Bit-wise operators.

1. Binary And (&)
2. OR (|)
3. XOR (^)
4. One's Compliment (~)
5. Left shift (<<)
6. Right shift (>>)

---

1. Binary AND (&)

0 & 0 → 0
0 & 1 → 0
1 & 0 → 0
1 & 1 → 1

5 & 6

```
 0101
 0110
------
 0100
```

& = 4

5 + 6 = 11
5 & 6 = 4.

---

2. Binary OR (|)

0 | 0 → 0
0 | 1 → 1
1 | 0 → 1
1 | 1 → 1

{ 5 | 6 }

```
 0101
 0110
------
 0111
```

⇒ 7

---

3. Binary XOR (^)

0 ^ 0 → 0
0 ^ 1 → 1
1 ^ 0 → 1
1 ^ 1 → 0

5 ^ 6

```
 0101
 0110
------
 0011
```

⇒ 3

Different bits result 1.

{ 5 ^ 5 = 0 }

{ even 1's ⇒ 0
odd 1's ⇒ 1 }

④ One's compliment (~)

~0 → 1

~1 → 0

5 → 0000 0101

MSB LSB

if MSB is 0 ⇒ num is +ve

if MSB is 1 ⇒ num is -ve

⇒ W.K.T To get The value of a num we take the 2's compliment.

⇓

1's compliment & add 1 to it.

5 ⇒ 0000 0101

$\boxed{\sim 5 \Rightarrow 1111\ 1010}$

⇓

⇒ 0000 0101
+ 1

0000 0110 = (6)

⇒ As we took 2's complement we will get the changed sign.

⇒ -6

⑤ left shift. <<

5 << 2
↳ digits to shift.

$a \times 2^{b}$

b → no. of digits

a ↓ Number.

$5 \times 2^{2} = 20.$

5 → 0 0 0 1 0 1

5<<2 → 0 1 0 1 0 0

⇓

5	4	3	2	1	0
0	16	0	4	0	0

⇒ 16 + 4 = 20.

⑥ Right Shift >>.

6 >> 2

6 >> 2

0000 0110
0000 0001 $\xrightarrow{10}$ $\underbrace{}_{1}$

1.

Floor Division

$\lfloor a / 2^b$ = $6/4 = 3/2 = (1.5) = 1$

round ⇓ 3.6 ⇒ 4, 3.4 ⇒ 3.

Floor ⇓ 3.6 = 3

ceil → 3.6 = 4.

---

① check if num is even or odd

0 → 000
1 → 001
2 → 010
3 → 011
4 → 100
5 → 101

⇒ from the patterns we can notice that
if LSB is zero → num is even
one → num is odd.

① To get rid of other digits. ⇒ for Masking all other digits we can use num & 1.

1101
& 0001
0001 → $\underbrace{000}_{1}$ we are left with LSB of 1.

100
001
000 → 0

(2) Operations

↳ Get i^{th} Bit

↳ Set i^{th} Bit

↳ Clear i^{th} Bit.

---

(1) Get i^{th} Bit

7 ⇒ 0111 → for $i = 2$

consider 1 , 0001 left shift it by 2 ⇒ 0100

$$\begin{array}{r} 7 \quad 0111 \\ \& \quad 0100 \\ \hline 0100 \end{array}$$

⇒ $n \ \& \ \boxed{1 << i}$

↳ if the result isn't zero then the number is 1 at i^{th} pos.

else : zero.

---

(2) Set i^{th} Bit (the i^{th} Bit maybe 0/1 we need to Make it 1.)

0000 1010

0000 0110 ← if $i = 2$

Consider a number 10 (1010)

and $i = 2$. ⇒ bit mask = $1 << i$

= 0100

$$\text{(or)} \quad \begin{array}{r} 1010 \\ 0100 \\ \hline 1110 \end{array} \Rightarrow 14.$$

③ clear ith Bit (To make ith bit zero).

&

0 1 1 1
1 0 1 1
0 0 1 1 ↳ 3

$1 << i \Rightarrow 0100$ ($i=2$)

Compliment ↳ $\boxed{1011}$ ✓

④ update

newbit = 0 or 1.

num = 7

i = 2.

if newbit == 1 :
  set bit
else
  Clear bit

⇝ approach 1

approach 2 :

⇒ clear Bit.

Bit mask = newbit << i

if newbit == 0 ⇒ 0000
else ⇒ 0100 if i=2

Now n | bitmask.

⑥ clear last ith Bits.

last 3 bits.

0110 1110
1111 1000 } and
0110 1|000

$(\sim 0) \Rightarrow$ 1111 1111

$\boxed{\sim 0 << 3}$
↳ 1111 1000 & num.

15 → 1111
i=3   1000
       1000
       8.

7 set last im Bits.

8 → 1000
$i=2$ 0011
1011

} or.

0000 0011 ⇒ ?

0000 0100
\- 1
0000 0011

⇒ $\underbrace{(1<<i)-1}_{\text{bit mask}}$

---

8 Clear Range of Bits.

$i=3$, $j=5$

1011 1101
& 1110 0011
1010 0001

→ (~ 0) ⇒ 1111 1111

$\underbrace{(\sim 0 << j)}_{a}$ = 1110 0000

1011 1101
189

1010 0001 → 161

b = 0000 0011
↳ $(1 << (i-1))$

⇒ $b = (1 << (i-1)) - 1$

bit mask = a|b
= 1110 0011

---

9 Check if number is power of 2 or not.

10
100
1000

1. 2 & 1
4 & 3

⇒ $(n) \& (n-1) = 0$ ✓ is power of 2.

2. Count 1's if 1 one in whole num then it's power of 2.

(10) Count Set Bits in a number.

10 → 1010

Process

(1) Consider LSB if 1 increment it.

(2) Right shift by 1 = store it in n.

Count = $\cancel{0}$ $\cancel{1}$ 2

→ 1010

→ 101

→ 10

→ 1

(11) Fast Exponentiation

a^n

$3^5 = 243$

$3^5 \Rightarrow 3^{\overset{4\,2\,1}{101}}$ → $(a^4)\,(1)\,(a^1) = a^5$

ans = 1

→ a = 3, ans = 1

ans = 3

$a = 3^2$

⤳ if LSB is 1

→ a = 9, ans = 3

ans = 3

$a = a^2 = 81$

→ a = 81, ans = 3

ans = 3 X 81 = 243

$a = a^2 = (81)^2$

(12) Swap 2 nums

$x = 3$

$y = 4$

$x = 3 \wedge 4$

$y = 3 \wedge 4 \wedge 4 = 3$

$x = 3 \wedge 4 \wedge 3 = 4$

---

(13) Flip Bits. [How many Bits should be flipped to obtain given num B].

$A = 7 \rightarrow 0111$

$B = 12 \rightarrow 1100$

} 3 bits

→ output.

→ Get LSB compare
if not equal increment count

→ Right shift by 1.

---

(14) Addition → 5, 3 = 8.