Binary Numbers

Lalith Kumar Reddy
ENIGMA, IIIT B

Computers can not understand plain English. They can only utilize data that is encoded with charges. These charges are represented using 0s and 1s.

Therefore, the 0s and 1s form the binary numbers, which are then decoded by the computers.

To those who do not already know, the binary numbers are stored as bits. Each bit is either a 0 or a 1.
4 bits forms a nibble. 8 bits forms 1 byte.
Here on groups of 1024 units form the next level.

UNITS	Abbreviation	SIZE
KILOBYTE	KB	1024 bytes
MEGABYTE	MB	1024 KB
GIGABYTE	GB	1024 MB
TERABYTE	TB	1024 GB
PETABYTE	PB	1024 TB
EXABYTE	EB	1024 PB
ZETTABYTE	ZB	1024 EB
YOTTABYTE	YB	1024 ZB

NOTE: kb, Kb, kB, KB are all different!

k is kilo as in 1000 units.
b is for bits.

K is kilo as in 1024 units.
B is for bytes.

You should be able to identify each of these without messing up.

If binary numbers are represented with only 0s and 1s, you need to know how to decode them by yourself.

Starting from the right, each digit is represented by power of 2. Everything to the left of the last one will be considered 0s.

128	64	32	16	8	4	2	1
1	0	0	1	1	0	0	1

128+16+8+1
= 153

Therefore, $10011001_2 = 153_{10}$.

For practice, solve for these values.
1. 10101011
2. 01110100
3. 01101001

Addition of two binary numbers is just like normal addition. Only difference is, sum of 1 and 1 will give a 0 and a carry of 1.

```
      +1  +1  +1  +1
    1   0   0   1   1   1   1   0
+   0   0   1   1   0   1   0   0
    -----------------------------
    1   1   0   1   0   0   1   0
```

Just the same way, we can do subtraction. But if you transfer from the previous number, it is going to be like adding two to the zero.

```
      -1  +2-1 +2
    1   0   0   1   1   1   1   0
-   0   0   1   1   0   1   0   0
    -----------------------------
    0   1   1   0   1   0   1   0
```

P.S. Don't worry about representation of negative numbers yet. You do not need that as of now.

For practice, solve these sums.
1. 10101011 + 01110100
2. 01101001 - 00110111
3. 11110000 - 01010101 + 00010011

All this is fine. Can you convert a decimal number to a binary number? Don't even think of subtracting powers of two from the number and writing it down. There is a faster and simpler way.

Say I want to reduce the number 105 into a binary number. Use the division method step by step.

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

First, we need to divide the number by two and write the remainder to its right.
For example, here 105/2 = 52, rem = 1
52/2 = 26, rem = 0

Continue this until we reach a one. And then, we write the number backwards.

Therefore, 105_{10} = 1101001 = 0110 1001_2

It is a good practice to write the numbers in groups of four. It is easier to read.

To get a grip in converting, try converting these into binary.
1. 78
2. 431
3. 1153

And we are done with the basics of binary numbers. You can try out more exercises if you wish.