Week 2 Thursday
Bitwise Operations

Fill in the check-in form on cs107a.stanford.edu!
Announcements

● Please use your enrollment code on Axess~
● I’ve decided to record sections. Watching recordings don’t count as attendance credit unless separately arranged with me
  ○ Recordings are on Canvas, so you can also use the People tab if you like
● If you get sick, let me know, we’ll find a way for you to get credit
● First 1:1s delayed because I’m tired~
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 (prefer <CTRL+C>)
  - Usually results in the program pausing, but not terminating
Agenda

- Overflow Practice
- Overflow in Pokemon
- Casting Practice
- Bitwise Operators
- Bitwise Practice
Overflow Practice

https://web.stanford.edu/class/cs107a/exercises/integer-representations/
p5
Overflow in Pokémon

https://www.youtube.com/watch?v=ftFtkc5C1Qg
Casting Practice

https://web.stanford.edu/class/cs107a/exercises/integer-casting/
p3.2, p4
Bitwise Operators
### Bitwise Operator Truth Tables

<table>
<thead>
<tr>
<th>&amp;</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>^</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>~</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

- **&**: Commonly used for isolating bits via a mask.
- **|**: Commonly used for adding in bits via a mask.
- **^**: Commonly used for toggling bits via a mask.
- **~**: Inverts bits.
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:
  - $\text{0xdeadbeef} \ll 12 = \text{0xdbbeef000}$
  - $\text{0xdeadbeef} \ll 1 = \text{0xbd5b7dde}$
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 = 0xFFFFFFFF00000000
  - (unsigned int)0xFFFFFFFF00 >> 4 = 0x0FFFFFFF0
Bitwise Practice

https://web.stanford.edu/class/cs107a/exercises/bitwise-operations/

p1 (debugging demo), p3