NEGATIVE BINARY NUMBERS

01010101010101010101010101010101001011111010
10101010110110

NEGATIVE BINARY NUMBERS

- **In decimal (base 10) we are familiar with placing a "-" sign in front of a number to denote that it is negative**
- **The same is not true for binary numbers as a computer won't understand that**
- **What happens in memory then?**

NEGATIVE BINARY NUMBERS

- **The two most common types are**
  - **Signed magnitude**
  - **Two's complement**

- **Two's complement is the system used in microprocessors**

- **Most significant bit becomes important**

SINGLE MAGNITUDE

- using one BYTE we can use the “left most” bit to represent positive (0) and negative (1).

EXAMPLES

27 = 00011011

-27 = **1**0011011

As you can see the only thing that changes is the left most bit.

Before a BYTE could represent 0 – 255… **What about NOW?**

TO APPRECIATE THE SHORTCUT FOR TWO'S COMPLEMENT…

The MSB (most significant bit) becomes negative. So instead of +128, it becomes -128. The other column weights stay the same.

-128	64	32	16	8	4	2	1

TO APPRECIATE THE SHORTCUT FOR TWO'S COMPLEMENT

Task:

Represent -8 as an 8 bit signed integer using two's complement. The MSB must be 1 to show the number is negative. So, …

1							
-128	64	32	16	8	4	2	1

TO APPRECIATE THE SHORTCUT FOR TWO'S COMPLEMENT

Positive numbers must be added to -128 to bring it to -8.

First step, add 64 …

1	1						
-128	64	32	16	8	4	2	1

The total so far is -128 + 64 = -64

TO APPRECIATE THE SHORTCUT FOR TWO'S COMPLEMENT

Positive numbers must be added to -64 to bring it to -8.

Second step, add 32 …

1	1	1					
-128	64	32	16	8	4	2	1

The total so far is -64 + 32 = -32

TO APPRECIATE THE SHORTCUT FOR TWO'S COMPLEMENT

Positive numbers must be added to -32 to bring it to -8.

Third step, add 16 …

1	1	1	1				
-128	64	32	16	8	4	2	1

The total so far is -32 + 16 = -16

TO APPRECIATE THE SHORTCUT FOR TWO'S COMPLEMENT

Positive numbers must be added to -16 to bring it to -8.

Fourth step, add 8 …

1	1	1	1	1			
-128	64	32	16	8	4	2	1

The total so far is -16 + 8 = -8

TO APPRECIATE THE SHORTCUT FOR TWO'S COMPLEMENT

We're done! We've reached -8!!

1	1	1	1	1	0	0	0
-128	64	32	16	8	4	2	1

We can now add 0's to the end of our new negative binary number

TO APPRECIATE THE SHORTCUT FOR TWO'S COMPLEMENT

8 (base 10) = **00001000** (base 2)
-8 (base 10) = **11111000** (base 2)

LETS TRY ANOTHER ONE

- What is -2 in binary using two's complement?

2 (base 10) = 00000010 (base 2)

LETS TRY ANOTHER ONE

- What is -2 in binary using two's complement?

Solution:

2 (base 10) = 00000010

-2 (base 10) = 11111110 (base 2)

1	1	1	1	1	1	1	0
-128	64	32	16	8	4	2	1

Can you see a pattern here?

TWO'S COMPLEMENT

- You need to flip each bit (1 becomes 0, 0 becomes 1) and then add one to the number.

- Example:

10111000 (184) becomes

01000111 + 1 = *01001000 (-184) in storage*

TWO'S COMPLEMENT

Question:

What is the highest number you can represent using two's complement?

What is the lowest number you can represent using two's complement?

TWO'S COMPLEMENT

Question:

What is the highest number you can represent using two's complement?

-128

What is the lowest number you can represent using two's complement?

127

WHY TWO'S COMPLEMENT

- **Only need one type of hardware/process to add both signed and unsigned numbers.**

- **Instead of calculating 9-8, you can now do 9 + (-8)**

Example:

```
  00001001
+ 11111000
  --------
  00000001
```