

```
1 #include <stdio.h>
2
3 int main() {
4     char hello[] = "Hello!";
5     char hellonum[] = { 72, 101, 108, 108, 111, 33, 0 };
6     char hellobin[] = { 0b1001000, 0b1100101, 0b1101100, 0b1101100, 0b1101111, 0b100001, 0b0 };
7
8     puts(hello);
9     puts(hellonum);
10    puts(hellobin);
11
12    printf("%c %c %c\n", hello[0], hello[1], hello[2]);
13
14    printf("%d %d %d\n", hello[0], hellonum[0], hellobin[0]);
15    printf("%c %c %c\n", hello[0], hellonum[0], hellobin[0]);
16 }
```

interpreted as decimal
could write as
could write as
could write as
0b10001000
0b1100101
0b1101100
0b1101100
0b1101111
0b100001
0b0

↳ Ob says "here comes binary"

```
$ gcc hellobin.c -o hellobin
$ ./hellobin
Hello!
Hello!
Hello!
Hello!
```

```
printf("____", ...values)
```

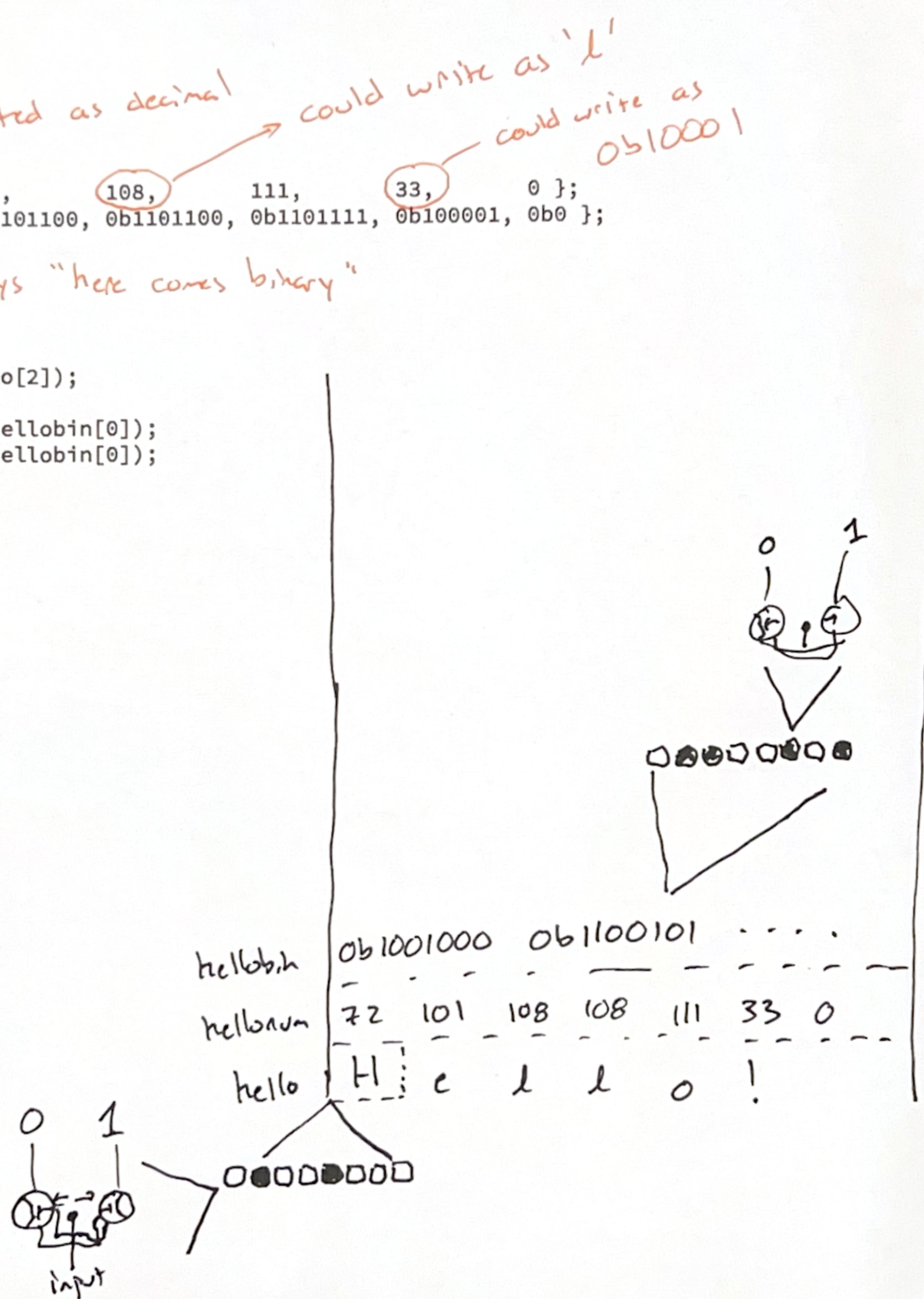
format string

%c means character

%d means decimal

hello[1] hellabin[0] etc.

arrays are 0-indexable



72

why?

$$01001000$$

$2^7 \quad 2^6 \quad | \quad / \quad / \quad | \quad | \quad \backslash \quad 2^0$
 $2^5 \quad 2^4 \quad 2^3 \quad 2^2 \quad 2^1 \quad 2^0$

$$1 * 2^6 + 1 * 2^3$$

$$64 + 8 = 72$$

01001111

in decimal: 79 '0'(the letter)

n	2^n
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

237

$$200 + 30 + 7$$

$$2 * 100 + 3 * 10 + 7 * 1$$

$$2 * 10^2 + 3 * 10^1 + 7 * 10^0$$

hundreds place → tens place
 / ones place
 4 0 3

$$400 + 00 + 3$$

$$4 * 100 + 0 * 10 + 3 * 1$$

$$4 * 10^2 + 0 * 10^1 + 3 * 10^0$$

"base 10"

"base 2"

10ⁿ because 10 digits 0-9In binary, use 2ⁿ

1011

$$1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0$$

$$1 * 8 + 0 * 4 + 1 * 2 + 1 * 1$$

11 (in decimal)

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 int main() {
6     char hello[] = "Hello class!";
7     char apple[] = "apple";
8     char hello_0[] = "Hello\0class!";
9
10    printf("%ld\n", strlen(hello));
11    printf("%ld\n", strlen(apple));
12
13    printf("%s, length: %ld\n", hello_0, strlen(hello_0));
14
15    hello[7] = 0;      "Hello_c\0ass!"
16
17    printf("%s, length: %ld\n", hello, strlen(hello));
18
19    return 0;
20 }
```

C implicitly includes stdlib.h

what's this this
"null"

```
$ gcc strlen.c -o strlen  
$ ./strlen  
12  
5  
Hello, length 5  
Hello c, length: 7
```

- C implicitly includes a 0 at the end of the string

what's this thing? \0
"null character"

";

'\0' == 0

.lo_0, strlen(hello_0));
means: here's the end of the string

%ld	long decimal
%s	string

```
1 #include <stdio.h>
2
3 int main() {
4     char hi[] = "Hi all!";
5     char helloeveryone[] = { 'H', 'e', 'l', 'l', 'o', ',', ',',
6                             ' ', 'e', 'v', 'e', 'r', 'y', 'o', 'n', 'e', '!', };
7     puts(hi);
8     puts(helloeveryone);
9 }
```

```
$ gcc adjacent.c -o adjacent  
$ ./adjacent
```