

# Binary Arithmetic

Kurt Bryan

NAME: \_\_\_\_\_

1. **Powers:** The “powers of 10” are the numbers

$$1, 10, 100, 1,000, 10,000, \dots$$

and so on. Each number in the list is obtained by multiplying  $10 \times 10 \times \dots \times 10$  some number of times. Each number is ten times the previous number in the list.

Write out the corresponding powers of 2 below, up to 256. I’ll get you started!

$$1, 2, \_, \_, \_, \_, \_, \_, \underline{256}.$$

2. **Binary versus Decimal:** When we write a number like 603, what we really mean is

$$(6 \times 100) + (0 \times 10) + (3 \times 1).$$

The “3” is in the “ones” place, the “0” is in the “tens” place, and the “6” is in the “hundreds” place. Larger numbers might have digits in the “thousands” place, and so on. This is called the *base 10* or *decimal system*. And we know how to do arithmetic (add, subtract, multiply, divide) in base 10.

But what’s so special about 10? Why can’t we do arithmetic in, say, base 2? We can! In fact, computers themselves use base 2 arithmetic. Base 2 is often also called “binary” arithmetic.

Here’s what a number looks like when written in base 2 or binary:

$$10011.$$

What this means is

$$(1 \times 16) + (0 \times 8) + (0 \times 4) + (1 \times 2) + (1 \times 1).$$

If you work out the above number you obtain 19. In other words, 10011 in base 2 is the same as 19 in base 10.

In binary the only digits we ever use (or need) are 0 and 1. Here are some numbers written in base 2. Convert each to base 10.

(a) 1 \_\_\_\_\_.

(b) 0 \_\_\_\_\_.

- (c) 10 \_\_\_\_\_.
- (d) 100 \_\_\_\_\_.
- (e) 11 \_\_\_\_\_.
- (f) 101 \_\_\_\_\_.
- (g) 1101 \_\_\_\_\_.
- (h) 10101101 \_\_\_\_\_.

3. Converting from base 10 to binary seems a little trickier, but it's not really hard. Let's convert 77 (base 10) into binary.

Our goal is to write

$$77 = (? \times 64) + (? \times 32) + (? \times 16) + (? \times 8) + (? \times 4) + (? \times 2) + (? \times 1).$$

The question marks will be the digits in the binary expansion of 77. We have to decide what to take for each question mark—either a 0 or a 1. Do you see why I didn't include the powers 128 or higher?

- The question mark in front of 64 should be a “1”. Why? Because we can get “one copy” of 64 out of 77. So we're now down to

$$77 = (1 \times 64) + (? \times 32) + (? \times 16) + (? \times 8) + (? \times 4) + (? \times 2) + (? \times 1).$$

We need to adjust the remaining digits to get “another 13” on the right side (since  $77 = 64 + 13$ ).

- What should the next digit (in the 32's place) be? It can't be a “1”, because then the right side will be too big. So it must be a zero and we have

$$77 = (1 \times 64) + (0 \times 32) + (? \times 16) + (? \times 8) + (? \times 4) + (? \times 2) + (? \times 1).$$

- What should the next digit (in the 16's place) be? It can't be 1, because then the right side will be too big. It must be zero, and so we're down to

$$77 = (1 \times 64) + (0 \times 32) + (0 \times 16) + (? \times 8) + (? \times 4) + (? \times 2) + (? \times 1).$$

- See if you can fill in the remaining digits! Write your answer below, and remember, each digit should be either a 0 or a 1:

$$77 = (1 \times 64) + (0 \times 32) + (0 \times 16) + (\_ \times 8) + (\_ \times 4) + (\_ \times 2) + (\_ \times 1).$$

4. Try writing 19 (base ten) in binary.
5. Write each of the decimal numbers 1 to 16 in binary.