostream>
```

```
#define wert1(x)      std::cout << #x " = " << x << std::endl;  
#define wert3(x, y, z) std::cout << #x " = " << x << \n  
                    "\t" #y " = " << y \n  
                    << "\t" #z " = " << z << std::endl;
```

```
int main () {
```

```
    // Arithmetische Ausdrücke
```

```
    int x1 = - 3 + 4 * 5 - 6;
```

```
    wert1 (x1);
```

```
    int x2 = 3 + 4 % 5 - 6;
```

```
    wert1 (x2);
```

```
    int x3 = - 3 * 4 % - 6 / 5;
```

```
    wert1 (x3);
```

```
    int x4 = (7 + 6) % 5 / 2;
```

```
    wert1 (x4);
```

```
    // Zuweisungen
```

```
    int y1 = 2;
```

```
    y1 *= 3 + 2;
```

```
    wert1 (y1);
```

```
    int y2, y3;
```

```
    y1 *= y2 = y3 = 4;
```

```
    wert1 (y1); wert1 (y2); wert1 (y3);
```

```
    int y4 = y2 == y3;
```

```
    wert1 (y4);
```

```
    y4 = (y2 == y3);
```

```
    wert1 (y4);
```

```
// Logische Operationen
```

```
int z1 = 2;
```

```
int z2 = 1;
```

```
int z3 = 0;
```

```
z1 = z1 && z2 || z3;
```

```
wert1 (z1);
```

```
wert1 ((z1 = z1 && z2 || z3));
```

```
wert1 ((z1 || !z2 && z3));
```

```
z1 = z2 = 1;
```

```
z3 = z1 ++ - 1;
```

```
wert1 (z1); wert1 (z3);
```

```
z3 = z3 += - z1 ++ + ++ z2;
```

```
wert1 (z1); wert1 (z3);
```

```
wert1 ((z3 = z1 / ++ z1));
```

```
// Bitmanipulationen
```

```
int a1 = 3;
```

```
int a2 = 2;
```

```
int a3 = 1;
```

```
wert1 ((a1 | a2 & a3));
```

```
wert1 ((a1 | a2 & ~a3));
```

```
wert1 ((a1 ^ a2 & ~a3));
```

```
wert1 ((a1 & a2 && a3));
```

```
a1 = 1; a2 = -1;
```

```
wert1 ((!a1 | a1));
```

```
wert1 ((~a1 | a1));
```

```
wert1 ((a1 ^ a1));
```

```
wert1 ((a1 <<= 3));
```

```
wert1 ((a2 <<= 3));
```

```
wert1 ((a2 >>= 3));
```

```

int b1 = 1, b2 = 1, b3 = 1;
b1 += b2 += b3;
wert1 (( b1 < b2 ? b2 : b1));
wert1 ((b1 < b2 ? b1 ++ : b2 ++));
wert1 (b1); wert1 (b2);
wert1 ((b3 += b1 < b2 ? b1 ++ : b2 ++));
wert1 (b2); wert1 (b3);
b1 = 3; b2 = b3 = 4;
wert1 ( ((b3 >= b2 >= b1) ? 1 : 0 ));
wert1 (( b3 >= b2 && b2 >= b1));

```

// Vorrang und Bewertung

```
int c1, c2, c3;
```

```

c1 = c2 = c3 = 1;
++c1 || ++c2 && ++c3;
wert3 (c1, c2, c3);

```

```

c1 = c2 = c3 = 1;
++c1 && ++c2 || ++c3;
wert3 (c1, c2, c3);

```

```

c1 = c2 = c3 = 1;
++c1 && ++c2 && ++c3;
wert3 (c1, c2, c3);

```

```

c1 = c2 = c3 = -1;
++c1 || ++c2 && ++c3;
wert3 (c1, c2, c3);

```

```

c1 = c2 = c3 = -1;
++c1 && ++c2 || ++c3;
wert3 (c1, c2, c3);

```

```

c1 = c2 = c3 = -1;
++c1 && ++c2 && ++c3;
wert3 (c1, c2, c3);

```

```

return 0;
} //main

```

/*

Ergebnis:

```

x1 = 11
x2 = 1
x3 = 0
x4 = 1
y1 = 10
y1 = 40
y2 = 4
y3 = 4
y4 = 1
y4 = 1
z1 = 1
(z1 = z1 && z2 || z3) = 1
(z1 || !z2 && z3) = 1
z1 = 2
z3 = 0
z1 = 3
z3 = 0
(z3 = z1 / ++ z1) = 1
(a1 | a2 & a3) = 3

```

(a1 | a2 & ~a3) = 3
(a1 ^ a2 & ~a3) = 1
(a1 & a2 && a3) = 1
(!a1 | a1) = 1
(~a1 | a1) = -1
(a1 ^ a1) = 0
(a1 <<= 3) = 8
(a2 <<= 3) = -8
(a2 >>= 3) = -1
(b1 < b2 ? b2 : b1) = 3
(b1 < b2 ? b1 ++ : b2 ++) = 2
b1 = 3
b2 = 3
(b3 += b1 < b2 ? b1 ++ : b2 ++) = 4
b2 = 4
b3 = 4
((b3 >= b2 >= b1) ? 1 : 0) = 0
(b3 >= b2 && b2 >= b1) = 1
c1 = 2 c2 = 1 c3 = 1
c1 = 2 c2 = 2 c3 = 1
c1 = 2 c2 = 2 c3 = 2
c1 = 0 c2 = 0 c3 = -1
c1 = 0 c2 = -1 c3 = 0
c1 = 0 c2 = -1 c3 = -1

***/**