Bit wise operators + Number system

Bit -manipulation

Operators

01 AND

a	b	a & b
0	0	0
0	1	0
1	0	0
1	1	1

Ques given a number N, find if it's even or not.

Note: when you & 1 with any number the digits ~~rema~~ remains the same

```
  1 1 0 0 1 0 1 0 0
& 1 1 1 1 1 1 1 1 1
-------------------
  1 1 0 0 1 0 1 0 0
```

02 OR

If anyone is true the entire expression is true.

a	b	a | b
0	0	0
0	1	1
1	0	1
1	1	1

03 XOR(^) (if and only if)

↳ exclusive OR ⟶ for odd ones it should be zero for 3 values

a	b	a^b
0	0	0
0	1	1
1	0	1
1	1	0

observations

(i) $a \wedge 1 = \bar{a}$

XOR any number with $\boxed{\text{one}}$ it'll give complement of it.

Ex: 0 XOR 1 = 1

1 XOR 1 = 0

(ii) $a \wedge 0 = a$

(iii) $a \wedge a = 0$

04 complement(~)

a = 10110

$\bar{a}$ = 01001

Number systems

(01) Decimal → 0, 1, 2, ..., 9   Base : 10

$(357)_{10}$, $(10)_{10}$

(02) Binary → 0 & 1   Base : 2

$(10)_{10} = (1010)_2$

$(7)_{10} = (111)_2$

(03) Octal → 0, 1, 2, 3 ... 7   Base : 8

Decimal : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

Octal : 0, 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 14, 15, 16, 17,
20, 21, 22, ...

$\Rightarrow (9)_{10} = \underbrace{001}_{1}\ \underbrace{001}_{1}$

$(11)_8$

$(9)_{10} = (11)_8$

(04) Hexadecimal → 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

A → 10, B → 11, C → 12, D → 13, E → 14, F → 15

$(10)_{10} = (A)_{16}$

$(12)_{10} = (C)_{16}$

Conversations

1. Decimal to base b
2. ~~Base b to decimal~~

Ques convert $(17)_{10}$ to base 2

keep divide by base, take remainders write in opposite

$$
\begin{array}{r|l}
2 & 17 \\
2 & 8 - 1 \\
2 & 4 - 0 \\
2 & 2 - 0 \\
2 & 1 - 0 \\
 & 0 - 1
\end{array}
$$

LSB ↑ MSB

$(10001)_2 = (17)_{10}$

$(17)_{10} = (?)_8$

$$
\begin{array}{r|l}
8 & 17 \\
8 & 2 - 1 \\
 & 0 - 2
\end{array}
$$

LSB ↑ MSB

$(21)_8$

2. Convert any base b to decimal

$(10001)_2 = (\quad)_{10}$

steps: multiply & add to the power of base with digits

$$
\begin{array}{ccccc}
4 & 3 & 2 & 1 & 0 \\
1 & 0 & 0 & 0 & 1
\end{array}
$$

$1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

$= 16 + 0 + 0 + 0 + 1 = (17)_{10}$

Ques $(21)_8 = (?)_{10}$

$$\begin{array}{cc} 1 & 0 \\ \uparrow & \uparrow \\ 2 & 1 \end{array}$$

$= 2 \times 8^1 + 1 \times 8^0$

$= 16 + 1 = (17)_{10}$

Continuing with operators

(05) Left shift operator ($<<$) $\rightarrow$ shift bits towards the left

$(10)_{10} = (1010)_2$ $\quad 10 << 1$

Step: $1010 << 1 = \begin{array}{ccccc} 1 & 0 & 1 & 0 & 0 \\ \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\ 4 & 3 & 2 & 1 & 0 \end{array}$ $\rightarrow$ (extra bit appended due to left shift)

$1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$

$= 16 + 0 + 4 + 0 + 0$

$= 20$

$\boxed{a << 1 = 2a}$ $\rightarrow$ any number left shift 1 doubles the number

General point:

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

Ex $2 << 4 = \boxed{2 \times 2^4}$

$= 2 \times 16 = 32$

06 Right shift operator (>>)

↳ shifts bits toward the right

Ques $0011001 >> 1 \Longrightarrow 0001100$

ignore these parts

$(0011234)_{10} = (11234)_{10}$

leading zeros are ignored, same for all number system.

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

Questions

Q1 given a number find if it is odd or even?

$4 \to 100$
$\& 1 \to 001$
000

Point every number is calculated in binary form internally

$5 \to 101$
001
001

$12+7 \Rightarrow$

$$\begin{array}{r} 1100 \\ +\ 0111 \\ \hline 10011 \end{array}$$

$7 \to 111$
$\& 1 \to 001$
001

$(19)_{10} = (10011)_2$

$8 - 1000$
$\& \ 001$
0000

$16 + 2 + 1 = 19$ ✓ this is known as LSB

Note: $\boxed{1\ 0\ 0\ 1}$ 1 leaving this, every other is a power of 2

This will always be even

hence If 2^0 0-th place $== 1 \Rightarrow$ odd

otherwise $\Rightarrow$ Even

$$\begin{array}{r} 100101 \\ \&\ 000001 \\ \hline 000001 \end{array} \Rightarrow (1)$$ Hence odd number

Sum up: $\boxed{n \& 1} == 1 \Rightarrow \boxed{\text{odd}}$

else $\Rightarrow \boxed{\text{even}}$

Q2 You're given an array of numbers in that array every no. appears twice ~~onl~~ except one number. Find that number.

arr = [2, 3, 4, 1, 2, 1, 3, 6, 4]

6 → Ans

Note: Bitwise operators support $\boxed{\text{associative}}$ property.

$(5\times3)\times(5\times4) = 5\times5\times3\times4$
$= 5\times4\times5\times4$
$= 3\times4\times5\times5$

all are same hence, order doesn't matter

we know,

$\boxed{a \wedge a = 0}$

and, $\boxed{a \wedge 0 = a}$

If, we XOR (^) the entire array all the duplicates will lead to zero

Only thing remaining will be $\boxed{0 \& 6}$ which will be $\boxed{6}$.

CamScanner

Ques. [-2, 3, 2, 4, -5, 5, -4]

↳ only number that doesn't contain negative value in the given array

Sum all these numbers then we'll get the answer.

Q3 You're given a number find $\boxed{i\text{-th bit}}$ of the number.

7	6	5	4	3	2	1	0
↑	↑	↑	↑	↑	↑	↑	↑
1	0	1	1	0	1	1	0

This is called a mask

Ans

```
  1 0 1 1 0 1 1 0
& 0 0 0 1 0 0 0 0
-----------------
  0 0 0 1 0 0 0 0
```

mask

separate entity that allows us to get our answer related

If given number is N

⇒ we need to create mask with N-1 zeros.

$1 << 4 \Rightarrow 10000$

$1 << (N-1)$

i ≥ 0 maybe check !? ?

Answer : $\boxed{n \,\&\, (1 << (n-1))}$

Ques Set the i-th bit,

turn it into 1 → If it's 0 → change to 1

else 1 → keep 1

4th

	1	0	0	1	0	1	1	0
|(OR)	0	0	0	0	1	0	0	0
	1	0	0	1	1	0	0	0

→ $\boxed{N \,|\, (1 << (N-1))}$

set operation

we have to do OR operation in this situation.

Ques Reset ith bit

1 → 0

0 → 0

5th

	1	0	1	0	1	1	1	0
&	1	1	0	1	1	1	1	1
	1	0	0	0	1	1	1	0

mask $\boxed{!\,(1 << (n-1))}$

Bitwise operators + Number Systems (cont.)

Ques Find the position of the right most set bit

Ex:

$$\underbrace{1\ 0\ 1\ 1\ 0\ 1}_{a}\ \ 1\ \ \underbrace{0}_{b}$$

Ans = 2 → The first 1 that occurs from the R.H.S.

$N = a\,1\,b$ $\qquad$ $a = 101101$

$-N = \bar{a}\,1\,b$ → How $\qquad$ $b = 00$

Ans = N & (−N)

Negative of a number in Binary form

1 byte = 8 bits

10 ⇒ | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

MSB → Tells us if number is positive or negative

LSB → value of the number

- 1 → −ve
- 0 → +ve

−10 ⇒ ?

Steps: (1) complement of no.
(2) Add 1 to it

⇒ also known as 2's complement ~~num~~ number method

$(10)_{10} = (0000\,1010)_2$

complement $= 1\,1\,1\,1\,0\,1\,0\,1$

$\qquad\qquad\qquad + \quad 1$

$(1\,1\,1\,1\,0\,1\,1\,0)_2 \Rightarrow (-10)_{10}$

Ques Why ~~does~~ does 2's complement give negative number?

$10\,|\,\boxed{1011 0111}$

it'll be discarded if storing in 8 bits

$$\begin{array}{r} \underline{1}\ 0000\ 0000 \\ -\ 0000\ 1010 \\ \hline 0\ 1111\ 0110 \end{array}$$

what's this?

$1000 = 111 + 1$

$8 \quad 7 + 1$

$10000 = 1111 + 1$

$16 \quad 15 + 1$

$$\begin{array}{r} 1111 \\ +\ 0001 \\ \hline 1\,0000 \end{array}$$

$\underline{1}\,0000\,0000 \rightarrow 256$

$= \underbrace{1111\,1111}_{255} + 1$

NOW: $1111\,1111 + 1 - 0000\,1010$

$\Rightarrow \underbrace{11111111 - 00001010}_{\text{complement}} + 1$

$$\begin{array}{r} 1111\ 11\ 1\ 1 \\ -\ 0000\ 10\ 1\ 0 \\ \hline 1111\ 01\ 0\ 1 \end{array}$$

→ complement

Range of numbers

→ sign of the number

(01) 1 byte :

0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1

Total = 2 × 2 × 2 × 2 × 2 × 2 × 2 × 2

$= 2^8 = 256$ → total number of unique numbers that can be stored in 1 byte

Actual number is stored in bits = $(n-1)$

In 1 byte : 7 bits will be used to store numbers

Total 2 can make from 7 bits

$= 2^7 = 128$ bits

~~-128 to 128~~

→ 0 is here as well

Range : -128 to 127 (0 included)

0000 0000 → negative zero = zero

(1) ⟹ 1 111 1 1 1 1

\+ 1

1 0 0 0 0 0 0 0 0 → 9 bit

→ will be discarded .

Range formula

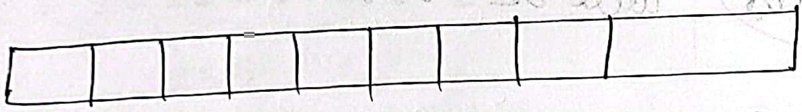
for n bits

$$\boxed{-2^{n-1} \text{ to } 2^{n-1}-1}$$

Ques Every number is appearing 3 times except one number. Find that one number

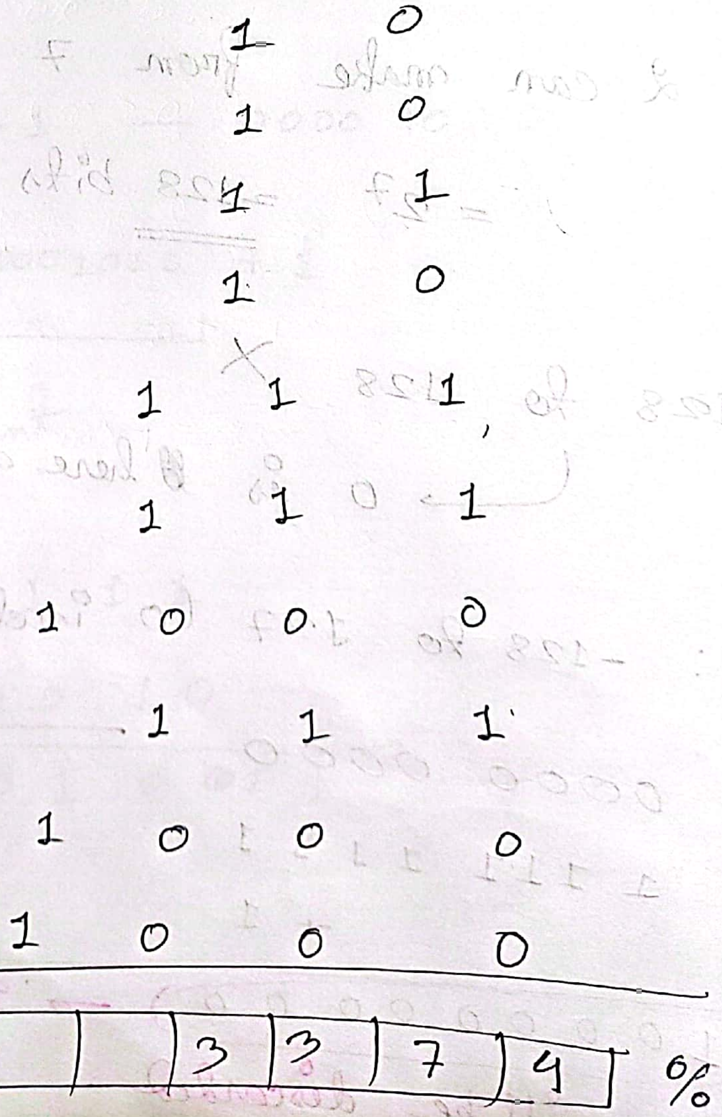
odd

nums = [2, 2, 3, 2, 7, 7, 8, 7, 8, 8]



If 3 is not presented

3	3	6	3



% 3

Every thing that was appearing 3 times there set bits will also appear 3 times. Hence there ~~sam~~ set bits will be modulo of 3.

But if an extra number is added then that will ~~be~~ not multiple of 3 or remainder will not be zero. Then, the remainder will be the answer

→ Amazon

Ques Find the N-th magic number. magic no.

5^3 5^2 5^1

1st magic number = 0 0 1 → 5

2nd " " = 0 1 0 → 25

3rd " " 1 1 → 30 (5+25)

4th " " = 1 0 0 → 125

5th " " = 1 0 1 → 130 (125+5)

⋮

Find N-th magic number

$n = 6 \Rightarrow 1\ 1\ 0$ $5^3 + 5^2 = 150$

loop [n & 1 ⇒ this will give me last digit in binary
n >> 1]

$0\times5^1 + 1\times5^2 + 1\times5^3$

```
int magicNumber(int n){
    ~~int / m = n~~
    int ans = 0;
    int base = 5;
    while (n > 0) {
        int last = n & 1;
        n = n >> 1;
        ans += last * base;
        base = base * 5;
    }

    return ans;
}
```

Ques: Find number of digits in base b.

$(6)_{10} = (110)_2 \rightarrow$ 3 digits

Formula

$\log_b a = x$

or, $a = b^x$

$\log_2 6 = x$

or, $\boxed{6 = 2^x} \rightarrow$ represent number of digits

$\log_2 10 = 3.32$

or, $10 = 2^{3.32}$

int + 1 = no. of digits

Formula: No. of digits in base b of no. n

$= \text{int}(\log_b n) + 1$

$$\boxed{\log_b a = \frac{\log_x a}{\log_x b}}$$

CamScanner

Pascal's triangle:

1					
1	1				
1	2	1			
1	3	3	1		
1	4	6	4	1	
1	5	10	10	5	1

Find the sum of n-th row:

Ans/ Sum of each row:

$$^nC_0 + {}^nC_1 + {}^nC_2 + \dots + {}^nC_n = 2^n$$

$1 << (n-1) = 1 \times 2^{n-1}$

Ans

Ques You are given a number. Find out if it is power of 2 or not.

1000000 , 100010 ✗
✓

approach 1

```
{    int n = 16;
     int cnt = 0;
     while (n > 0){
          if(n & 1) cnt++;
          n >>= 1;
     }
     if(cnt == 1) cout << "power of 2";
     else   cout << " Not power of 2\n";
}
```

approach 2

$$1000\ 0000 = \underbrace{1111111}_{(n-1)} + 1$$

$$\begin{array}{r} 1000\ 0000 \\ \&\ 0111\ 1111 \\ \hline 0000\ 0000 \end{array} \qquad \begin{array}{r} 100\ 10 \\ \&\ 011\ 11 \\ \hline 000\ 10 \end{array}$$

O(log(n))

Ans If $\boxed{n \& (n-1) == 0}$ // It is power of 2

Ques : Find $\boxed{a^b}$

// O(b)

$3^6 \Rightarrow 3\times3\times3\times3\times3\times3$

$3^6 = 3^{2+4} = 3^2 * 3^4$

$\longrightarrow (6)_2 = 110$

$= 3^{110}$

ans = ~~1~~ 9

base = 3

base = 9

81

base = base * base

n = 1 1 0

n & 1 ⇒ 0

n>>1

n = 11 & 1 ⇒ 1

n>>1

n = 1 & 1

ans = ans * base

O(log(b))

$3^{110} \Rightarrow 3^4 \times 3^2 \times 3^0$

Imp to understand

```
int base = 3;
int power = 6;
int ans = 1;

while (power > 0) {
    if ((power & 1) == 1) {
        ans *= base;
    }
    base *= base;
    power = power >> 1;
}
```

Ques Given a number, find the number of set bits in it.

$n = 9$

$n = (1001)_2$

Ans = 2

```
while (n>0){
    if((n&1)==1)
        cnt++;
    n >>= 1
}
return cnt
```

$n \& (-n) = 0001$

$n - [n \& (-n)]$

$$\begin{array}{r} n = 1001 \\ \&(n-1) \rightarrow 1000 \\ \hline 1000 \end{array}$$

$8 \& 7 \Rightarrow$

$$\begin{array}{r} 1000 \\ \& \ \ 111 \\ \hline 1000 \end{array}$$

$\boxed{\text{No. of set bits = no. of iterations}}$

Ques Find XOR of number from 0 ~~to~~ a.

a	XOR from 0 to a
0	0
1	$0 \wedge 1 = 1$
2	$0 \wedge 1 \wedge 2 = 3$
3	$3 \wedge 3 = 0$
4	$4 \wedge 0 = 4$
5	$4 \wedge 5 = 1$
6	$1 \wedge 6 = 7$
7	$7 \wedge 7 = 0$
8	$8 \wedge 0 = 8$

$[a \wedge 0 = a]$

$$\begin{array}{r} 01 \\ \wedge\ 10 \\ \hline 11 \end{array} \rightarrow 3$$

$[a \wedge a = 0]$

$$\begin{array}{r} 100 \\ \text{xor} \rightarrow (\wedge)\ 101 \\ \hline 001 \end{array}$$

$$\begin{array}{r} 001 \\ (\wedge)\ 110 \\ \hline 111 \end{array}$$

if $a \% 4 == 0$ ans $= a$;

else if $a \% 4 == 1$ ans $= 1$;

else if $a \% 4 == 2$ ans $= a+1$;

else if $a \% 4 == 3$ ans $= 0$

watch the pattern

Ques XOR of all numbers between a & b.

$a = 3$ & $b = 9$

$3 \wedge 4 \wedge 5 \wedge 6 \wedge 7 \wedge 8 \wedge 9$

$0 \wedge 1 \wedge 2 \wedge 3 \wedge 4 \wedge 5 \wedge 6 \wedge 7 \wedge 8 \wedge 9$

These are the extras

This is $0 \rightarrow (a-1)$

If we XOR these entire thing then it'll be removed.

$\boxed{f(b) \wedge f(a-1)} \longrightarrow$ **Ans**

$f(x) \rightarrow$ XOR of $0 \rightarrow x$

RangeXOR.cpp

↑ Leetcode

832: flipping an Image

1	1	0
1	0	1
0	0	0

horiz →

0	1	1
1	0	1
0	0	0

Invert →

1	0	0
0	1	0
1	1	1

solution → next page

```
vector<vector<int>> flipAndInvertImage
                        (vector<vector<int>>& image){

    int n = image.size();
    for(i = 0; i<n; i++){
        for(j = 0; j<n; j++){
            image[i][j] = image[i][j] ^ 1;
        }
        reverse(image[i].begin(), image[i].end());
    }
    return image;
}
```

$[a \wedge 1 = \bar{a}]$