```
 1 #include <stdint.h>
 2 #include <stdio.h>
 3
 4 void print_as_bin(char c) {
 5     for(int place = 128; place > 0; place /= 2) {
 6         if((c & place) == 0) printf("0"); else printf("1");
 7     }
 8 }
 9
10 int main() {
11     printf("sizeof(char): %ld\n", sizeof(char));
12
13     printf("sizeof(int8_t): %ld\n", sizeof(int8_t));
14     printf("sizeof(uint8_t): %ld\n", sizeof(uint8_t));
15
16     printf("sizeof(int16_t): %ld\n", sizeof(int16_t));
17     printf("sizeof(uint16_t): %ld\n", sizeof(uint16_t));
18
19     printf("sizeof(int32_t): %ld\n", sizeof(int32_t));
20
21     printf("sizeof(int64_t): %ld\n", sizeof(int64_t));
22
23     printf("sizeof(int): %ld\n", sizeof(int));
24
25     char c = 128;
26     int32_t i = 32;
27     char c2 = c + 1;
28
29     printf("sizeof(c): %ld\n", sizeof(c));
30     printf("sizeof(i): %ld\n", sizeof(i));
31     printf("sizeof(c * 4000): %ld\n", sizeof(c * 4000));
32     printf("sizeof(c + 1): %ld\n", sizeof(c + 1));
33     printf("sizeof(c2): %ld\n", sizeof(c2));
34
35     char s = 200;
36     unsigned char u = 200;
37
38     printf("s: "); print_as_bin(s); printf("\t\tu: "); print_as_bin(u); printf("\n");
39
40     printf("s as hhx: %hhx\t\tu as hhx: %hhx\n", s, u);
41     printf("s as x: %x\tu as x: %x\n", s, u);
42
43     printf("s < 127: %d u < 127: %d\n", s > 127, u > 127);
44
45 }
```

Handwritten annotations:

sizeof(___)

NOT a function
it's a neat compiler feature

takes a type or expression

reports # of bytes taken up
by that type

— 1 (line 11)
— 1 (line 13)
— 1 (line 14)
2 (line 16)
2 (line 17)
4 (line 19)
8 (line 21)
8 (line 23)

8 bits of integer
e.g. 0 - 255
goes $0 - (2^{16} - 1)$
$-2^{15} - (2^{15}-1)$

4 codespaces
4 jeng6
4 mac

there are systems where int is 8

→ 1 (line 29)
→ 4 (line 30)
→ 4 (line 31)
→ 4 (line 32)
→ 1 (line 33)

char c2 = c + 1;

stored as 1 byte char in C2
despite maybe briefly using 4 bytes for c+1

```
$ gcc size.c -o size
$ ./size
sizeof(char): 1
sizeof(int8_t): 1
sizeof(uint8_t): 1
sizeof(int16_t): 2
sizeof(uint16_t): 2
sizeof(int32_t): 4
sizeof(int64_t): 8
sizeof(int): 4
sizeof(c): 1
sizeof(i): 4
sizeof(c * 4000): 4
sizeof(c + 1): 4
sizeof(c2): 1
s: 11001000        u: 11001000
s as hhx: c8        u as hhx: c8
s as x: ffffffc8        u as x: c8
s < 127: 0 u < 127: 1
```