Bitwise operator review:

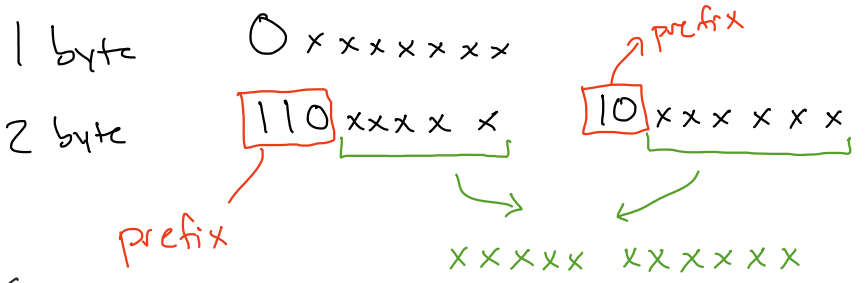
```
uint8_t last4(uint8_t n) {
    return 0b00001111 & n;
}
```

bitmask: 0000 1111 — mask: 1010 1100

```
  0000 1111
& 1010 1100
-----------
  0000 1100
```

last4(0b10101100) => 0b00001100

last4(0b10100000) => 0b00000000




```
uint8_t last4(uint8_t n) {
    return 0b00001111 & n;
}
uint8_t combine(uint8_t a, uint8_t b) {
    return (last4(a) << 4) | last4(b);
}
```

combine(0b10101111, 0b1100 0011) => _____

(last4(0b10101111) << 4) | last4(0b1100 0011)

(0b0000 1111 << 4) | last4(0b1100 0011)

0b1111 0000 | last4(0b1100 0011)

0b1111 0000 | 0b0000 0011

0b1111 0011

Signed vs. unsigned numbers

4-bit numbers (2's comp)

	unsigned	signed
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	-8
1001	9	-7
1010	10	-6
1011	11	-5
1100	12	-4
1101	13	-3
1110	14	-2
1111	15	-1

uint8_t int8_t

uint32_t int32_t

unsigned signed

int

abcd

a*8 + b*4 + c*2 + d*1] unsigned interpretation

a* -8 + b*4 + c*2 + d*1] signed interp

Every bit pattern can be interpreted as signed or unsigned.

We use code (types, etc) to tell C which operations to use.

%u vs %d

(unsigned) (signed)