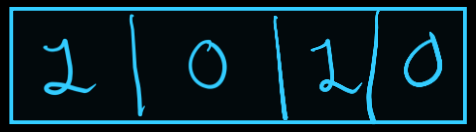
Day 17 - Number System

Decimal To binary

10 10 → 10

1) 

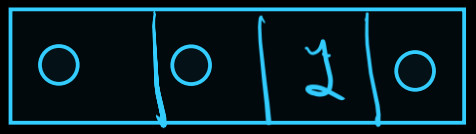
⇒ 0

num & 1 ⇒

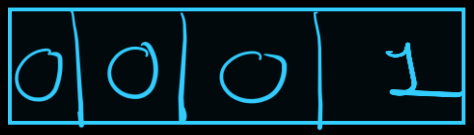
num = num >> 1

2) 

→ 1

3) 

→ 0

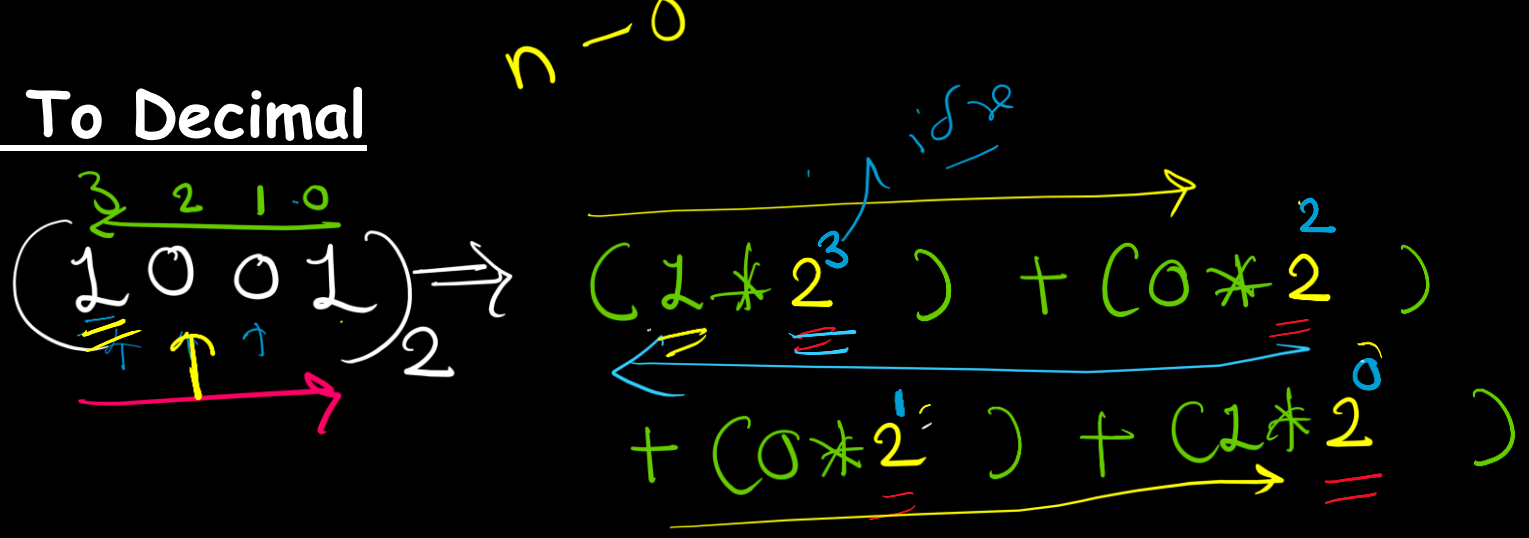
4) 

→ 1

Intution

1. Get the last bit of number (num & 1)
2. Left Shift by 1
3. Repeat the step until number won't become '0'

Binary To Decimal



$= (1*8) + (0*4) + (0*2) + (1*1)$

$= 8 + 0 + 0 + 1$

$(1001)_2 = (9)_{10}$

1) how find power.

↳ Math.pow(2, 1)

num power

2) int number = 0

3) Iterate on each character of given Binary string

↳ multiply each char by 2^{x}

4) What is the value of x

a) left to right

decrease it by 1 ⟷ x = binary_string.length() - 1

b) right to left:

x = 0 → increase by 1

(1010)$_2$ / (9)$_{10}$ / (294)$_8$ / (A15)$_{16}$

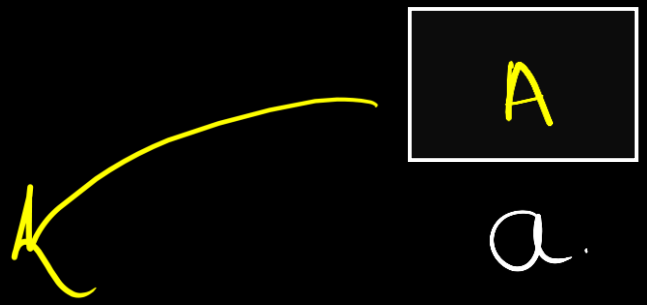
↓ binary → Decimal → Octal → Hex

Ascii values.

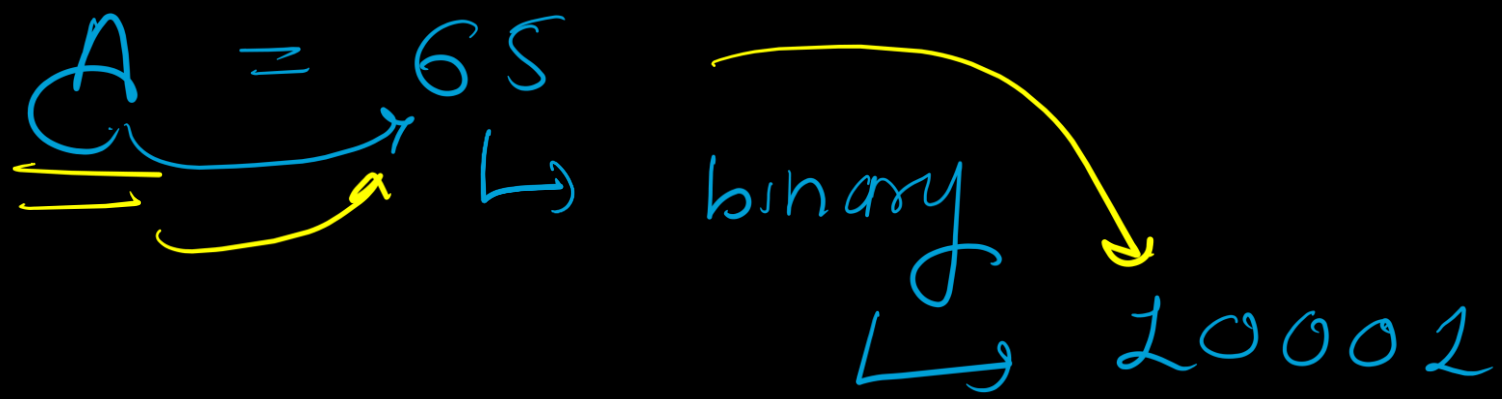
num : i	Ascii
0	48
1	49
2	50
⋮	⋮
9	57

→ char store

Char a = 'A'



0 — 127 ⇒ Ascii value



0	0	_	1	0	0	0	2

⇒

0

0 = 48 = 48 - 48 = 0

1 = 49 = 49 - 48 = 1

2 = 50 = 50 - 48 = 2

9 = 57 = 57 - 48 = 9

```java
Musaddik *
public class BinaryToDecimal {
    Musaddik *
    public static void main(String[] args) {
        String s = "1111";
        // left to right
//        int x = s.length() - 1;

        /** for(int i = 0; i < s.length(); i++){
            int bit = s.charAt(i) - '0';
            number += (bit * Math.pow(2, x));
            x = x - 1;
        }
        */
        int number = 0;
        int x = 0;
        for(int i = s.length() - 1; i>= 0; i--){
            int bit = s.charAt(i) - '0';
            number += (bit * Math.pow(2, x));
            x++;
        }
        System.out.println(number);
    }
}
```

Decimal to Octal

$(98)_{10} = (?)_8$

string s = num%8
num = num/8

number	quotient	Remainder
98/8	12	2
12/8	1	4
1/8	0	1

$(142)_8$

$(98)_2 = (142)_8$

```java
Musaddik *
public class BinaryToDecimal {
    Musaddik *
    public static void main(String[] args) {
        String s = "1111";
        // left to right
//        int x = s.length() - 1;

        /** for(int i = 0; i < s.length(); i++){
            int bit = s.charAt(i) - '0';
            number += (bit * Math.pow(2, x));
            x = x - 1;
        }
        */
        int number = 0;
        int x = 0;
        for(int i = s.length() - 1; i>= 0; i--){
            int bit = s.charAt(i) - '0';
            number += (bit * Math.pow(2, x));
            x++;
        }
        System.out.println(number);
    }
}
```

Octal to Decimal

$$(142)_8 = (?)_{10}$$

$$= (2 * 8^2) + (4 * 8^1) + (2 * 8^0)$$

$$= (2 * 64) + (4 * 8) + (2 * 1)$$

$$= 64 + 32 + 2$$

$$= (98)_2$$

```java
public class OctalToDecimal {
    new *
    public static void main(String[] args) {
        String s = "142";
//          left to right
        int x = s.length() - 1;
        int number = 0;
        for (int i = 0; i < s.length(); i++) {
            int bit = s.charAt(i) - '0';
            number += (bit * Math.pow(8, x));
            x = x - 1;
        }
        System.out.println(number);

    }
}
```

Decimal to hexadecimal

$(256)_{10} = (0)_{?}$

number	Quotient	Remainder
$256/16$	16	0
$16/16$	1	0
$1/16$	0	1

100 Hex

$(256)_{10} \Rightarrow (100)_{16}$

$(1228)_{10} \Rightarrow (?)_{16}$

Number	Quotient	Remainder
$1228/16$	76	$12 \Rightarrow C$
$76/16$	4	$12 \Rightarrow C$
$4/16$	0	$4 \Rightarrow 4$

4CC

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

0 → 9, 10, 11, 12, 13, 14, 15

$(1228)_{16} = 4CC$

4CC ⟹ number ✗

↳ string

10, 11, 12, 13, 14, 15

A, B, C, D, E, f

65, 66 67 68, 69. 70

A) 65 + 10 - 10 = 65 = B

65 + 11 - 10 = 66 ⇒ B

13 ⇒ 65 + 13 - 10 = 68

D

```java
public class DecimalToHex2 {
    new *
    public static void main(String[] args) {
        int num = 1228;
        String  hex = "";
        while (num != 0){
            int bit = num % 16;
            if(bit <= 9){
                hex += bit;
            }
            else{
                hex += (char)('A' + bit - 10);
            }
            num /= 16;
        }
        for (int i = hex.length()-1; i >= 0; i--){
            System.out.print(hex.charAt(i));
        }
    }
}
```

Hexadecimal to Decimal

$$(4CC)_{16} = (4 * 16^2) + (C * 16^1) + (C * 16^0)$$

$$(4 * 16^2) + (12 * 16^1) + (12 * 16^0)$$

$$(4 * 256) + (12 * 16) + (12 * 1)$$

$$1024 + 192 + 12$$

$$= (1228)_{10}$$

	Hex	Ascii
A =	10	65
B =	11	66
C =	12	67
D =	13	68
E =	14	69
F =	15	70

$67 - 65 = 2 + 10 = 12$

('C' − 'A' + 10)

(D − A + 10) = ?

(68 − 65 + 10) = 13

```java
public class HexToDecimal {
    new *
    public static void main(String[] args) {
        String hex = "4CC";
        int x = hex.length()-1;
        int num = 0;
        for(int i = 0; i < hex.length(); i++){
            char ch = hex.charAt(i);
            int bit = 0;
            if(ch >= '0' && ch <= '9'){
                bit = ch - '0';
            }
            else{
                bit = (int)((ch - 'A') + 10);
            }
            num  += (bit * Math.pow(16, x));
            x = x - 1;
        }
        System.out.println(num);
    }
}
```