Converting between Decimal (Base 10) and Binary (Base 2)

Base 10 vs. Base 2

We're used to thinking about numbers in base 10. People probably started using base 10 because we have 10 fingers so it's easiest for us to count in groups of 10. In a base 10 (or decimal) number, each column in the number stands for a power of 10. Numbers have a 1's column (= 10^0), 10's column (= 10^1), 100's column(= 10^2), etc. We write numbers with the most significant column (the one with the highest value) at the left. We can determine the value of number by multiplying the digit in the column by how much the column is worth:

```
1000 100  10   1
----------------
   2   0   3   6

Total: 2 * 1000 + 0 * 100 + 3 * 10 + 6 * 1 = 2036
```

We're all used to working with base 10, but because information in a computer is stored in bits (just 0 or 1, so it's kind of like just having two fingers to count on), it's easiest to work in base 2. In a base 2 (or binary) number, each column stands for a power of 2: 2^0 (= 1), 2^1 (= 2), 2^2 (= 4), 2^3 (= 8), etc.

```
16   8   4   2   1
------------------
 1   0   1   1   0

Total: 1 * 16 + 0 * 8 + 1 * 4 + 1 * 2 + 0 * 1 = 22
```

We will consider two methods for converting between decimal and binary numbers, the table method and the division/multiplication by 2 method.

Table Method

Decimal to Binary with a Table

1. Create a table whose leftmost column is the greatest power of 2 less than the number you want to convert. If the number is between 64 and 127, the leftmost column will be 64. If the number is between 128 and 255, the leftmost column will be 128. Each column in the table is a power of 2. The rightmost column is 2^0 (= 1), the next one to the left is 2^1 (= 2), the next is 2^2 (= 4), etc. Instead of thinking about powers of 2, you can just think about doubling each number to get the value for the next column to the left. This will typically look like:

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

2. Start by comparing the number you want to convert to the value leftmost column. If you've set up the table correctly, it should be less than the column value, so put a 1 in the column.
3. Add up the values of all the columns you've put 1's in so far. This is your running total. Take the running total and add the value of the next column. If the result is less than the number you want to convert, put a 1 in this column. If the result is greater than the number you want to convert, put a 0 in this column.
4. Repeat the previous step until all columns are filled.

```
Decimal number: 87

64  32  16   8   4   2   1
--------------------------
 1                                 (64 < 87 ? yes)
     0                             (64 + 32 < 87 ? no)
         1                         (64 + 16 < 87 ? yes)
             0                     (64 + 16 + 8 < 87 ? no)
                 1                 (64 + 16 + 4 < 87 ? yes)
                     1             (64 + 16 + 4 + 2 < 87 ? yes)
                         1         (64 + 16 + 4 + 2 + 1 = 87, yes, done)

Binary number: 1010111
```

```
Decimal number: 105

64  32  16   8   4   2   1
--------------------------
 1                                (64 < 105 ? yes)
     1                            (64 + 32 < 105 ? yes)
         0                        (64 + 32 + 16 < 105 ? no)
             1                    (64 + 32 + 8 < 105 ? yes)
                 0                (64 + 32 + 8 + 4 < 105? no)
                     0            (64 + 32 + 8 + 2 < 105? no)
                         1        (64 + 32 + 8 + 1 = 105, yes, done)

Binary number: 1101001
```

Binary to Decimal with a Table

1. Create a table with enough columns for your binary number. (See instructions in the previous section.)

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

2. Fill in the binary number in the table making sure that the rightmost digit is aligned under 1.

3. For each column containing a 1, take the value of that column (e.g., 64, 32, etc.) and add it to the total.

```
Binary number: 1010111

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

Total: 64 + 16 + 4 + 2 + 1 = 87


Binary number: 1101001

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

Total: 64 + 32 + 8 + 1 = 105
```

Division/Multiplication by 2 Method

Decimal to Binary: Divide by 2

1. Take the decimal number and divide it by two keeping track of the remainder (instead of decimal place). For example, 10 / 2 is 5 r 0. 11 / 2 is 5 r 1. The result for both is 5. The remainder for 10 / 2 is 0 and the remainder for 11 / 2 is 1.
2. Take the result and divide it by two in the same way, always keeping track of the remainder.
3. Repeat Step 2 until you reach a result of 0. Your last step should always look like 1 / 2 = 0 r 1.
4. Read the remainders (all 0 or 1) off in reverse order starting at the bottom with the one you just finished. This is the answer.

```
Decimal number:

87                                 105

87 / 2 = 43 r 1                         105 / 2 = 52 r 1
43 / 2 = 21 r 1                          52 / 2 = 26 r 0
21 / 2 = 10 r 1                          26 / 2 = 13 r 0
10 / 2 =  5 r 0                          13 / 2 =  6 r 1
 5 / 2 =  2 r 1                           6 / 2 =  3 r 0
 2 / 2 =  1 r 0                           3 / 2 =  1 r 1
 1 / 2 =  0 r 1                           1 / 2 =  0 r 1

Result (read remainders bottom to top):

1010111                                 1101001
```

Binary to Decimal: Multiply by 2

1. Start with 0 as the result. Add to 0 the digit from the leftmost column in the binary number.
2. Take the result, multiply it by 2, and add the next digit following left-to-right in the binary number.
3. Repeat Step 2 until you have used all digits in the binary number. The final result is your answer.

```
Binary number:

1010111                               1101001

 0     + 1 =  1                        0     + 1 =   1
 1 * 2 + 0 =  2                        1 * 2 + 1 =   3
 2 * 2 + 1 =  5                        3 * 2 + 0 =   6
 5 * 2 + 0 = 10                        6 * 2 + 1 =  13
10 * 2 + 1 = 21                       13 * 2 + 0 =  26
21 * 2 + 1 = 43                       26 * 2 + 0 =  52
43 * 2 + 1 = 87                       52 * 2 + 1 = 105

Decimal number:

87                                    105
```

Checks

The best way to check your answer is to convert it back and make sure you get what you started with. A few quick checks might also help you catch mistakes:

1. If the decimal number is even, the equivalent binary number should end in 0. If the decimal number is odd, the binary number should end in 1.
2. If the binary number ends in 1, the equivalent decimal number should be odd. If the binary number ends in 0, the decimal should be even.
3. If the decimal number is between 64 and 127, the equivalent binary number should contain 7 digits (ignoring any leading 0's).
4. If the decimal number is between 128 and 255, the equivalent binary number should contain 8 digits (ignoring any leading 0's).