Now we know how to represent a positive decimal[1] number in binary[2].

128	64	32	16	8	4	2	1
1	0	1	0	0	1	1	0
							128 + 32 + 4 + 2 = 166

Binary Addition

Binary addition is easy if you remember these rules:

1 + 0 = 1
1 + 1 = 1 0
1 + 1 + 1 = 1 1

Let's add 2 binary numbers (start from the right!!):

0	0	1	1	0	0	1	1
1	0	1	0	0	1	1	0
1	1	0	1	1	0	0	1

According to the above rule, can you see how sometimes a 1 carries over into the next column. Take a moment to figure it out.

CHECK

Add these two numbers together and *compare your answer with a partner*.

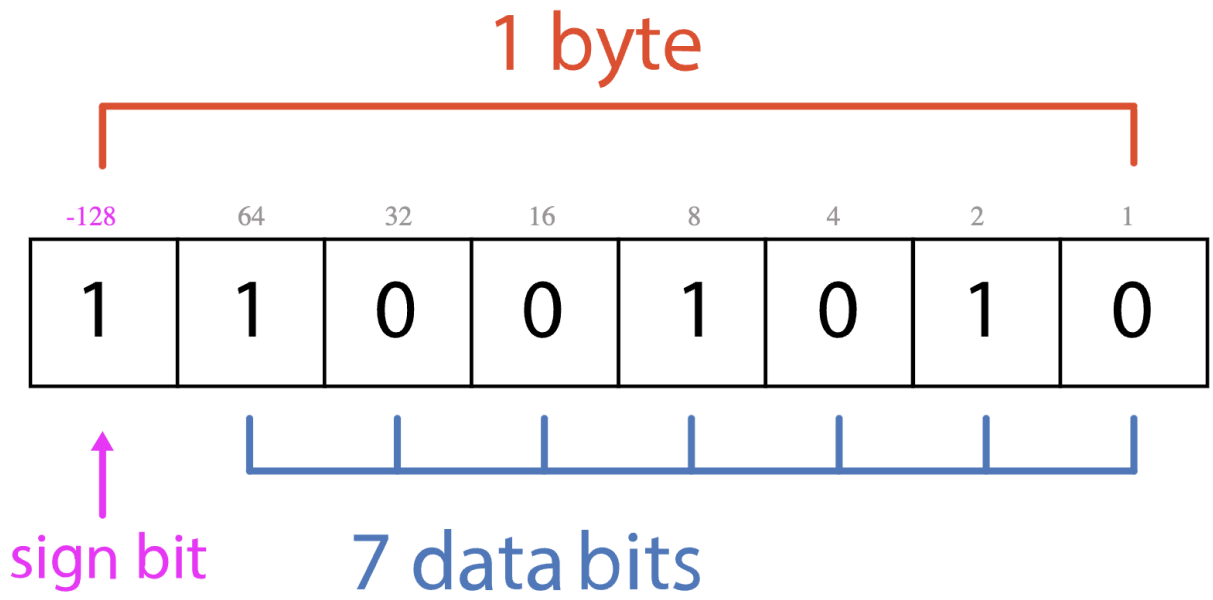
0	0	1	1	1	1	1	1
1	0	1	1	0	0	1	0

[1] Base-10 or Denary
[2] Base-2

Negative numbers

How can we represent negative numbers in binary?

The trick is to reserve the left-most bit as a **sign-bit**.



Look at the binary number above[3].

What do you think the above byte of information is in decimal?

What do you think this byte of information is in decimal?

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

[3] If the sign bit is 1, it means we have a negative number.

Here is a useful trick to convert a negative decimal number into binary. The example we will use is -1

Step 1

Write the number as a positive number, ie 1

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

Step 2

Invert all of the numbers

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

Step 3

Add 1

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

Yes! If we use a sign bit, 11111111 is equal to -1!

This method of representing numbers using a sign bit is known as:

two's complement

CHECK

1. Add these two binary numbers together.

0	**0**	**1**	**1**	**1**	**1**	**1**	**1**
1	**0**	**0**	**1**	**0**	**0**	**1**	**1**

2. Convert your answer to decimal.

Working Area

3. Convert your answer to hexadecimal.

Working Area

4. Using two's complement, find the binary equivalent of -72

5. Add the numbers -127 and 10 together in binary. Use two's complement and show your working.