

# Addition / Subtraction

ICS<sub>3</sub>U  
Mr. Emmell

Try these, in decimal

$$\begin{array}{r} 5 \\ + 3 \\ \hline \end{array}$$

Try these, in decimal

$$\begin{array}{r} 5 \\ + 3 \\ \hline 8 \end{array}$$

Try these, in decimal

$$\begin{array}{r} 8 \\ + 4 \\ \hline \end{array}$$

Try these, in decimal

$$\begin{array}{r} 8 \\ + 4 \\ \hline 12 \end{array}$$

How did you do that?

---

When one column overflowed....  
It incremented the next column



# How did you know it overflowed?

When one column...  
tried to have a higher value than our  
number system allowed!

---

Then you

- Subtracted ten
- Wrote the remainder
- Added the carry

Try these, in octal

$$\begin{array}{r} 2_8 \\ + 4_8 \\ \hline \end{array}$$

Try these, in octal

$$\begin{array}{r} 2_8 \\ + 4_8 \\ \hline 6_8 \end{array}$$

Try these, in octal

$$\begin{array}{r} 5_8 \\ + 5_8 \\ \hline \end{array}$$

Try these, in octal

$$\begin{array}{r} 5_8 \\ + 5_8 \\ \hline \end{array}$$

5 + 5 = 10!!!!  
That's too  
much

This is base 8  
so...

$$10 - 8 = 2$$

Try these, in octal

$$\begin{array}{r} 5_8 \\ + 5_8 \\ \hline 12_8 \end{array}$$



Try these, in octal

$$\begin{array}{r} 15_8 \\ + 5_8 \\ \hline \end{array}$$

Try these, in octal

$$\begin{array}{r} 15_8 \\ + 5_8 \\ \hline 22_8 \end{array}$$

Try these, in octal

$$\begin{array}{r} 35_8 \\ + 46_8 \\ \hline \end{array}$$

Try these, in octal

$$\begin{array}{r} 35_8 \\ + 46_8 \\ \hline 103_8 \end{array}$$



Hexadecimal works the same way!

$$\begin{array}{r} 3_{16} \\ + 6_{16} \\ \hline \end{array}$$

Hexadecimal works the same way!

$$\begin{array}{r} 3_{16} \\ + 6_{16} \\ \hline 9_{16} \end{array}$$

Hexadecimal works the same way!

$$\begin{array}{r} 7_{16} \\ + 6_{16} \\ \hline \end{array}$$

Hexadecimal works the same way!

$$\begin{array}{r} 7_{16} \\ + 6_{16} \\ \hline D_{16} \end{array}$$

Hexadecimal works the same way!

$$\begin{array}{r} B9_{16} \\ + A_{16} \\ \hline \end{array}$$

Hexadecimal works the same way!

$$\begin{array}{r} B9_{16} \\ + A_{16} \\ \hline \end{array}$$

Remember!  
In decimal,  
this is:

$$\begin{array}{r} 9 \\ + \\ 10 \\ = 19 \end{array}$$



Hexadecimal works the same way!

$$\begin{array}{r} \text{B } 9_{16} \\ + \text{A }_{16} \\ \hline \text{C } 3_{16} \end{array}$$

Hexadecimal works the same way!

$$\begin{array}{r} 26_{16} \\ + 7E_{16} \\ \hline \end{array}$$

Hexadecimal works the same way!

$$\begin{array}{r} 26_{16} \\ + 7E_{16} \\ \hline A4_{16} \end{array}$$

Don't forget binary too

$$\begin{array}{r} 1001_2 \\ + 1011_2 \\ \hline \end{array}$$

Don't forget binary too

$$\begin{array}{r} 1001_2 \\ + 1011_2 \\ \hline 10100_2 \end{array}$$

Now... how about subtraction?

---

Good news! We can't... Not really.  
We need to learn about complements first



So instead, we add the complement.

Consider a 12 hour clock.

If it is 11:00 now, then three hours later it will be 2:00

$$11 + 3 = 14 \equiv 2 \pmod{12}$$

2 o'clock



So instead, we add the complement.

Similarly, if it is 1:00,  
then 4 hours ago it was 9  
since

$$1 - 4 = -3 \equiv 9 \pmod{12}$$

9 o'clock



So instead, we add the complement.

Notice that subtracting 4 hours on the clock is the same as *adding* 8 hours  $(12 - 4)$ .

In particular, we could have computed it as follows:

$$1 - 4 \equiv 1 + 8 = 9 \pmod{12}$$

Binary numbers work the same way

Use an 8-bit number as an example

Those numbers range from 0-255

If we have 255 and add one,  
then we get zero because it 'cycles back'.



So....?

How do we find the complement for a number in binary?

We know that  $200 - 50$  is the same as  $200 + (255 - 50)$   
(Because we roll over after 255!)

## HOW TO FIND “TWO’S COMPLEMENT”

- Define how many bits you are using! 8-bit?
  - Always need to have leading zeroes.
- Invert all the bits
- Add one
- BOOM – Two’s complement. Add normally using binary addition.



## Example

$$100 - 58 = 42$$

Now for this operation,  
we are really saying:

$$100 + (-58) = 42$$

Let's convert to binary

## Example

2	100	
2	50	0
2	25	0
2	12	1
2	6	0
2	3	0
2	1	1
	0	1

$$100 = 01100100_2$$

Remember! 8-bits

What about -58?

## Example – Start with 58



$$58 = 00111010_2$$

Remember! 8-bits

What about -58?

## Example

$$\begin{array}{r} 00111010_2 \\ 11000101_2 \\ \hline \phantom{00111010_2} +1 \\ \hline 11000110 \end{array} \quad \begin{array}{l} //58 \\ //FLIP \\ //Add one \end{array}$$

$$-58 = 11000110_2$$

Now we do the 'addition'

$$\begin{array}{r} 01100100_2 \\ \underline{11000110_2} \\ \end{array}$$

Now we do the 'addition'

$$\begin{array}{r} 01100100_2 \\ \underline{11000110_2} \\ 100101010_2 \end{array}$$

If there is a leading one, we drop it (remember! 8-bits)

Answer =  $00101010_2$   
(42 !)



*Note:*

*Rather than having our numbers go from 0 to 255 around a clock, we have them go from -128 to 127*

**NOTE: Use 8-bit representations of every number below**

Complete the following in binary

a)  $00001110_2 + 00100011_2$       b)  $75_{10} + 26_{10}$

Using two's complement

a)  $37_{10} - 37_{10}$       b)  $55_{10} - 18_{10}$       c)  $85_{10} - 65_{10}$   
ex:  $37_{10} + (-37_{10})$

Using two's complement

a)  $-14_{10} + 28_{10}$       b)  $20_{10} - 36_{10}$       c)  $15_{10} - 50_{10}$