Patterns in binary numbers

In those last two questions, you converted odd numbers. There's 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?

The last bit is always the ones' place, and if a number is odd, it must have a 1 in that ones' place. There's no 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's 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 999, 999999, and 999999999 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	$(2^4 - 1)$

What do you think: what does 11111represent in decimal?

You could calculate that using our strategy from before fairly quickly. However, there's one more strategy, keeping in mind what we just learned: you could count the number of bits (5), calculate $2^5 = 2\times2\times2\times2\times2=32$, and then subtract 1.

All of this is to help you gain a more intuitive understanding of binary.