1.4 Number System

Made by:-

Aditya Jain

Number System

Method to represent numeric values or quantities using digits.

0 1 2 3 4 5 6 7 8 9

Decimal System → Everyone uses this

→ The decimal number system has base 10
→ It uses digits from 0 → 9
→ Base : It is the number of symbols (digits) a number system uses.

Binary Number System → CPU & memory uses this

→ Number system using base 2
→ It uses only 2 digits i.e. 0 & 1

Decimal to Binary Conversion

Approach -1 Division Method

- → Divide number by 2
- → Store reminder (that will be a bit in binary number)
- → Repeat above steps with the quotient until quotient is less than 2
- → Reverse the bit so obtained

for n=10;

Division	Reminder
$10/2 = 5$	0
$5/2 = 2$	1
$2/2 = 1$	0
$1/2 = 0$	1

↑ Read reverse

$$(10)_{10} \Rightarrow (1010)_2$$

How to visualize

$$\overset{1}{2^3} + \overset{0}{2^2} + \overset{1}{2^1} + \overset{0}{2^0}$$

$$8 + 0 + 2 + 0 \Rightarrow 10$$

Approach -2 Bitwise

num = 5; => 101 (Binary representation)

num & 1 < 1 → odd
0 → even

```
n ! = 0          ans =0;
   ⇓
bit = n & 1 ;
ans = (10^i × bit) + ans;
n >> 1 ;
```

How to store ? → 101

↳ 1

↳ 01

↳ 101

for reverse

$$\boxed{ans = (10^i \times digit) + ans}$$

ans = 0;

ans = $(10^0 \times 1) + 0$ => 1

ans = $(10^1 \times 0) + 1$ => 1

ans = $(10^2 \times 1) + 1$ => $\boxed{101}$

```
while (n!=0)
{
    bit = n & 1;
    n = n >> 1;
}
```

---

for Ex 1, 2, 3 → 123

int ans = 0;

ans = $1 \times 10^0 + 0$ => 1

ans = $2 \times 10^1 + 1$ => 21

ans = $3 \times 10^2 + 21$ => $\boxed{321}$ → But we have to reverse this

1, 2, 3 ⇒ 123

ans = 0; $\boxed{ans = (ans \times 10) + digit}$

ans = 0 × 10 + 1 ⇒ 1

ans = 1 × 10 + 2 ⇒ 12

ans = 12 × 10 + 3 ⇒ $\boxed{123}$

```
int n;
cin >> n;

int ans = 0;
 int i = 0;
while (n != 0) {
      int bit = n & 1;
      ans = ( bit * pow(10, i) + ans;
       n = n >> 1;
       i++;
}
cout << ans;
```

if we give 100000 i/p then they throw garbage value

Bcoz int has $[-2^{31}, 2^{31}-1]$ range

To resolve this we have to store in string/ array

for -ve number

n = -6 → -ve ignore

→ n = 6

↓

2's compliment → 1's compliment +1

↓

00000110 → 11111010 ⇒ -6

Binary to Decimal

$$\begin{array}{ccccc} 1 & 0 & 1 & 0 & 1 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \end{array} \Rightarrow ?$$

Ignore the 0 values

$$2^4 + 2^2 + 2^0 \Rightarrow 16+4+1 \Rightarrow 21$$

Checking correct or not

Division	Rem
$21/2 \rightarrow 10$	1
$10/2 \rightarrow 5$	0
$5/2 \rightarrow 2$	1
$2/2 \rightarrow 1$	0
$1/2 \rightarrow 0$	1

$$\Rightarrow (10101)_2 \Rightarrow (21)_{10}$$

```
   n = 110 ;
   int i=0; ans =0;
 while (n!=0)
{
     int digit = n %10;
     if ( digit ==1)
     {
         ans = ans + pow(2,i);
     }
     n = n/10;
     i++;
 }
 cout << ans ;
```