CS107 Lecture 3: Extra Practice

Bits and Bytes, Integer Representations

Reading: Bryant & O'Hallaron, Ch. 2.2-2.3 (skim)

Based on slides created by Cynthia Lee, Lisa Yan, Nick Troccoli, Chris Gregg, and others.

Practice: Two’s Complement

Fill in the below table:

It’s easier to compute base-10 for positive numbers, so use two’s complement first if negative.

	`char x = ____;`		`char y = -x;`	
	`decimal`	`binary`	`decimal`	`binary`
1.		`0b1111 1100`		
2.		`0b0001 1000`		
3.		`0b0010 0100`		
4.		`0b1101 1111`		

Practice: Two’s Complement

Fill in the below table:

It’s easier to compute base-10 for positive numbers, so use two’s complement first if negative.

	`char x = ____;`		`char y = -x;`	
	decimal	binary	decimal	binary
1.	-4	0b1111 1100	4	0b0000 0100
2.		0b0001 1000		
3.		0b0010 0100		
4.		0b1101 1111		

Practice: Two’s Complement

Fill in the below table:

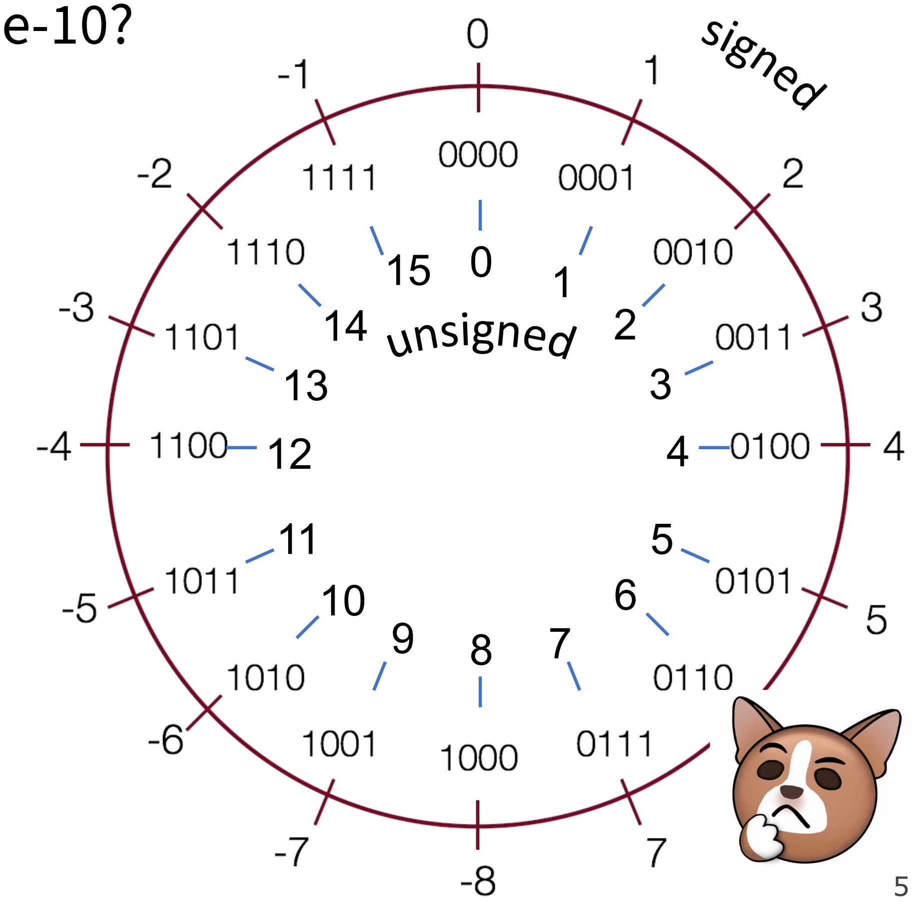
It’s easier to compute base-10 for positive numbers, so use two’s complement first if negative.

	`char x = ____;`		`char y = -x;`	
	decimal	binary	decimal	binary
1.	-4	0b1111 1100	4	0b0000 0100
2.	24	0b0001 1000	-24	0b1110 1000
3.	36	0b0010 0100	-36	0b1101 1100
4.	-33	0b1101 1111	33	0b0010 0001

Intentionally Ambiguous Question

What is the following base-2 number in base-10?

`0b1101`



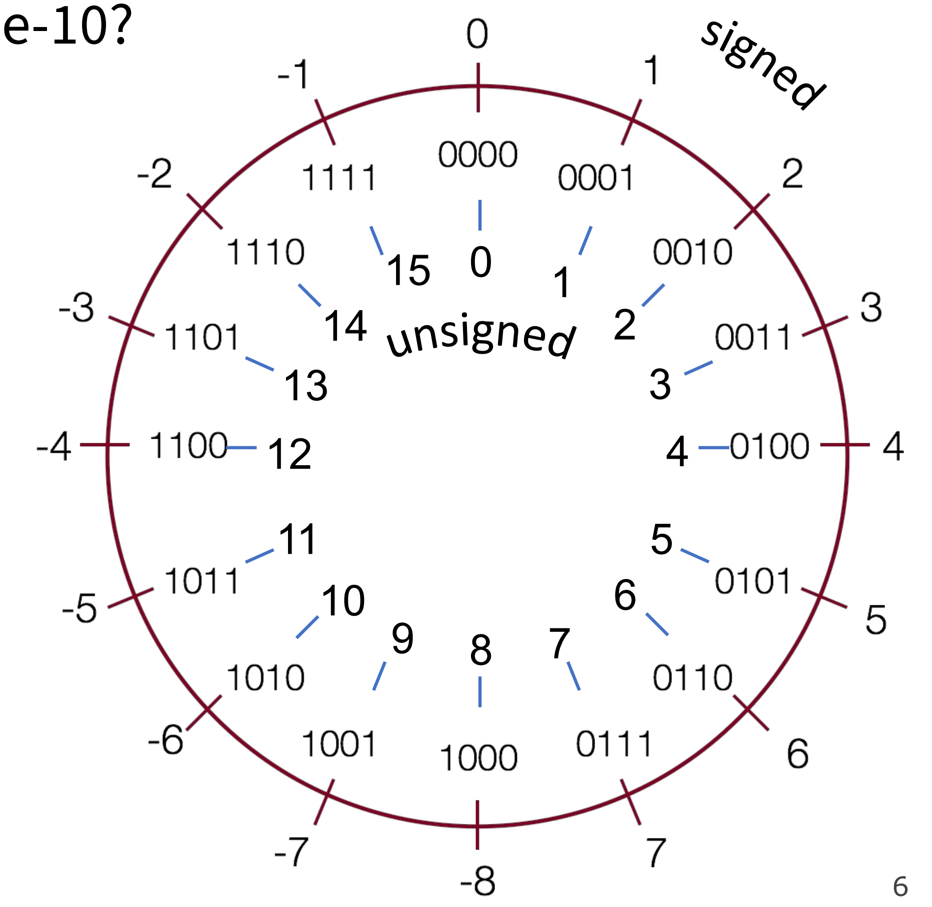
Intentionally Ambiguous Question

What is the following base-2 number in base-10?

```
0b1101
```

If 4-bit signed: -3
If 4-bit unsigned: 13
If > 5-bit signed or unsigned: 13

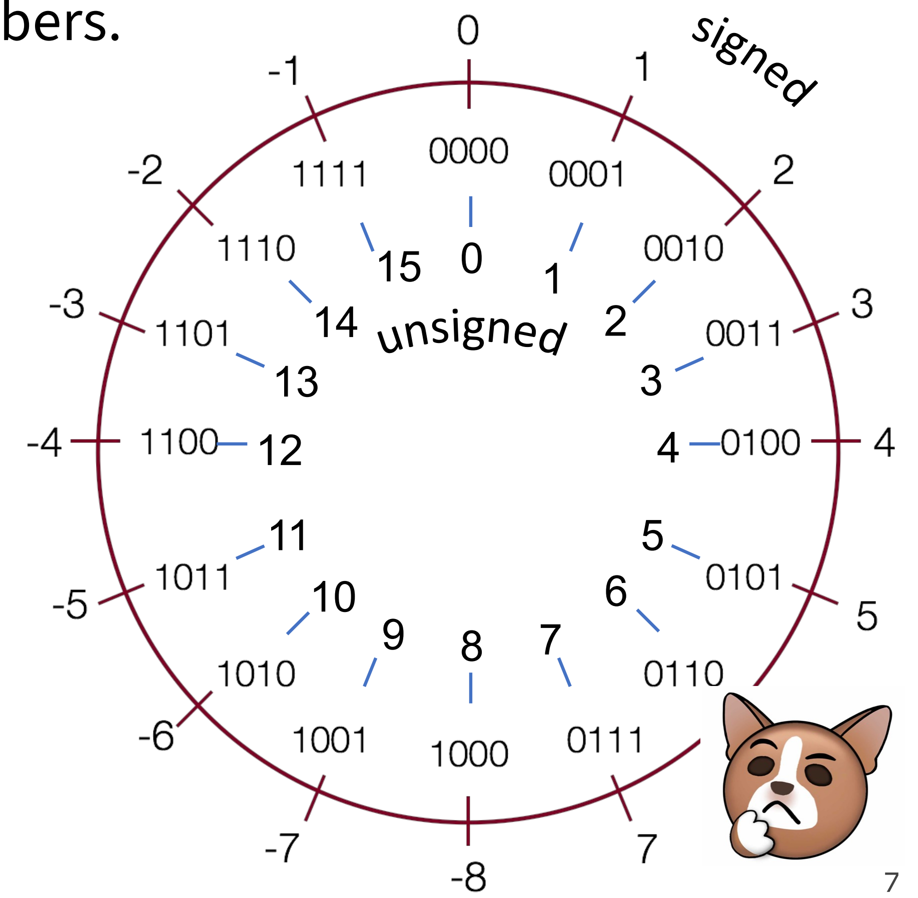
You need to know the type if you're to know the number! (Note by default, numeric constants in C are **signed int**s)



Overflow Question

What is happening here? Assume 4-bit numbers.

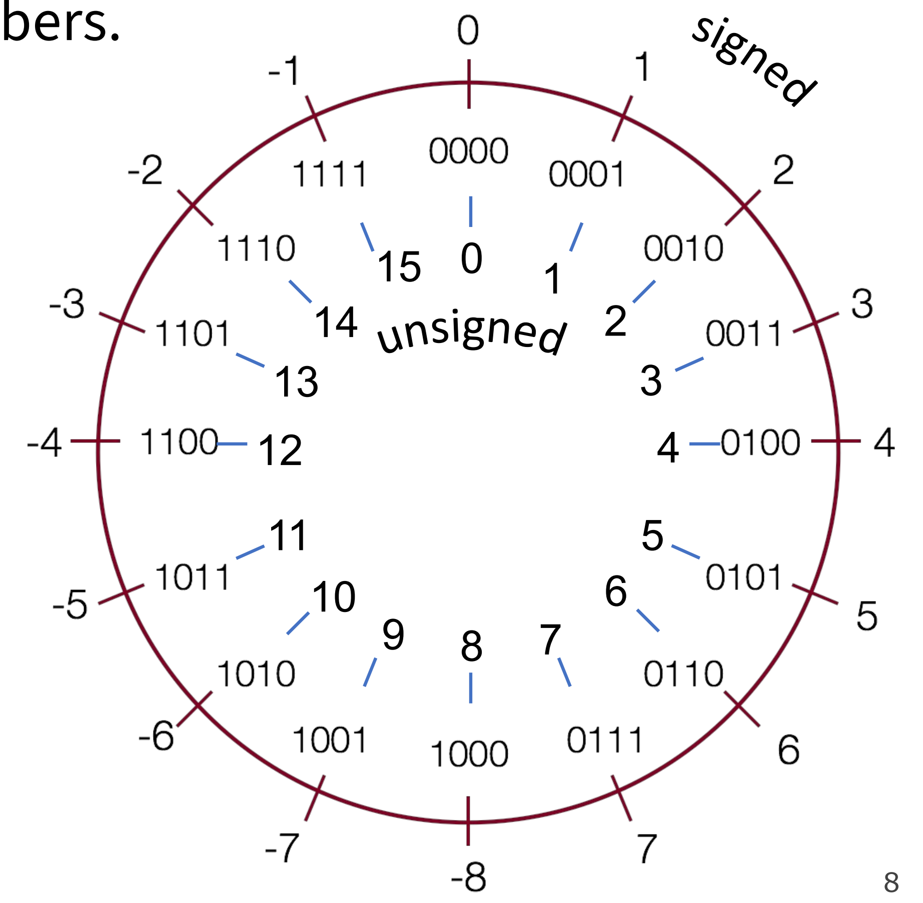
```
   0b1101
 + 0b0100
 ________
```



Overflow Question

What is happening here? Assume 4-bit numbers.

```
  0b1101
+ 0b0100
```



Signed	Unsigned
-3 + 4 = 1	13 + 4 = 1
No overflow	Overflow

Limits and Comparisons Question

1. What is the …

	largest unsigned?	largest signed?	smallest signed?
`char`			
`int`			

Limits and Comparisons Question

1. What is the …

	largest unsigned?	largest signed?	smallest signed?
char	$2^{8}-1 = 255$	$2^{7}-1 = 127$	$-2^{7} = -128$
int	$2^{32}-1 = 4294967296$	$2^{31}-1 = 2147483647$	$-2^{31} = -2147483648$

These are available as **UCHAR_MAX**, **INT_MIN**, **INT_MAX**, etc., via **<limits.h>**.