

Forum:[Hack Forums](#)

Thread: [Boolean operators in C](#)

schassis - August 27, 2025, 11:36 am

i do not understand them in the slightest

Code A. `!( 1 || 0 )` ANSWER: 0

B. `!( 1 || 1 && 0 )` ANSWER: 0 (AND is evaluated before OR)

C. `!( ( 1 || 0 ) && 0 )` ANSWER: 1 (Parenthesis are useful)

ill start with A, i read it as "not or 1 or 0 so it must equal 0" like how in the fuck does that work?!

i dont understand "!" either; if true: false, if false: true. like what the hell??? why???

webdev77 - August 27, 2025, 3:13 pm

It's the order of operations you aren't seeing or apparently following. Just like in normal math, C also has order of operations.

https://en.cppreference.com/w/c/language/operator_precedence.html

1a. 1 or 0 results in 1

1b. Not 1 results in 0

2a. 1 and 0 results in 0.

2b. 1 or 0 results in 1

2c. Not 1 results in 0

3a. 1 or 0 results in 1

3b. 1 and 0 results in 0

3c. Not 0 results in 1

schassis - August 27, 2025, 3:49 pm

(Aug 27, 2025 03:13 PM)webdev77 Wrote: [\[CITE\]](#) It's the order of operations you aren't seeing or apparently following. Just like in normal math, C also has order of operations.

https://en.cppreference.com/w/c/language/operator_precedence.html

1a. 1 or 0 results in 1

1b. Not 1 results in 0

2a. 1 and 0 results in 0.

2b. 1 or 0 results in 1

2c. Not 1 results in 0

3a. 1 or 0 results in 1

3b. 1 and 0 results in 0

3c. Not 0 results in 1

oh my god i get it now! haha!

the order of operations is what i didnt know existed, thank you dude

i hate being slow tho, like i can feel my slowness - its aggravating

(Aug 27, 2025 03:49 PM)schassis Wrote: [CITE](#) oh my god i get it now! haha!

the order of operations is what i didnt know existed, thank you dude

i hate being slow tho, like i can feel my slowness - its aggravating

You're very much welcome. Keep at it and you'll be building fun stuff in no time.

Here's a freely available hex dump function that'll help you when you play around with strings and any buffers containing non-printable characters. I don't know if you have covered strings and buffers in C yet but you likely will soon if you haven't yet.

Function ([Click to View](#))

Code include <stdio.h>

include <string.h>

```
void hexdump(const void *data, size_t size) {
    const unsigned char *byte = (const unsigned char *)data;

    for (size_t i = 0; i < size; i += 16) {
        printf("%08x ", i);

        for (size_t j = 0; j < 16; ++j) {
            if (i + j < size) {
                printf("%02x ", byte[i + j]);
            } else {
                printf(" ");
            }
            if (j == 7) printf(" ");
        }

        printf(" |");

        for (size_t j = 0; j < 16 && (i + j) < size; ++j) {
            unsigned char c = byte[i + j];
            if (c >= 32 && c <= 126) {
                printf("%c", c);
            } else {
                printf(".");
            }
        }

        printf("\n");
    }
}

int main() {
    const char *text = "Demo \x00 containing null chars \x00 you can't print as a string";
    hexdump(text, 64);
    return 0;
}
```

