- Bitmasking

&	and	done
|	or	
^	xor	
~	not	
<<	Left Shift	
>>	Right shift	

directly on bit

$\boxed{5}$ ↓ 0000101

← 32 bit →

Left Shift (<<) → Two operands

$\boxed{a << b}$

$\boxed{5}$

5 → 000000 101

← fill with 0 on right

5 << 1 → 000000 101 0 ⇒ $\boxed{10}$

$13 \Rightarrow \boxed{x}$

$$2^3 \; 2^2 \; 2^0$$
$$8 \; 4 \; 2 \; 1$$

$\boxed{0\,1\,1\,0\,1}$ $= 8 + 4 + 1 = \boxed{13}$

$13 << 1 = 26$

$= \boxed{y}$

$\boxed{1\,1\,0\,1\,0}$ $= 16 + 8 + 2 = 26$

$$16 \; 8 \; 4 \; 2 \; 1$$
$$2^4 \; 2^3 \; 2^1$$

$$x = 2^3 + 2^2 + 2^0$$

Doing left by 1 multiplies the no by 2

$$y = 2^4 + 2^3 + 2^1$$
$$= 2(\underbrace{2^3 + 2^2 + 2^0})$$
$$=$$
$$\boxed{y = 2x}$$

$5 \rightarrow 101$ — 5

$5<<1 \rightarrow 1010$ — 10

$10<<1 \rightarrow 10100$ — $\boxed{20}$

Loss of Data for Large no

$\boxed{0000}$

New No is outside the range of int. ✗

$5<<2 \rightarrow 10100 = \boxed{20}$

$5<<1$ Two times

$\boxed{5<<2}$

$5 \times 2 \times 2$

5×2^2

$= \boxed{20}$

$$\boxed{a << b \Rightarrow a * 2^b}$$

Left shift operator

$6 << \underline{3} \Rightarrow \overline{11}0000 = \boxed{48}$

32 16 8 4 2 1

bit

$\parallel$

$6 * 2^3$

$= 6 \times 8$

$= \boxed{48}$

$8 << 2 \Rightarrow \boxed{1000}\,00$

2^3

$2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$

$= \underline{32}$

$2^3 \curvearrowright 2^5$

$\parallel$

$8 * 2^2$

$= 8 \times 4$

$= 32$

Right Shift Operator

↳ Divide by Powers of 2

$\boxed{20}$

$\boxed{1010\cancel{0}}$ (16 8 4 2 1) $\quad 20 >> 1$

$\boxed{0101\cancel{0}}$ $\quad = 10 >> 1$

$\boxed{00101}$ $\quad = 5 >> 1 \qquad 5 >> 1 = \frac{5}{2^1} \Big\}$ int div

$\boxed{0001\cancel{0}}$ $\quad = 2 >> 1$

$\boxed{00001}$ $\quad = 1 >> 1$

$0 \quad = 0$

$$\boxed{a >> b = \frac{a}{2^b}}$$

Result

$$20 >> 3 = \frac{20}{2 \cdot 2 \cdot 2} = \frac{20}{2^3} = \frac{20}{8} = 2$$

Java
(specific)

"Signed" Right Shift

>>

Unsigned Right Shift

>>>

mean the MSB (32^{nd} bit) denotes the sign of no

+ve	-ve
0 | 1 | 1 | 1 | 1 | 1	1 |
32	32

uses the MSB / left most bit to fill trailing positions after the shift.

if you have negative, then filler → 1

——— positive, then filler → 0

1 1 0 1 1 0 - - 1 ⇒ 1 | 1 1 0 1 1 0 - -

x

1

-ve   -ve

$x = -9$ $\qquad\qquad$ $y = 9$

$\boxed{x >> 2}$ $\qquad\qquad$ $\boxed{y >> 2}$

$\Rightarrow \dfrac{-9}{2^2} = \dfrac{-9}{4} = \boxed{-2}$ $\qquad\qquad$ $\dfrac{9}{2^2} = \boxed{2}$

Q $\boxed{-5} \to$ Stored $\to$ 2's compliment form

$5 \to \boxed{0\ 1\ 0\ 1}$

$-5 \to$

1 0 1 0

+ 1

$-5 \longrightarrow \boxed{1\ 0\ 1\ 1}$

1 0 1 ~~1~~

1 filler

$-3 \longleftarrow \boxed{-5 >> 1}$

1 1 0 1 $\to$ -ve no $-(-x)$

$-(-x)$

$\|$

$+ \boxed{x}$

$= 3$

$-a >> b$

$\to \dfrac{a}{2^b}$

Flip all

$\boxed{0}$ 0 1 0

+ 1

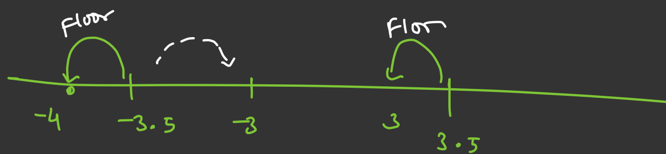
0 0 1 1 $\to$ 3

2s compliment for

Flip +1

$a >> b = \text{Floor}\left(\frac{a}{2^b}\right)$

for both +ve & -ve no's



$-7 >> 1 = -4$

$+7 >> 1 = 3$

Recap

$\boxed{7} = \boxed{0\ 1\ 1\ 1}$

$\boxed{-7} =$

1 0 0 0

\+ 1

$\boxed{1\ 0\ 0\ \cancel{1}}$

1

$-7 >> 1 \longrightarrow \boxed{1\ 1\ 0\ 0} \longrightarrow \boxed{-4}$

$\boxed{-x}$

$-(-x) = +x$

$\downarrow$ 2s

1 1

0 0 1 1

\+ 1

$\boxed{0\ 1\ 0\ 0}$

$x = 4$

-4

$1 + 1 = \boxed{10}$

$0 + 1 = 1$

$1 + 0 = 1$

$0 + 0 = 0$

Give a no, how to find in 2's compliment

$\boxed{7} \longrightarrow \boxed{0\,1\,1\,1}$ Flip all bit & add 1

$\boxed{-7}$

1 0 0 0

$+ \quad 1$

$-7 \rightarrow \boxed{1001}$

$\boxed{-7 >> 1}$ $\boxed{1100} = \boxed{-4}$

$\boxed{-7}$

There are 2 ways to write ~~deci~~ binary negative

$\boxed{-ve}$ 1 0 0 1 $-\boxed{-x}$ $= \boxed{+x}$

[2s compliment form]

0 1 1 0

$+ \quad\quad 1$

0 1 1 1 $\Rightarrow \boxed{+7}$

1 $\boxed{0\,0\,1}$ 1

Regular form

$\boxed{-1}$

+ve no's

$x = -1$

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

32bit

+1 0 0 0 0 0 1

-ve

1 1 1 1 1 1 0
+ 1

1 1 1 1 1 1 1 1

Guess the output

$\boxed{-1 >> 4}$

1 1 1 1 1 1 1 ~~1 1 1 1~~

1 1 1 1

$-1 >> 4 = -1$

$1 >> 4$

$= \frac{1}{2^4} = 0$

0 | 0 0 0 0 0 ~~0 0 0 0~~

Unsigned Right Shift $\boxed{>>>}$ Filler ↗ 0 → +ve ↘ 1 → −ve

doesn't differentiate b/w +ve & −ve no

↓

$\boxed{\text{Filler is always } 0}$

$x = -1$

$\boxed{x >>> 29}$

29 bit →

| 1 1 1 1 1 1 1 1 1 1 1 1 1 |

29 0's

$\boxed{0}$ 0 0 0 0 0 0 0 0 1 1 1 = 7 +ve

29 ⟷ 3 bit as one

5 >> 1 → 0 1 0 $\cancel{1}$

>>, >>>

$\boxed{-1}$ >> 31 ⇒ | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |

Add 31 0's ⇒ | 0 0 0 0 0 0 0 0 0 0 0 1 | $\boxed{+1}$

unsigned

$-5 >>> 2$ → 00 10 ** → $\boxed{0010}$ = 2

+5 000000 101

-5 1010
+ 1

Q Given a +ve No, find out the No of set bits (ones) in that No

shared memory

Input

5 → 00 0101 → 2

7 → 000111 → 3

Decimal to Binary

2	13
2	6, 1
2	3, 0
2	1, 1
	0, 1

rem

$\boxed{1101}$

```
n = 13 , cnt = 0
while (n > 0) {
    rem = n % 2;
    cnt = cnt + rem;   → 1
                       → 0
    n = n/2;
}
3
print(cnt);
```

cnt 0

n = 13 > 0   rem = 1   = +1
↓
6 > 0   = 0   + 0
|

3 > 0 = 1 + 1

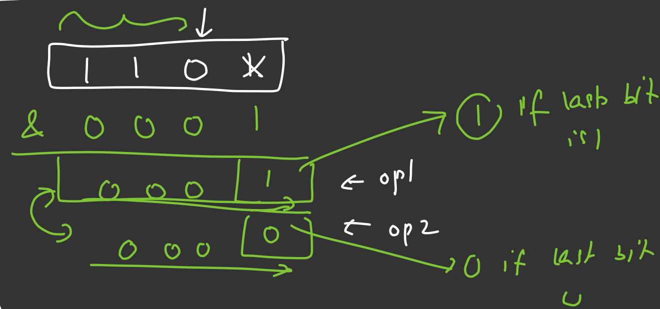
1 > 0 = 1 + 1

0

3 set bit

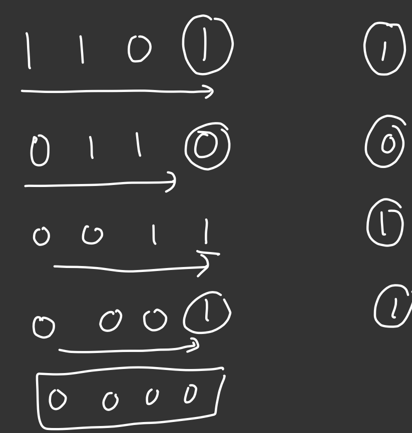
Bitwise way

N = $\boxed{13}$



n = 13, cnt = 0

```
while( n>0 ){
    last_bit = n&1
    cnt = cnt + last bit
    n = n>>1
}
print(cnt)
```



$\boxed{0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ \cancel{0}}$

↓

. . 0 0 0 0 1 0 1 0 $\cancel{x}$ = 20

↓

1 0 1 $\cancel{0}$ = 10

↓

1 0 1 = 5

↓

1 0 = 2

↓

$\cancel{1}$ = 1

↓

0

N

↓ /2

0

→ $\boxed{\text{Log N}}$

≤ 32

O(1)

complexity

$\boxed{40}$

1 0 1 0 0 0

32 16 8 4 2 1

n = n>>1

↓

20

↓

10

↓

5

→ 4

↓

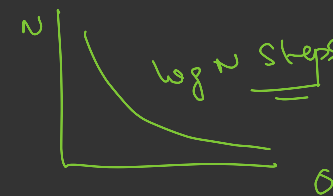
2

↓

1

↓

0



10·40

Q. Generate a no in which only bits at loc x and y are set, rest everything is 0.

0 based index from Right

$x = 3$

$y = 1$

without taking an array

31	.....	5	4	3	2	1	0
0 0		0	0	1	0	1	0

$= 2^3 \quad 2^1$

$\boxed{1 << 3} + \boxed{1 << 1} = (8+2) = 10$

$1000 + 10 = 1010$

8	4	2	1
1	0	1	0

$= 10$

$2^3 + 2^1 =$

$$\begin{array}{r} 1000 \\ +\ 10 \\ \hline 1010 \end{array}$$

$\Downarrow$

10

$$\frac{10}{2}$$

$\boxed{OR} \rightarrow$ Bit Level

$$\begin{array}{rcr} 1<<3 & = & 1000 \\ 1<<1 & = & 0010 \\ \hline & & 1010 \end{array}$$

$$\boxed{ans = (1<<x) \mid (1<<y)}$$

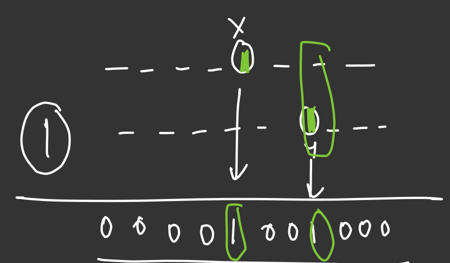
OR $\rightarrow$ no carry.

Addition

$1+1 = 10$

$1 \mid 1 = 1$

ORing



$$5 \mid 7 = \begin{array}{r} 101 \\ 111 \\ \hline 111 \end{array}$$

$$\left\{ \begin{array}{l} 1 \text{ or } 1 = 1 \\ 1 \text{ or } 0 = 1 \\ 0 \text{ or } 1 = 1 \\ 0 \text{ or } 0 = 0 \end{array} \right\}$$

Bits

$$\begin{array}{cccccccc} & & & & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{array}$$

OR

0000 1 0000
000000 0 0 1 0

powers of 2

$2^3 + 2^1$

$= 8+2$

$= 10$

get I^{th} Bit (no, i)

$\boxed{13}$

$$\begin{array}{cccccc} 31 & \cdots & 3 & 2 & 1 & 0 \\ 0000 & 0 & 1 & 1 & 0 & 1 \end{array}$$

→ make this bit as last bit

way-1

7 6 5 4 3 2 1 0

no → 0 0 0 0 $\boxed{1}$ 1 0 1

0 0 0 0 1 1 0

0 0 0 0 1 1

0 0 0 0 $\boxed{1}$

i^{th} bit now at last

i^{th} times Right Shift

$$\underbrace{(no >> i) \;\&\; 1}$$

$\Rightarrow 1$

$\Rightarrow 0$

way-2

ith bit $\quad i=5$

no	0 0 1 0 0 $\boxed{1}$ 0 0 0 1 0	← original
&	0 0 0 0 0 $\boxed{1}$ 0 0 0 0 0	← mask
result	0 0 0 0 0 $\boxed{1}$ 0 0 0 0	

1 $\quad$ 0

if org is 1

$1<<i$

mask $= (1<<i)$

no & mask

1 00000 ← ith place

result $== 0$ → ith bit 0

└→ > 0 → ith bit 1

$1\ 1\ 0\ \boxed{1}\ \cancel{0}\ \cancel{1}$

last bit

Ap-1

$$
\begin{array}{r}
 0\ 1\ 2 \\
\cancel{1}\ \cancel{1}\ \cancel{0}\ \boxed{1}\ \cancel{0}\ \cancel{1} \\
\&\ \ \boxed{0\ 0\ 0\ 1\ 0\ 0} \quad \boxed{1 \ll 2} \\
0\ 0\ 0\ 1\ 0\ 0 \rightarrow >0 \Rightarrow 1 \\
\hline
0\ 0\ 0\ 0\ 0\ 0 \rightarrow 0
\end{array}
$$

$0\ 0\ 0\ 0\ 1\ 0\ 0\ 0$

ith loc $\Rightarrow$ $1 \ll i$

Q. Given N, set i^{th} Bit in N

$$N = \begin{array}{cccc} 3 & 2 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{array}$$

1 1 1 1

set → 1st bit as 1

set → 3rd bit as 1

i^{th} bit

i^{th} bit → 1

1 1 0 1 1 1 0 1

0 0 1 0 0 0 0 0 ← mask = $1<<i$

Bitwise OR

1 1 1 1 1 1 0 1

$\boxed{N = N \mid mask}$

$N \mid (1<<i)$

$1<<0 = 1$

$1<<1 = 10$

$1<<2 = 100$

$1<<3 = 1000$

$1<<4 = 10000$

$1<<i = 1\underbrace{000000}_{i \text{ zeroes}}$

$N = 13$

$i = 1$

$$\begin{array}{cccc} 3 & 2 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{array}$$

$\boxed{15}$

$13 \mid (1<<1)$

$= 1101 \mid 10$

$= \boxed{1111}$

ToDo

Unset at loc i,

→ Given a No, make i^{th} bit as 0. ??

$N = 13$

$$\begin{array}{cccc} i=3 & i=2 & i=1 & i=0 \\ 1 & 1 & 0 & 1 \end{array}$$

$\Rightarrow \boxed{1001}$

Binary Addition

	16	8	4	2	1
carry		1	1		
	1	0	1	1	0
+	0	0	1	1	1
	1	1	$\boxed{1}$	0	1
	16	8	4	2	1

→ 22
+
→ 7 = 29

→ $\boxed{29}$

$1+1 = \boxed{10}$

↑

$1+1 = \boxed{2}$

$2 \% 2 \Rightarrow 0$

$2/2 \Rightarrow 1$

$3 \% 2 \Rightarrow$

$3/2 \Rightarrow 1$

carry →
rem →

$1+1+1$
$= 3$
$= \boxed{11}$

$\boxed{11}$

$\underbrace{1+1}+1$
$= 10+1$
$= \boxed{11}$

$\boxed{\text{DIY}}$

Addition

	1		1	
	7	8	3	9
	3	9	4	8
1	1	7	8	7

$9+8 = \boxed{17}$

rem $= 17 \% 10 = 7$

carry $= 17 / 10 = \boxed{1}$

↑
Base