lecture 3

makefile, assembled & interpreted by make

- investigate make-undo & makefile, take notes

- we're going to have to make our own makefiles

logical operations
||

Bitwise operator
|

XOR

$A \oplus B = (A \vee B) \wedge \neg(A \wedge B)$

$= (A \wedge \neg B) \vee (\neg A \wedge B)$

xor tells us how many bits have changed

- $A \oplus A = 0$
- $A \oplus 0 = A$
- $A \oplus 1 = \neg A$
- $A \oplus (B \oplus C) = (A \oplus B) \oplus C$

```
printf("%d", 1==2);
        ↑ print integer
```

- can include <stdbool.h>
  - allows you to have bool data type

---

note that can'll let you divide by 0, it won't check if a value is 0 before dividing it.

this is called guarding

C has the goto ability like assembly

```
( ... ) goto hello;

hello: msg;
 goto done;

Done: printf(msg);
```

---

while & for are top test loops

do while is a bottom test loop
- allows you to do something atleast once

use break to get out of a loop if needed

C++ allows you to handle exceptions, C does not

continue, doesn't jump out of loop. jumps to start of loop

---

lets compute $\sqrt{2}$
there is no sqrt library built in

$$\sqrt{2} = x^2 - 2 = 0 \qquad 0 \le x \le 2$$

ternary operator

- binary search
be careful of floats! equivalence is hard to achieve