

# Section 4

# Bitwise Operations





Don't forget to start recording



# Unix Tip Spotlight

- How do you stop a program that's stuck in an infinite loop?
- <CTRL+C>: Asks the program nicely to stop
  - Usually results in the program terminating itself
- <CTRL+Z>: Asks the program not so nicely to stop
  - Usually results in the program pausing, but not terminating



# Announcements

- Thanks for enrolling! (Please enroll if you haven't :))
- I'll announce 1:1s next Tuesday when my personal schedule should be more finalized
- Office Hours have been fairly empty



# Bird's Eye View

Day	Week 2 Wednesday	Thursday	Friday	Week 3 Monday	Tuesday	Wednesday	Thursday	Friday
CS 107A		Section: Bitwise Operations			Section: Chars and Strings		Section: More C-Strings	
CS 107	Lab 1: Integers / Bits		Lecture: Chars and C Strings	Lecture: More C-Strings		Lab 2: C-Strings		Lecture: Arrays and Pointers
CS 107 assignments						assign1 due, assign2 released		assign1 due



# Agenda

- Overflow in Pokemon
- Bitwise Operators
- Bitwise Operator Practice
- Bitwise Coding Practice



# Overflow in Pokemon

<https://www.youtube.com/watch?v=ftFtkc5C1Qg>



# Bitwise Operators





# Bitwise AND: &

- 1 ONLY IF both are 1
- Like &&, but works on ints, not booleans, and the operation is applied to each bit in the int
- Example:
  - `0x9009 & 0x1e = 0x8`

&	0	1
0	0	0
1	0	1



# Bitwise OR: |

- 1 UNLESS both are 0
- Like ||, but works on ints, not booleans, and the operation is applied to each bit in the int
- Example:

○ `0x9009 | 0x1e = 0x901f`

	0	1
0	0	1
1	1	1



# Bitwise XOR: $\wedge$

- 1 when they're different
- Like  $\neq$ , but works on ints, not booleans, and the operation is applied to each bit in the int
- Example:
  - $0x9009 \wedge 0x1e = 0x9017$

$\wedge$	0	1
0	0	1
1	1	0

# Bitwise Complement: ~

- Just flip the bit
- Somewhat like !, but works on ints, not booleans, and the operation is applied to each bit in the int
- Example:
  - $\sim 0x90091e = 0xff6ff6e1$
  - $\sim 0 = -1$
  - $\sim (-1) = 0$
  - $\sim 1 = -2$

~

0

1

1

0



# Bitwise Left Shift: <<

- ~Move to the left~
- The leftmost bits get deleted (check the bit width!), and 0s are brought in on the right
- Example:
  - `0x90091e << 12 = 0x91e000`
  - `0x90091e << 1 = 0x120123c`



# Bitwise Right Shift: >>

- ~Move to the right~
- The rightmost bits get deleted, and bits are brought in on the left
  - For signed integer types, the bits brought in are copies of the sign bit (“arithmetic right shift”)
  - For unsigned integer types, the bits brought in are 0 (“logical right shift”)
- Example:
  - `(signed int)0xFFFFFFFF00 >> 4 = 0xFFFFFFFFF0`
  - `(unsigned int)0xFFFFFFFF00 >> 4 = 0x0FFFFFFF0`



# Section 4 Worksheet

On the course website

- Bitwise Operator Practice
- Bitwise Coding Practice