Binary Conversion

Decimal to Binary

1. Write out your decimal number
2. Divide it by 2. Write the quotient below it and the remainder in a column to the right.
3. Repeat step 2 until when you get 0 for your quotient.
4. Now start at the ***bottom** of your list of remainders* and build your binary number from ***left to right***.
5. **Remark on leading zero's**: You can always add any number of leading zeros to the left hand side of your binary number to make it a binary string of more bits without affecting the actual number it represents. For example, **1101001, 01101001, 001101001, 0001101001, …** are the binary representation of the decimal number **105** in 7 bits, 8 bits, 9 bits, 10 bits, ….

Example: Convert the decimal number **105** into the binary number **01101001** as a binary string of 8 bits

	Remainders
105	
52	**1**
26	**0**
13	**0**
6	**1**
3	**0**
1	**1**
0	**1**
0	**0**

Answer is: **105** $_{\text{base 10}}$ = **01101001** $_{\text{base 2}}$

Binary to Decimal

1. Write out your binary number.
2. Below it write out the set position weights.
3. For each '1' in the binary number, write down the associated position weight.
4. Add up the position weights written down in **step 3** to get the decimal number.

Binary	**0**	**1**	**1**	**0**	**1**	**0**	**0**	**1**	
Position	7	6	5	4	3	2	1	0	

Weight	128	64	32	16	8	4	2	1	
Add 'em		**64**	**32**		**8**			**1**	= **105**

What is the largest natural number you can represent using *n* bits?

- **Base 2:** **111 …11** $_{\textbf{base 2}}$
- **Base 10:** $2^{n-1} + 2^{n-2} + \ldots + 2^1 + 2^0$ $_{\textbf{base 10}}$ $= 2^n - 1$

Addition:

- **Base 10:**

```
          7
+         6
=========
         13
```

- **Base 2:**

```
        111
+       110
=========
       1101
```

- **Note that**
  1. **7** $_{\textbf{base 10}}$ = **111** $_{\textbf{base 2}}$
  2. **6** $_{\textbf{base 10}}$ = **110** $_{\textbf{base 2}}$
  3. **13** $_{\textbf{base 10}}$ = **1101** $_{\textbf{base 2}}$