

Single Number

XOR ans = 0
ans = a[i] ✓

$$\boxed{1^1 \cdot 1 = 0}$$

$$\boxed{\underline{a^n a = 0}}$$

$$a^{\wedge}1 = \begin{cases} a-1 \\ a+1 \end{cases}$$

$a^a \cdot b^b \cdot c^c \cdot d^d$

$$\boxed{a \quad | \quad ^\wedge} \Rightarrow \boxed{0(1)} \quad \leftarrow \quad \text{—}$$

$$\begin{array}{r}
 10110 \\
 01101 \\
 \hline
 \end{array}
 \quad
 \begin{array}{c}
 13 \\
 \swarrow \\
 \underline{+} \\
 17
 \end{array}
 \quad
 a+b = \underline{\hspace{2cm}}$$

+ } min 5 clock
* } cycles
/ }
 slower

\Rightarrow bitwise operation \rightarrow 1 clock cycle

$$\left(\begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix} \right)_{10} \Rightarrow \left(\begin{matrix} \textcircled{4} & \textcircled{3} & \textcircled{2} & \textcircled{1} \\ 1 & 0 & 1 & 1 \end{matrix} \right)_2$$

Check if 2nd bit is ON/OFF

$$\begin{array}{r} \boxed{0010} \\ \underline{-} \quad (2) \\ \hline 11 \end{array} = (2) \leftarrow \begin{array}{l} \text{non zero value} \\ \text{bit is ON} \end{array}$$

100 if 4th bit is on or off
= 

AND with 8

2nd bit | 000010
4th bit | 001000
5th bit | 010000

K^{th} bit is ON or OFF

$$, \quad (11)_{10} = \underline{\underline{(1011)_2}} \quad \leftarrow$$

$$\begin{array}{r} \cancel{\times 2} \\ \cancel{\downarrow} \end{array} \left(\begin{array}{l} \overset{1}{2} \ 2 \\ \cancel{2} \ \cancel{1} \end{array} \right)_{10} = \left(\begin{array}{l} \cancel{1} 0 \ 1 \ 1 \ 0 \\ \cancel{1} \end{array} \right)_2$$

$$✓ \quad (44)_{10} = \left(\begin{array}{c} 101100 \\ \hline \end{array} \right)_2 \quad ✓$$

$$\begin{array}{rcl} \underline{\underline{11}} & = & 8 + 2 + 1 \\ - & = & \boxed{2^3 + 2^1 + 2^0} \end{array}$$

$$\underline{(22)} = \underline{11} * \underline{2}$$

$$\begin{array}{rcl} 22 = 11 * 2 & = & 2 \left(2^3 + 2^1 + 2^0 \right) = \begin{array}{r} 3 \ 2 \ 1 \ 0 \\ | \quad | \quad | \quad | \\ 1 \ 0 \ 1 \ 1 \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ 1 \ 0 \ 1 \ 1 \ 0 \end{array} \\ & = & \boxed{2^4 + 2^2 + 2^1} \end{array}$$

We observe that when we multiply a number by $\underline{2}$, the bits in binary representation shift towards left.

(Last added bit is zero)

\Rightarrow $\textcircled{<<}$ left shift operator

$$\boxed{a \ll 1}$$

$$\begin{array}{l} \text{print}(\underline{\underline{10}} \underline{\underline{\ll}} \underline{\underline{1}}) \\ \qquad\qquad\qquad (\underline{1010})_2 \\ \qquad\qquad\qquad (\underline{10100})_2 \\ \qquad\qquad\qquad \textcircled{20} \end{array}$$

$a \ll n$

$$a \ll \underline{1} = a * 2 = a^{\times} \text{pow}(2, \underline{1})$$

$$a \ll \underline{2} = a * 4 = a^{\times} \text{pow}(2, \underline{2})$$

$$a \ll n = \dots \boxed{a^{\times} \text{pow}(2, n)}$$

right shift operator

$a \gg 1$

$$11 \gg 1 \Rightarrow \begin{array}{c} \boxed{1011} \\ \Downarrow \\ \boxed{0101} \end{array} \quad \begin{array}{l} 11 \\ \downarrow \\ 5 \end{array} \quad \begin{array}{l} \text{Integer division} \\ \text{by } 2 \end{array}$$

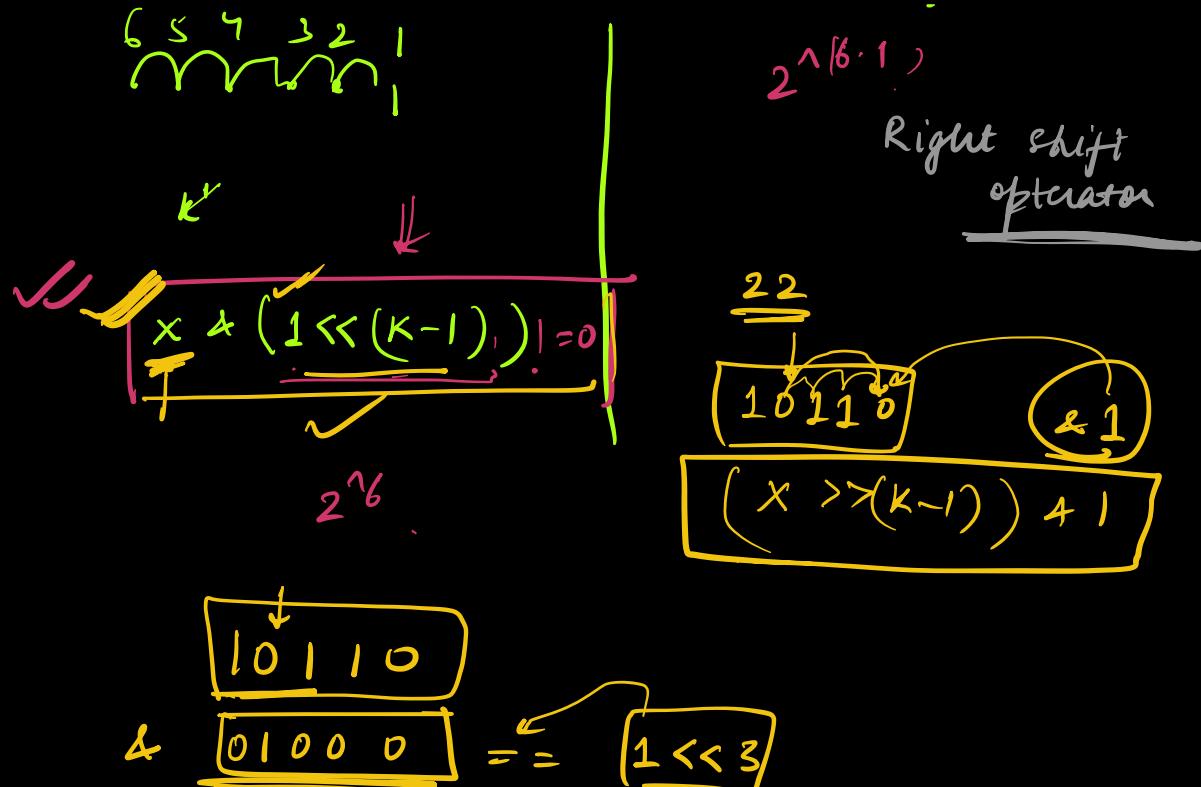
$\boxed{11/2 == 11 \gg 1}$

$$11 = 8 + 2 + 1$$

$$\begin{aligned} &= 2^3 + 2^1 + 2^0 \\ &= \cancel{2^3/2} + \cancel{2^1/2} + 2^0/2 \\ &\quad \left(\cancel{2^3/2} \right) + \cancel{2^1/2} + 2^0/2 \end{aligned}$$

Q8

check if 6th bit is ON/OFF
in numol 997 = x



10110
1011
101
10

$(x >> (k-1) \& 1)$

3 time right shifts

$10 \& 1$

1010
 ↓ >> 1
 0101
 ↓ >> 1
 0010
 ↓ >> 1
 0001
 ↓
 0000 ✓

$\begin{array}{r} 10 \\ \hline 10 >> 4 = 0 \end{array}$
 [50] >> 2

~~11001~~
 Integer division by 2
 right shift operator

~~11001~~
 ⇒ 25 ✓
 ↓
 [1100] ⇒ 12

Q Find no. of bits switched ON in
number x



----- \Rightarrow 4 bits \Rightarrow different
nos. can
be stored

$$= \underline{\underline{2^4}}$$

$$\text{-----} = \underline{\underline{2^8}} = \boxed{0 \text{ to } 255}$$

$$32 \text{ bits} = \underline{\underline{2^{32}}} \quad (0 \text{ to } \underline{\underline{(2^{32}-1)}})$$

$$= 0 \text{ to } 10^9$$

$$\overbrace{\text{or } 4 \times 10^9}$$

18 digits

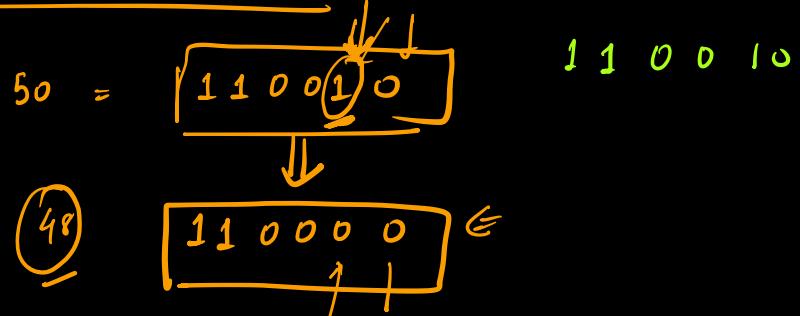
64 bits

$$\underline{\underline{2^{64}}}$$

64 bits to cover

$$\underline{\underline{10^{98}}}$$

Q Uncet the last set bit :



① find i where i^{th} bit ON

② - $x = x \wedge (1 \ll i)$

$i = 0$

while True :

if $x \wedge (1 \ll i) \neq 0$

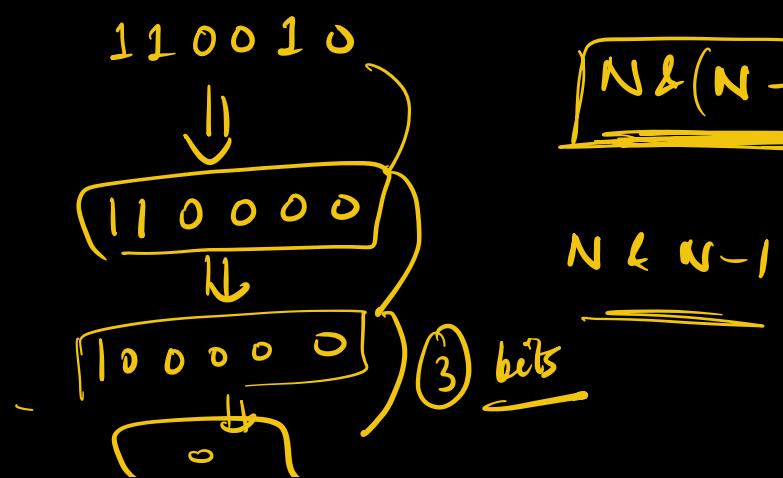
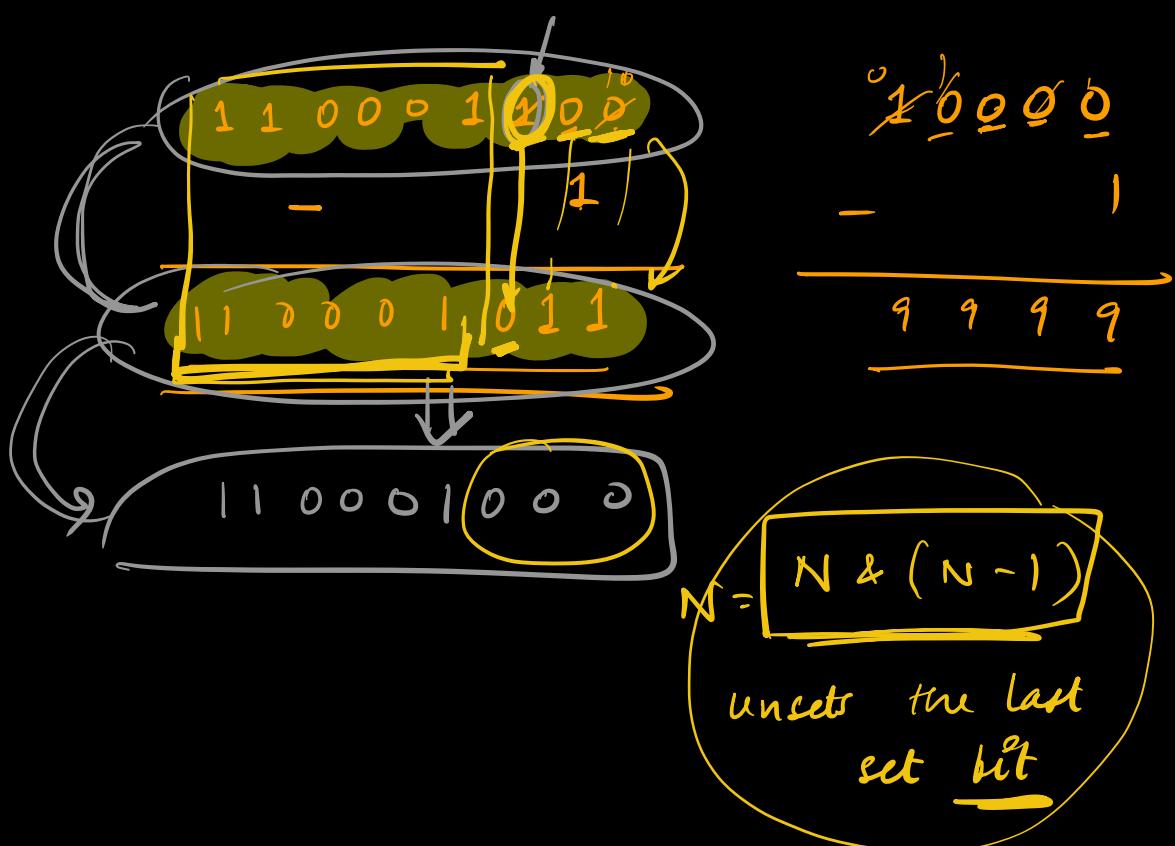
break.

$i += 1$

$$x = x \wedge (1 \ll i)$$

return x

$$\begin{array}{r} 1 & 1 & 0 & 0 & 1 & 0 \\ - & & & & & 1 \\ \hline 1 & 1 & 0 & 0 & 0 & 1 \end{array}$$



signed bits



How do we represent negative

No. 1

$$2^3 = -2^3 \Rightarrow 8$$

vee

ON
—

Then number
is negative

This bit
represents

-ve
magnitude

2⁴ numbers

$$= [0 - 15] \Rightarrow +ve \text{ nos.}$$

are represented

ONLY 16 numbers can be
represented in 4 bits.

$$[-8 \text{ to } 7] \Rightarrow 16 \text{ numbers}$$

2⁸ complements

0	→	0 000
1	→	0 001
2	→	0 010
3	→	0 011
4	→	0 100
5	→	0 101
6	→	0 110
7	→	0 111

$$-8 \Rightarrow 1000$$

$$-7 = 1001$$

$$-6 = 1010$$

$$-5 = 1011$$

$$-4 = 1100$$

$$-3 = 1101$$

$$-2 = 1110$$

$$-1 = 1111$$

$$\begin{array}{r} 1 \\ - \\ -2^3 \\ \hline \end{array}$$

$$\begin{array}{r} 0 \\ - \\ 2^2 \\ \hline \end{array}$$

$$\begin{array}{r} 0 \\ - \\ 2^1 \\ \hline \end{array}$$

$$\begin{array}{r} 0 \\ - \\ 2^0 \\ \hline \end{array}$$

$$\begin{array}{r} -3 \\ -2 \\ -1 \\ \hline \end{array}$$

signed

Data
types

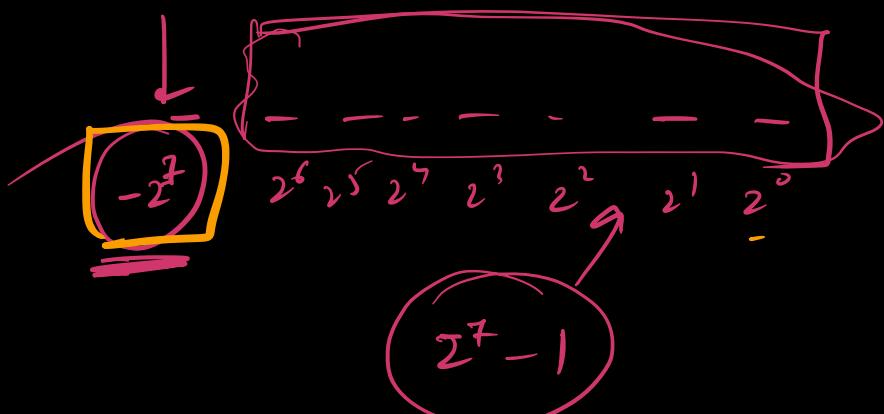
$$-2^3 + 0 + 0 + 0 = -8 = \textcircled{-8}$$

1 0 0 1

$$\begin{array}{r} -2^3 \\ 2^2 2^1 2^0 \\ \hline \end{array}$$

$$-2^3 + 2^0$$

$$-8 + 1 = \textcircled{-7}$$



8

8 bits

0 - 255

0 -

$-128 \text{ to } 127$

$\textcircled{-1} \Rightarrow$ 8 bits ?

-2^7

$$-2^7 + \boxed{2^6 + 2^5 - \dots - 2^0} \quad \downarrow \quad (GP)$$

$$-2^7 + 2^7(-1) \checkmark$$

(-3) binary representation

2^7 complements

$$\begin{array}{rcl} -3 & = & 00000011 \\ \text{char('0')} & & \xrightarrow{\quad\quad\quad} \\ \text{char('1')} & & \\ \text{int('01', 2)} & & \\ \hline & & \\ & & \text{int('1')} \\ & & \text{ord('1')} \\ & & \hline & & \\ & & 11111100 \\ & & + \\ & & 11111101 \\ & & \hline & & -3 \end{array}$$

Algo 2^7 complements

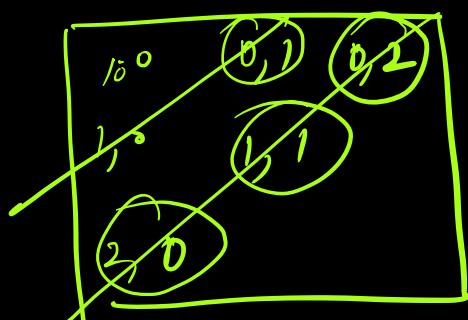
1st complement Toggle all bits of its positive magnitude representation

2nd Add 1

-8 ~ 1000

0111
1000

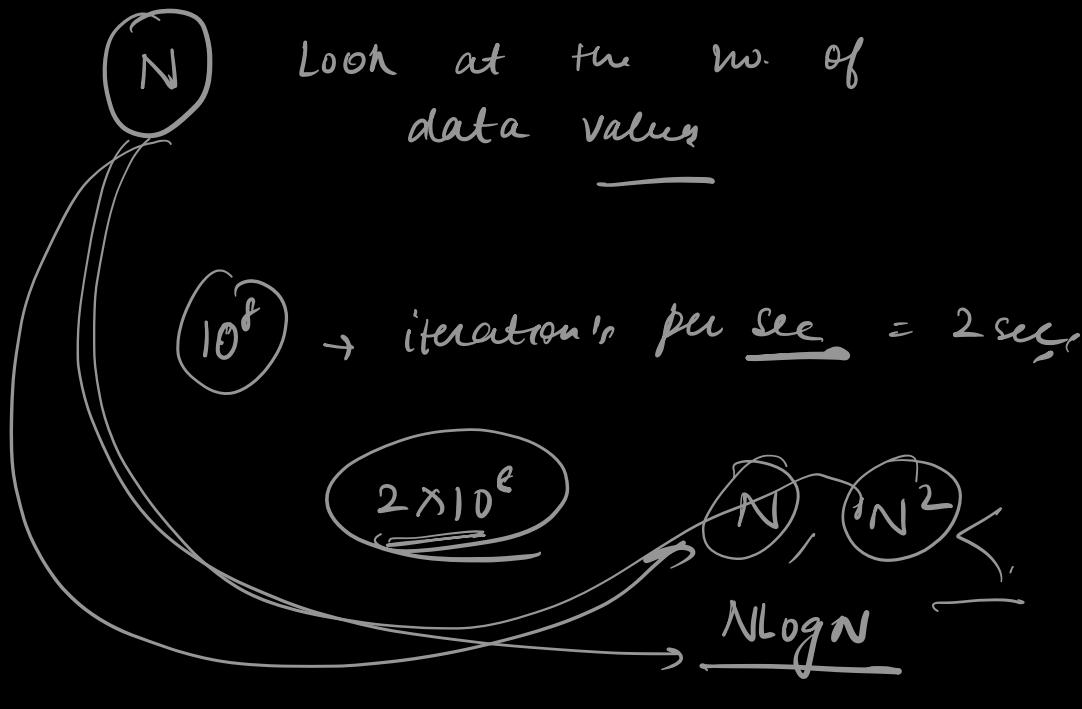
-4 = 0100
0011
+ 1100 →



(i+j) = constant

[]
[]
[]
[]

① $T(n) \rightarrow$ code you have written



$$N \approx 10^3$$

$$\overbrace{N^2}$$

$$\overbrace{10^6 < 10^8}$$