Decimal Number System {0, 1, 2 ... 9} base → 10

342 → 300 + 40 + 2 → $3*10^2 + 4*10^1 + 2*10^0$

2563 → 2000 + 500 + 60 + 3 → $2*10^3 + 5*10^2 + 6*10^1 + 3*10^0$

Binary Number System {0, 1} (bits)

0, 1, 10, 11, 100, 101 ...

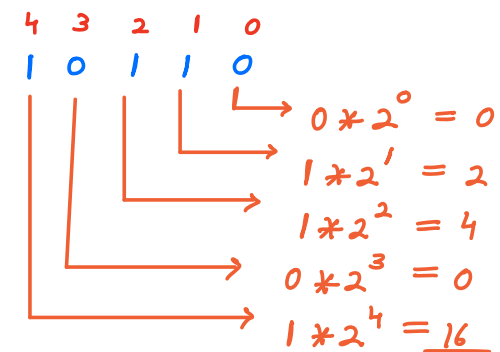
0 1 2 3 4 5 ...

110 → $1*2^2 + 1*2^1 + 0*2^0 = 4+2 = 6$

1011 → $1*2^3 + 0*2^2 + 1*2^1 + 1*2^0 = 8+2+1 = 11$

3 2 1 0

Binary to Decimal

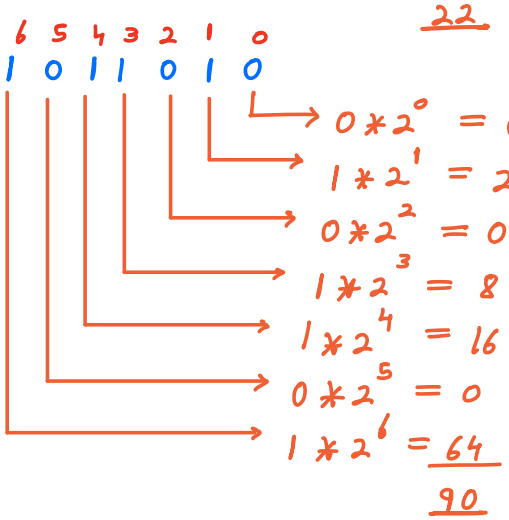


$(10110)_2 = (22)_{10}$

4 3 2 1 0

1 0 1 1 0

$2^4 + 2^2 + 2^1 = 16+4+2 = 22$



$(1011010)_2 = (90)_{10}$

6 5 4 3 2 1 0

1 0 1 1 0 1 0

$2^6 + 2^4 + 2^3 + 2^1$

$= 64 + 16 + 8 + 2 = 90$

Decimal to Binary

2	20	0
2	10	0
2	5	1
2	2	0
2	1	1
	0	

4 3 2 1 0
1 0 1 0 0
$2^4 + 2^2$
$= 16 + 4 = 20$

2	45	1
2	22	0
2	11	1
2	5	1
2	2	0
2	1	1
	0	

5 4 3 2 1 0
1 0 1 1 0 1
$2^5 + 2^3 + 2^2 + 2^0$
$= 32 + 8 + 4 + 1$
$= 45$

Addition

carry

```
  1 1
  3 6 8
+ 4 5 3
-------
  8 2 1
```

```
  2 1 0
  1 0 1  → 5
+ 0 1 0  → 2
-------
  1 1 1  → 7
```

2 → 10 3 → 11

```
  1 1 1
  0 1 0 1  → 5
+ 0 1 1 1  → 7
-------
  1 1 0 0  → 12
```

4 3 2 1 0
1 0 1 1 0
$2^4 + 2^2 + 2^1$
$= 16 + 4 + 2 = 22$

```
    1 1
  1 0 1 1 0  → 22
+ 0 0 1 1 1  → 7
-------------
  1 1 1 0 1  → 29
  4 3 2 1 0
```

```
  2
  1 8
  2 9
  3 6
  8 3
```

1 + 1 = 10
1 + 1 + 1 = 11

3 + 8 + 10 + 6
11 / 21 / 27

Bitwise Operations

{ NOT, AND, OR, XOR, left shift, Right shift }
~/! & | ^ << >>

1 → 0
0 → 1

A	B	A & B	A | B	A ^ B
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

0 → false (unset bit)
1 → true (set bit)

XOR → addition without carry
same same puppy shame

If all bits are 1 ⇒ ans = 1
else ans = 0

If any one bit is 1 ⇒ ans = 1
else ⇒ ans = 0

NAND → ~ (AND)
NOR → ~ (OR)

Numbers

a = 5 → 1 0 1
b = 6 → 1 1 0
a & b → 1 0 0 → 4
(2 1 0)

a = 20 → 0 1 0 1 0 0
b = 45 → 1 0 1 1 0 1
a | b → 1 1 1 1 0 1 → 61
(5 4 3 2 1 0)

a = 20 → 0 1 0 1 0 0
b = 45 → 1 0 1 1 0 1
a ^ b → 1 1 1 0 0 1 → $2^5 + 2^4 + 2^3 + 2^0 = 32 + 16 + 8 + 1 = 57$
(5 4 3 2 1 0)

Properties

1) A & 1 = ?

A = 12 → 1 1 0 0
1 → 0 0 0 1
A & 1 → 0 0 0 0 → 0

A = 13 → 1 1 0 1
1 → 0 0 0 1
A & 1 → 0 0 0 1 → 1

A & 1 → 0 ⇒ last bit is unset for A ⇒ A is even
A & 1 → 1 ⇒ last bit is set for A ⇒ A is odd
(last bit = 0th bit)

(4 3 2 1 0)
1 0 1 1 0
Even + Even → Even

(4 3 2 1 0)
1 0 1 1 1
Even + Odd → Odd

---

2) A & 0 = 0

A → 1 0 1
0 → 0 0 0
A & 0 → 0 0 0

3) A & A = A

A → 1 0 1
A → 1 0 1
A & A → 1 0 1

4) A | 0 = A

A → 1 0 1
0 → 0 0 0

A → 1 0 1
A → 1 0 1

5) $A | A = A$ — $A|0 \rightarrow 101$ — $A|A \rightarrow 101$

6) $A \wedge 0 = A$ — $A \rightarrow 101$, $0 \rightarrow 000$, $A^0 \rightarrow 101$ — $A \rightarrow 101$, $A \rightarrow 101$, $A^A \rightarrow 000$

7) $A \wedge A = 0$

---

8) Commutative Property

$a \& b = b \& a$
$a | b = b | a$
$a \wedge b = b \wedge a$

$a \& b \& c$ (x) — $c \& a \& b$ (x)

$c \& b \& a$
$a \& c \& b$
$b \& a \& c$
$b \& c \& a$
} same

$5 \& 7 \& 4$ → 5, 4

$5 \rightarrow 101$
$7 \rightarrow 111$
$5 \& 7 \rightarrow 101$

101
100
100

---

9) Associative Property

$(a \& b) \& c = a \& (b \& c)$
$(a | b) | c = a | (b | c)$
$(a \wedge b) \wedge c = a \wedge (b \wedge c)$

Q → $a \wedge b \wedge a \wedge d \wedge b = a \wedge a \wedge b \wedge b \wedge d$

$0 \wedge 0 \wedge d = d$

$1 \wedge 3 \wedge 5 \wedge 1 \wedge 2 \wedge 5 \wedge 3 = 1 \wedge 1 \wedge 3 \wedge 3 \wedge 5 \wedge 5 \wedge 2$

$0 \wedge 0 \wedge 0 \wedge 2 = 2$

Q → Given an integer array of size N s.t all elements appear twice except for one element which appear once.
Find the unique element.

$A = [6 \ 9 \ 6 \ 10 \ 9]$ Ans = 10

A = [2 3 5 6 3 6 2] Ans = 5

$6 \wedge 9 \wedge 6 \wedge 10 \wedge 9 = (6 \wedge 6) \wedge (9 \wedge 9) \wedge 10$

$0 \wedge 0 \wedge 10 = 10$

```
ans = A[0]
for i → 1 to (N-1)
    ans = ans ^ A[i]
return ans
```

TC = O(N)

SC = O(1)

Left Shift (<<)

int → 4 Bytes = 32 bits

→ 8 bit number

discard | 7 6 5 4 3 2 1 0

a = 22 → 0 0 0 1 0 1 1 0 → 22

(a << 1) → 0 0 0 1 0 1 1 0 0 → 44

(a << 2) → 0 0 0 1 0 1 1 0 0 0 → 88

(a << 3) → 1 0 0 0 1 0 1 1 0 0 0 0 → 176

176 * 2 = 352

(a << 4) → 0 1 1 0 0 0 0 0 → 352 − 256 = 96

$2^6 + 2^5 = 64 + 32 = 96$ ✓

$2^8 \rightarrow 256$

*2 for every left shift

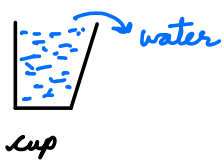
$a << n = a * 2^n$ ✓

$1 << n = 2^n$

1 1 1 1 1 1 1 1 → $2^7 + 2^6 + 2^5 + \dots 2^0$

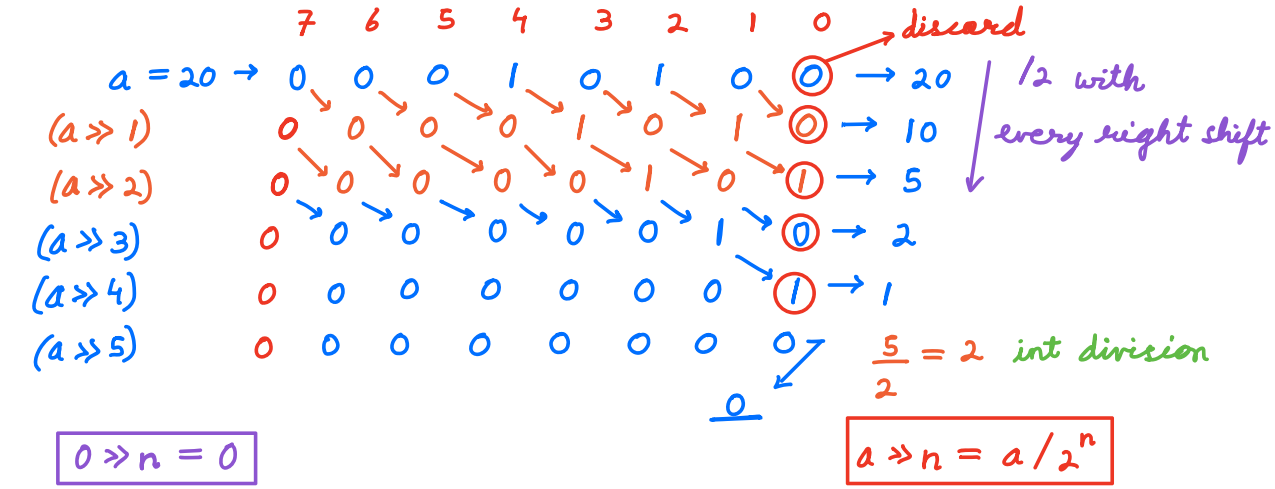
$= 2^8 - 1 = 255$ ✓

Overflow



	7	6	5	4	3	2	1	0	
1 →	0	0	0	0	0	0	0	1	
(1 << 5) →	0	0	1	0	0	0	0	0	→ $2^5 = 32$

Right shift (>>)



$0 \gg n = 0$

$a \gg n = a / 2^n$

---

on → 1

off → 0