Let’s get started !

Q: How do people and computers represent numbers ?

Base 10

$$238 = 2 \cdot 10^2 + 3 \cdot 10^1 + 8 \cdot 10^0$$

$$m = \sum_{i=0}^{k-1} a_i \, 10^i$$

$$= (a_{k-1} \; a_{k-2} \; \cdots \; a_2 \, a_1 \, a_0)_{\text{ten}}$$

Base 2

$$11010 = 1\cdot 2^4 + 1\cdot 2^3 + 0\cdot 2^2$$
$$+ 1\cdot 2^1 + 0\cdot 2^0$$
$$= 16 + 8 + 2 = 26$$

$$m = \sum_{i=0}^{n-1} b_i \, 2^i$$
$$= (b_{n-1} b_{n-2} \cdots b_2 b_1 b_0)_{\text{two}}$$

Counting in binary

decimal	binary	binary
0	0	0000
1	1	0001
2	10	0010
3	11	0011
4	100	0100
5	101	0101
6	110	0110
7	111	0111
8	1000	1000
⋮	⋮	

To convert from binary to decimal, you need to know the powers of 2.

n	2^n
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024
⋮	⋮

memorize

How can we convert **m** from decimal to binary?

Idea 1:

Find the biggest power of two less than or equal to m, and subtract it from m.

Repeat until done.

(Requires memorizing powers of 2.)

How can we convert **m** from decimal to binary?

Idea 2: Consider familiar idea from base 10:

$$238 = 230 + 8$$

$$m = (m / 10) \times 10 + m \% 10$$

integer division

Same idea works in base 2.

$$m = (m/2) \times 2 + m \% 2$$

Example:

$m = (100111)_{two}$

$m/2 = 10011$

$m \% 2 = 1$

$$m = \sum_{i=0}^{n-1} b_i \, 2^i$$

$$= (b_{n-1} b_{n-2} \cdots b_2 b_1 b_0)_{two}$$

$$m/2 = \sum_{i=1}^{n-1} b_i \, 2^{i-1}$$

$$= (b_{n-1} b_{n-2} \cdots b_2 b_1)_{two}$$

$$m \% 2 = b_0$$

Algorithm:

given m in decimal, convert it to binary.

$$
\begin{aligned}
& i \leftarrow 0 \\
& \text{while } m > 0 \ \{ \\
& \qquad b_i \leftarrow m \,\%\, 2 \\
& \qquad m \leftarrow m / 2 \\
& \qquad i \leftarrow i + 1 \\
& \}
\end{aligned}
$$

Example

$m = 241 = (11110001)_{two}$

m	b_i
241	
$= 120 \times 2 + 1$	1
60	0
30	0
15	0
7	1
3	1
1	1
0	1

m	b_i
$(11110001)_{two}$	
1111000	1
111100	0
11110	0
1111	0
111	1
11	1
1	1
	1

Q: How to add two numbers in binary ?

Addition (base 10)

```
  1 0 1      ← carry
  2 3 4 3
+ 5 8 1 9
---------
  8 1 6 2
```

You need to memorize single digit sums to do this.

Addition (base 2)

$$\begin{array}{cccccccccc}
0 & 0 & 1 & 1 & 0 & 1 & 0 & & & \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & \leftarrow & 26 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & \leftarrow & 27 \\
\hline
0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & \leftarrow & 53
\end{array}$$

Subtraction (base 10)

$$\begin{array}{r} 2343 \\ -\;5819 \\ \hline \end{array}$$

ASIDE: the grade school algorithm doesn't work when the bigger number is on the bottom. To take the difference using the grade school algorithm, you put the bigger number on top and take the negative of the result.

$$a - b$$

$$= a + (-b)$$

Next class we will learn how to represent *negative numbers in binary* which allows us to perform this sum.