1/20/23

Minute & seconds are 60 b.c of Babylonians - very easy to divide

0 from hindus

number system from arabics

metric system from French

Freeman Dyson Veruer vici

- Dyson spheres

$$\pi = 2 \cdot \log_i(i)$$

Any alphabet → every member of $\mathbb{N}$ can be represented as

$$a_k b^k + a_{k-1} b^{k-1} + \ldots + a_1 b^1 + a_0 b^0 \quad a_i \in \{0, \ldots, b-1\}$$

where b is base, looks complex but this is how we code any number

(balanced ternary [negative number base])

- standard unix permissions are in octent

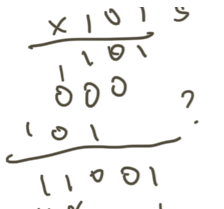
License plates are just base 32

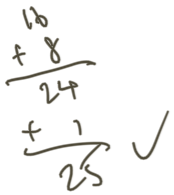
Int categories:

char 8	uint8_t	- if you don't care & you know it won't going to be enormous, just use an int
short 16	⋮	
int 32 long 32	⋮	
long long 64		- if it's huge, then use the appropriate number

$101_5 = 25$

computers cannot technically represent





all numbers, all natural #s.
you would need infinite storage

real #s are continuous, computers can only represent a subset

floating points are only approximation
- technically only a subset of rationals

float single precision 32 bits
double double precision 64 bits
long double extended precision 128 bits
(intel stores double as 80 bits)

fundamental theorem
n∈ℕ has unique prime factorization

0.1 = $\frac{1}{10}$ cannot be represented perfectly in binary

this is why it's so dangerous to use equality w/ floating points
sometimes compilers will print out decimals that aren't the actual value
only things like $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$ can be perfectly represented

Algorithms can never create truely random #s
can only get pseudo random
(technically intel has a module that uses physics to create entropy)

Mod only works on integers & non-negative