

Engineering



& Physics

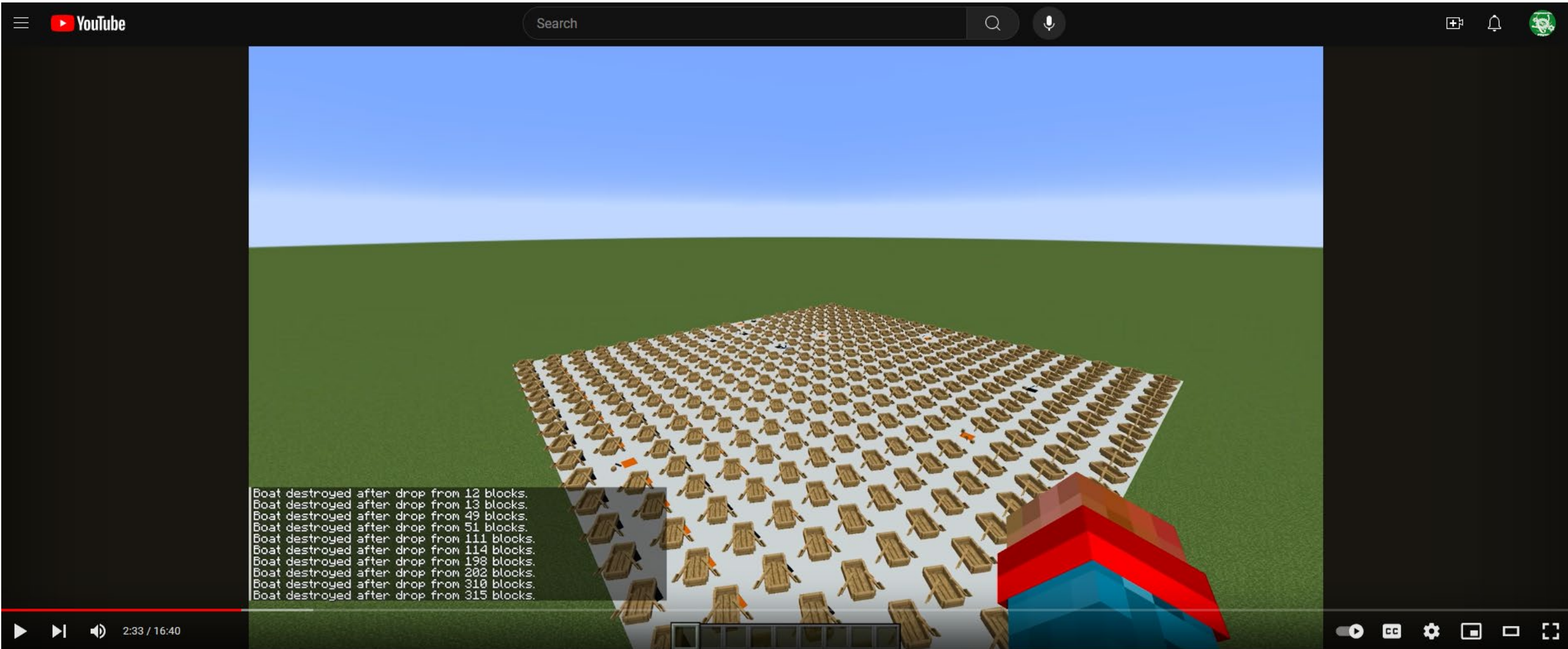
PHYS 351 ERRORS

Dr. Daugherty
Abilene Christian University

Warm-up

Minecraft Boat Drop Mystery

<https://www.youtube.com/watch?v=ei58gGM9Z8k>



The Minecraft boat-drop mystery



Stand-up Maths
1.21M subscribers

Subscribe

65K



Share

Clip

Save



All

From Stand-up Maths

Minecraft

Polit



Code Analysis (tentative):

net.minecraft.world.entity.vehicle.Boat.java

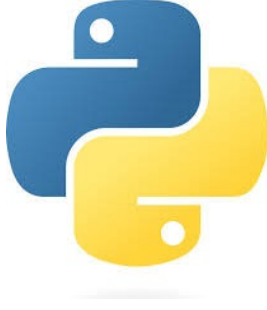
```
public class Boat extends Entity {
    ...
    protected void checkFallDamage(double p_38307_, boolean p_38308_, BlockState p_38309_, BlockPos p_38310_) {
        this.lastYd = this.getDeltaMovement().y;
        if (!this.isPassenger()) {
            if (p_38308_) {
                if (this.fallDistance > 3.0F) {
                    if (this.status != Boat.Status.ON_LAND) {
                        this.resetFallDistance();
                        return;
                    }

                    this.causeFallDamage(this.fallDistance, 1.0F, DamageSource.FALL);
                    if (!this.level.isClientSide && !this.isRemoved()) {
                        this.kill();
                        if (this.level.getGameRules().getBoolean(GameRules.RULE_DOENTITYDROPS)) {
                            for(int i = 0; i < 3; ++i) {
                                this.spawnAtLocation(this.getBoatType().getPlanks());
                            }

                            for(int j = 0; j < 2; ++j) {
                                this.spawnAtLocation(Items.STICK);
                            }
                        }
                    }
                }

                this.resetFallDistance();
            } else if (!this.level.getFluidState(this.blockPosition().below()).is(FluidTags.WATER) && p_38307_ < 0.0D) {
                this.fallDistance = (float)((double)this.fallDistance - p_38307_);
            }
        }
    }
    ...
}
```

Daily Work - Python Practice



Gravity in Minecraft: $g = 0.04 \frac{\text{blocks}}{\text{tick}^2}$ (You don't need this but $1 \text{ tick} = \frac{1}{20} \text{ sec}$)

Algorithm for falling objects:

1) $v_{\text{new}} = v_{\text{old}} - g$ (remember that Δt is always 1)

2) $y_{\text{new}} = y_{\text{old}} + v_{\text{new}}$

time	v	y
0	0	0
1	$-g$	$-g$
2	$-2g$	$-3g$

Questions:

- How many ticks to fall one block
- The first two boat breaks occur somewhere between 10-15 ticks. Find something weird that happens here
- Predict the next weird height

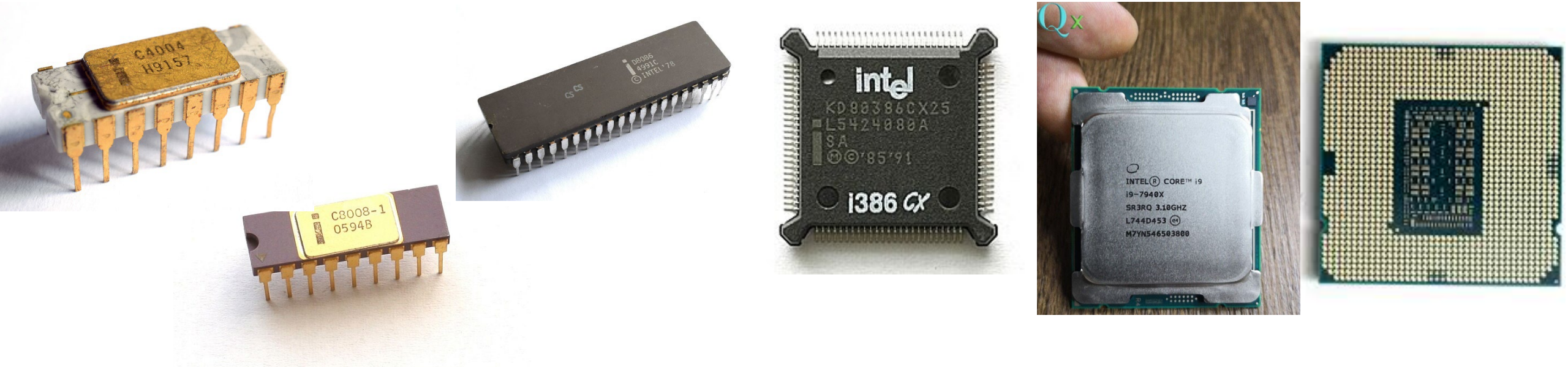
How Numbers Are Stored

MICROPROCESSORS

TAYLOR SWIFT THE ERAS TOUR



# Bits	Dates (approx.)
4	1971
8	1972
16	1973-1980
32	1980-2003
64	2003- ∞



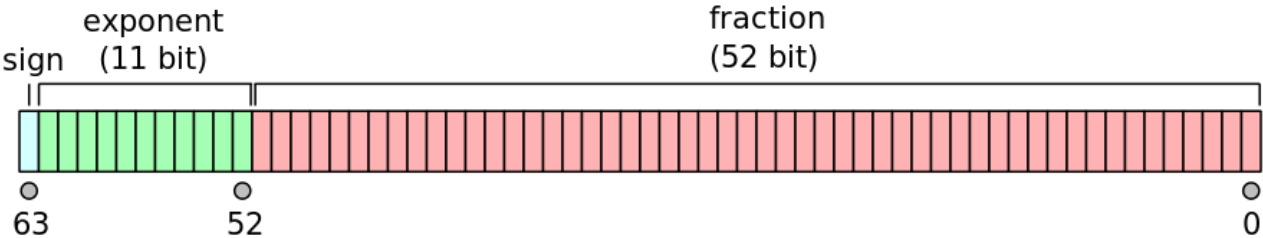
Name	Date	Bits	Cost (adjusted)	Transistors	Notes
Intel 4004	1971	4	\$450	2,300	First commercial microprocessor
Intel 8008	1972	8	\$874	3,500	
Intel 8086	1978	16	\$420	29,000	
Intel iAPX 432	1981	32	\$1,250 - \$3,500	97,000	Commercial flop
Intel i386	1985 – 2007!	32	\$875 - \$1,750	275,000	
Intel Itanium	2001	64	?	~200,000,000	
...					
Core i9	March 2024	64	\$689	4,200,000,000	

$$(-1)^s \times (1.F) \times 2^{E-Bias}$$

sign

fraction

exponent



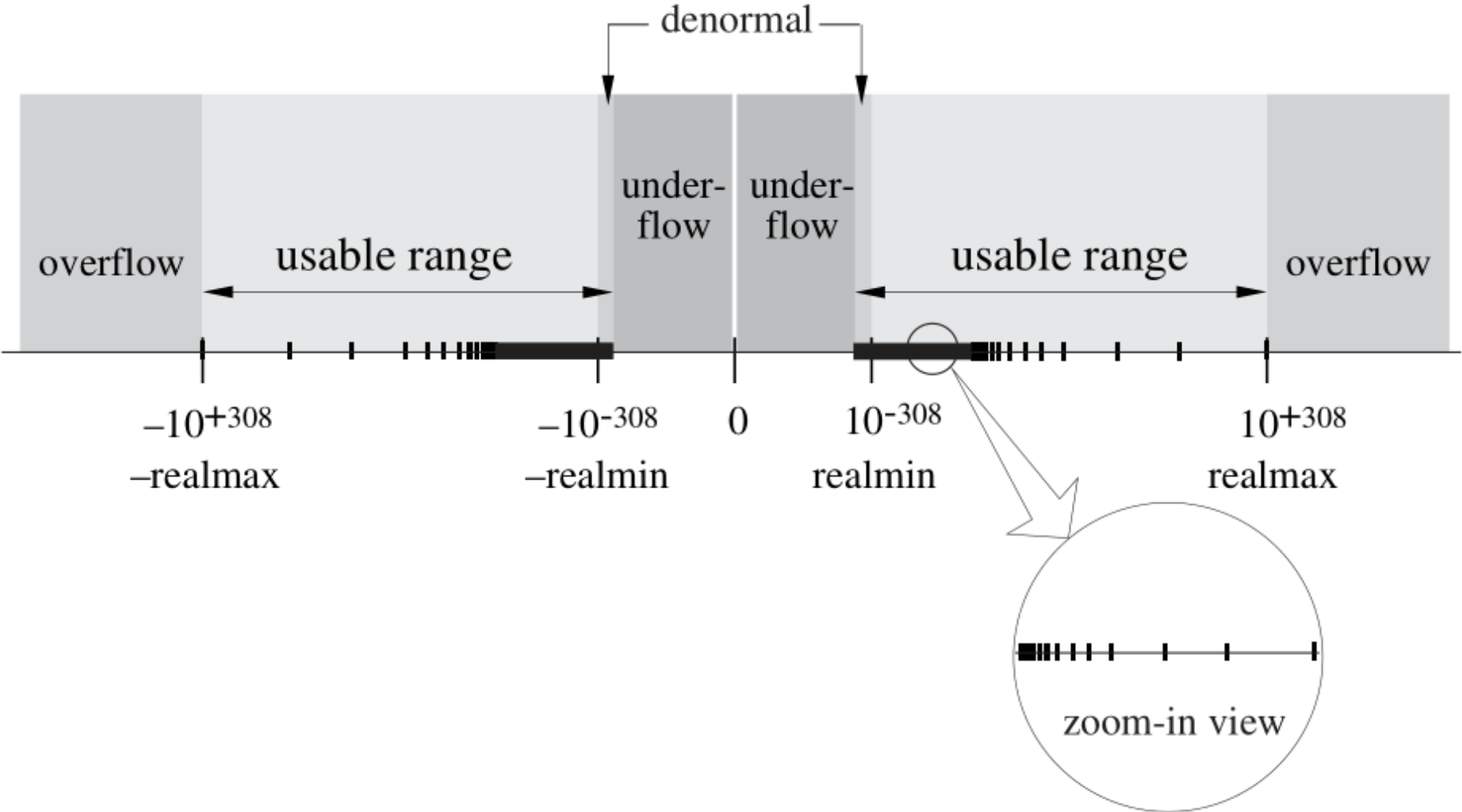
Values:

	# bits	Notes
Sign	1	0 pos, 1 neg
Fraction	52	ranges from 0 to 0.999999... 4 quadrillion options
Exponent	11	range from 0-2046
Bias		1023 for 64 bits, E-Bias from -1023 to +1023

Special Cases:

E	F	Usage
0	≠ 0	Subnormal 1.F → 0.F
0	0	Zero (signed)
2047	0	Infinity (signed)
2047	≠ 0	NaN

Floating Point Number Line



Practice

<https://www.h-schmidt.net/FloatConverter/IEEE754.html>

This demo is for 32-bit numbers, not 64. The bias is 127

IEEE-754 Floating Point Converter

Translations: [de](#)

This page allows you to convert between the decimal representation of a number (like "1.02") and the binary format used by all modern CPUs (a.k.a. "IEEE 754 floating point").

IEEE 754 Converter, 2024-02

	Sign	Exponent	Mantissa
Value:	+1	2^1	$1 + 0.5707986354827881$
Encoded as:	0	128	4788206
Binary:	<input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>

Decimal Representation

Value actually stored in float:

Error due to conversion:

Binary Representation

Hexadecimal Representation

1

-1

EXAMPLE How is 6.5 stored?

$$\underbrace{(-1)^s}_{\text{sign}} \times \underbrace{(1.F)}_{\text{fraction}} \times \underbrace{2^{E-Bias}}_{\text{exponent}}$$

STEP 1) Pretend $F = 0$, find value of exponent

- Value is $(1.0) \times 2^{E-Bias}$, make as large as possible without going over
- Since $2^2 = 4$ and $2^3 = 8$, I set the exponent so that $E - Bias = 2$
- Call our number $N = 6.5$, then $E - Bias = \lfloor \log_2 N \rfloor$
- $Bias = 1023$, so $E = 1025$

STEP 2) Solve for F

- $(1.F) = \frac{N}{2^{E-Bias}} = \frac{6.5}{4} = 1.625$, therefore $F = 0.625$
- Find bits for F , turn on every bit you can without going over
- $F = \frac{1}{2} + \frac{1}{8}$

Final Answer: $6.5 = (1.625) \times 2^2$

EXAMPLE How is 6.5 stored?

$$\underbrace{(-1)^s}_{\text{sign}} \times \underbrace{(1.F)}_{\text{fraction}} \times \underbrace{2^{E-Bias}}_{\text{exponent}}$$

IEEE 754 Converter, 2024-02

	Sign	Exponent	Mantissa
Value:	+1	2^2	$1 + 0.625$
Encoded as:	0	129	5242880
Binary:	<input type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Decimal Representation	<input type="text" value="6.5"/>		
Value actually stored in float:	<input type="text" value="6.5"/>		
Error due to conversion:	<input type="text" value="0"/>		
Binary Representation	<input type="text" value="01000000110100000000000000000000"/>		
Hexadecimal Representation	<input type="text" value="40d00000"/>		

1

-1

PROBLEM: How is 0.04 stored?

$(-1)^s \times (1.F) \times 2^{E-Bias}$

sign

fraction

exponent

IEEE 754 Converter, 2024-02

	Sign	Exponent	Mantissa
Value:	+1	2 ⁻⁵	1 + 0.27999999713897705
Encoded as:	0	122	2348810
Binary:	<input type="checkbox"/>	<input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
Decimal Representation	<input type="text" value="0.04"/>		
Value actually stored in float:	<input type="text" value="0.0399999999105930328369140625"/>		
Error due to conversion:	<input type="text" value="-0.000000000894069671630859375"/>		
Binary Representation	<input type="text" value="00111101001000111101011100001010"/>		
Hexadecimal Representation	<input type="text" value="3d23d70a"/>		

1

-1

PROBLEM: How is 0.04 stored?

$(-1)^s \times (1.F) \times 2^{E-Bias}$

sign

fraction

exponent

IEEE 754 Converter, 2024-02

	Sign	Exponent	Mantissa
Value:	+1	2 ⁻⁵	1 + 0.27999999713897705
Encoded as:	0	122	2348810
Binary:	<input type="checkbox"/>	<input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
Decimal Representation	<input type="text" value="0.04"/>		
Value actually stored in float:	<input type="text" value="0.0399999999105930328369140625"/>		
Error due to conversion:	<input type="text" value="-0.000000000894069671630859375"/>		
Binary Representation	<input type="text" value="00111101001000111101011100001010"/>		
Hexadecimal Representation	<input type="text" value="3d23d70a"/>		

1

-1


```
1 # Sum up vs sum down
2 N = 1000
3 sumup = 0
4 for i in range(1,N):    # make sure to skip zero!
5     sumup+=1/i
6
7 sumdown = 0
8 for i in range(N-1,0,-1): # counts down from N-1 to 1
9     sumdown +=1/i
10
11 print(sumup)
12 print(sumdown)
13 print(sumup-sumdown)
```

7.484470860550343

7.484470860550341

1.7763568394002505e-15