Set 5. Binary Numbers

Skill 5.01: Explain how the binary number system works
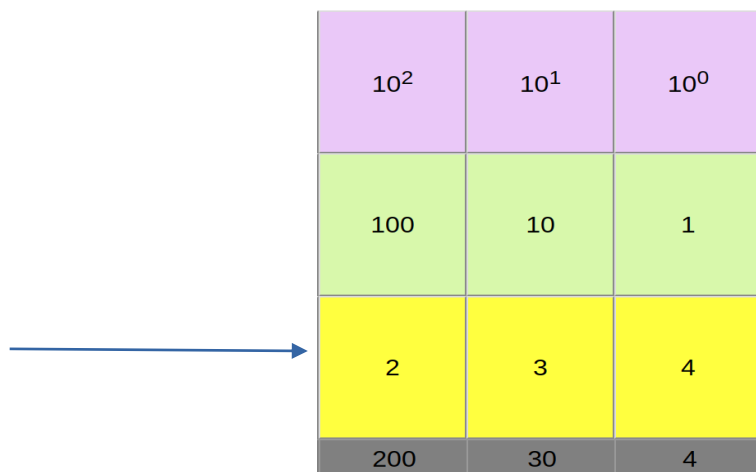
Skill 5.01 Concepts

As humans, we typically represent numbers in the decimal system. Counting to ten is as simple as 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.

As we just learned, computers represent all information in bits. In order to represent numbers with just 0s and 1s computers use the binary number system. Here's what it looks like when a computer counts to ten: 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010.

Decimal Number Refresher

Before exploring how the binary system works, let's revisit our old friend, the decimal system. When you learned how to count, you might have learned that the right-most digit is the "ones' place", the next is the "tens' place", the next is the "hundreds' place", etc.

Another way to say that is that the digit in the right-most position is multiplied by 1, the digit one place to the left is multiplied by 10, and the digit two places to its left is multiplied by 100. The 234 can be visualized as follows,

10^2	10^1	10^0
100	10	1
2	3	4
200	30	4

When we multiply each digit by its place, we can see that 234 is equal to,

(2 x 100) + (3 x 10) + (4 x 1).

We can also think of those places in terms of the powers of ten. The ones' place represents multiplying by 10^0, the tens' place represents multiplying by 10^1, and the hundreds' place represents multiplying by 10^2. Each place we add, we are multiplying the digit in that place by the next power of 10.

Binary numbers

The binary system works the same way as decimal. The only difference is that instead of multiplying the digit by a power of 10, we multiply it by a power of 2.

Let's look at the decimal number 1, represented in binary as 0001:

2^2	2^1	2^0
4	2	1
0	0	1
0	0	1

That's the same as (0 x 8) + (0 x 4) + (0 x 2) + (1 x 1), or 0 + 0 + 0 + 1.

Now let's consider the decimal number 10 which is represented in binary as 1010:

2^3	2^2	2^1	2^0
8	4	2	1
1	0	1	0
8	0	2	1

That's the same as (1 x 8) + (0 x 4) + (1 x 2) + (0 x 1), or 8 + 0 + 2 + 0.

If figuring out how to determine the decimal equivalent of a binary number is confusing at this point, that's to be expected! Next we will explore some techniques that make converting between number systems easy.

 Skill 5.01 Exercise 1

Skill 5.02 Concepts

To convert a decimal number to binary, try the steps below,

1. Grab a piece of paper
2. Draw dashes to represent the bits. The number of dashes you draw depends on how large the number is. For now, just write 8 dashes. This will allow us to represent numbers up to 255.
3. Write the powers of 2 under each dash. Start under the right-most dash, writing 1, then keep multiplying by 2.
4. Now start at the left-most dash and ask: "Is the number greater than or equal to this place value?". If you answer yes, then write a 1 in that dash and subtract that amount from the number. If you answer no, then write a 0 and move to the next dash.
5. Keep going from left to right, keeping track of how much remainder you still need to represent. When you are done, you will have converted the number to binary.

Below is an example that illustrates how to apply the steps above to convert a decimal number to binary.

Example: What is 6 in binary?

In the example below, the top row of 8 boxes represents 8 dashes. The middle row represents the powers of 2. The bottom row is the calculated powers of two.

dashes	0	0	0	0	0	1	1	0
Powers of 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Powers of 2 calculated	128	64	32	16	8	4	2	1

Now, beginning from the LEFT ask: Is 6 equal to or greater than 128? The answer is no, so we write a 0 in the dash. Is 6 equal to or greater than 64? The answer is no, so we write a 0 in the dash. We continuing asking until the answer is yes, at which point we write a 1. Notice this happens when the power of 2 is 4.

Now, we subtract the power of 2 value from 6 and get 2. $(6 - 4 = 2)$. We are not to zero yet, so we keep going. Is 2 greater than or equal to 2? Yes! So we write a 1 when the power of 2 is 2.

Next, we subtract the power of 2 value from 2 and get 0. $(2 - 2 = 0)$.

We are done! The last step is to fill in the remaining bits with zeros.

Skill 5.02 Exercise 1

Skill 5.03 Concepts

In the previous example, you converted odd numbers. There is something interesting about odd numbers in binary. Here are a few more odd numbers to give you an idea.

Decimal	Binary
3	0011
5	0101
7	0111
9	1001

Do you see the pattern? You do not need to convert a binary number to decimal to figure out whether or not a binary number is odd. You only need to look at a single bit of information – the very last bit! The last bit is always the ones' place, and if a number is odd, it must have a 1 in that ones' place. There is not way to create an odd number in the binary system without that ones' place since every other place is a power of 2. Knowing this can give you a better intuitive understanding of binary numbers.

There is another interesting pattern in binary numbers. Take a look at these:

Decimal	Binary
3	11
7	111
15	1111

Each of the decimal numbers are a power of 2, minus 1: 4 – 1 = 3, 8 – 1 = 7, 16 – 1 = 15. When a binary number has a 1 in each of its places, then it will always equal the largest number that can be represented by that number of bits. If you want to add 1 to that number, you need to add another bit. It's like 9, 99, 999 in the decimal system.

As it turns out, the highest number that can be represented by n bits is the same as $2^n - 1$

Bits (n)	Highest number	$(2^n - 1)$
1	1	$(2^1 - 1)$
2	3	$(2^2 - 1)$
3	7	$(2^3 - 1)$
4	15	$(2^4 - 1)$

Skill 5.03 Exercises 1 thru 3