🔑 Operators in C (with examples)

1. Arithmetic Operators

Perform basic math.

c

Copy code

```c
#include <stdio.h>
int main() {
    int a = 10, b = 3;
    printf("Add: %d\n", a + b);   // 13
    printf("Sub: %d\n", a - b);   // 7
    printf("Mul: %d\n", a * b);   // 30
    printf("Div: %d\n", a / b);   // 3 (integer division)
    printf("Mod: %d\n", a % b);   // 1
    return 0;
}
```

2. Unary Operators

Work on a single operand.

c

Copy code

```c
#include <stdio.h>
int main() {
    int x = 5;
    printf("Pre-Increment: %d\n", ++x); // 6 (x becomes 6, then printed)
    printf("Post-Increment: %d\n", x++); // 6 (prints old value, then x=7)
    printf("Now x: %d\n", x);           // 7
    printf("Negative: %d\n", -x);       // -7
    printf("Size of x: %lu\n", sizeof(x)); // 4 (on most systems)
    return 0;
}
```

3. Relational Operators

Compare two values → return **1 (true)** or **0 (false)**.

c

Copy code

```c
#include <stdio.h>
int main() {
    int a = 5, b = 8;
    printf("a == b: %d\n", a == b); // 0
    printf("a != b: %d\n", a != b); // 1
    printf("a > b : %d\n", a > b);  // 0
    printf("a < b : %d\n", a < b);  // 1
    printf("a >= b: %d\n", a >= b); // 0
    printf("a <= b: %d\n", a <= b); // 1
    return 0;
}
```

4. Logical Operators

Combine relational expressions.

c

Copy code

```c
#include <stdio.h>
int main() {
    int a = 5, b = 10;
    printf("(a > 0 && b > 0): %d\n", (a > 0 && b > 0)); // 1
    printf("(a > 0 || b < 0): %d\n", (a > 0 || b < 0)); // 1
    printf("!(a == 5): %d\n", !(a == 5));               // 0
    return 0;
}
```

5. Assignment Operators

Assign and update values.

c

Copy code

```c
#include <stdio.h>
int main() {
    int a = 10;
    a += 5;  // a = 15
    a -= 3;  // a = 12
    a *= 2;  // a = 24
    a /= 6;  // a = 4
    a %= 3;  // a = 1
    printf("Final value of a: %d\n", a); // 1
    return 0;
}
```

6. Conditional (Ternary) Operator

Shorthand for `if-else`.

c

Copy code

```c
#include <stdio.h>
int main() {
    int a = 20, b = 15;
    int max = (a > b) ? a : b;
    printf("Max = %d\n", max); // 20
    return 0;
}
```

7. Bitwise Operators

Operate at the bit level (binary).

c

Copy code

```c
#include <stdio.h>
int main() {
    int x = 5, y = 3; // (5=0101, 3=0011 in binary)

    printf("AND: %d\n", x & y);   // 1  (0101 & 0011 = 0001)
    printf("OR: %d\n", x | y);    // 7  (0101 | 0011 = 0111)
    printf("XOR: %d\n", x ^ y);   // 6  (0101 ^ 0011 = 0110)
    printf("NOT: %d\n", ~x);      // -6 (bitwise NOT of 0101 → ...1010)
    printf("Left Shift: %d\n", x << 1);  // 10 (0101 << 1 = 1010)
    printf("Right Shift: %d\n", x >> 1); // 2  (0101 >> 1 = 0010)
    return 0;
}
```