Leetcode Problem

① Add digit (#258)

e.g ① 27

2+7 = 9

(ii)
```
10 | 62
10 |  6    2
   |  0    6
```

Ans=0 2+6 = 8

Ans = rem + ans

= 2+0 = 2

6+2 = $\boxed{8}$.

(iii) 249

2+4+9 = 17

= 8

we need single digit ans only.

```
int num;
while(num>9)
{ int ~~num~~, ans=0, rem;
  while (num != 0)
  {
      rem = num % 10;
      num = num/10;
      ans = ans + rem;
  }
      num = ans;
}
cout << ans;
```

② Leap year (GFG)

1600	✓
1700	X
1800	X
1900	X
2000	✓
2004	✓

(i) year which divides by 400 then that year is leap year.

(ii) year which divides by 4 but not with 100. is also leap year.

```
int year
 if( year % 400 ==0)
       cout <<"Leap year";
else if (year % 4 ==0 && year%100 !=0)
       cout <<" Leap year";
 else
     cout<< "Not leap year";
```

③ Reverse Integer

(i) i/p = 234 (ii) i/p 476
o/p : 432 o/p 674

rem = 234 % 10 = 4

ans=0 ~~rem + an~~
ans = ans×10 + rem
= 0×10+4 = 4
= 4×10+3 = 43
= 43×10 + 2 = 432.

```
int ans=0, rem;
while(num>0)
{
~~int ans=0, rem;~~
  rem = num%10;   ← last digit.
  num = num/10;  ←
  ans = ans×10+rem;
}
cout<<ans;
```

→ integer overflow

①

INT_MAX

INT_MIN

$$ans \times 10 + rem > INT\_MAX$$

$$ans > \frac{INT\_MAX - rem}{10}$$

$$ans > \frac{INT\_MAX}{10}$$

$$if\ (ans > \frac{INT\_MAX}{10})$$

return 0;

II

$$ans \times 10 + rem < INT\_MIN$$

$$if(\ ans < \frac{INT\_MIN}{10}\ )$$

return 0;

(4) Power of 2

$2^1 \to 2$ yes

$2^2 \to 4$ yes

8 yes

15 → NO

Num<1 ← NO ← Negative number.

(i)

1 —	1 ~~0~~	← 2^0
2 —	10	← 2^1
4 —	100	← 2^2
8 —	1000	← 2^3
16 —	10000	← 2^4
32	100000	← 2^5

(ii)

2   31

4

8

16   64

32

divide num by Power of 2 till num exided or reached that num.

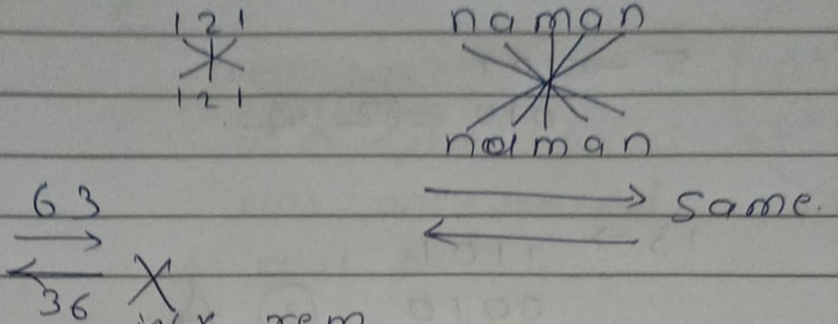
```
while ( x != 1)
{
    if( x% 2 == 1)
        return 0;
    x /= 2;
}
return 1;
```

5 sqrt x   H/W

$4 \rightarrow \sqrt{16}$

$\sqrt{25} \rightarrow 5$

(6) Pallindrome



```
int x, rem.
 ans = 0
   num = x;
while ( num 0 )
{
     rem = num%10;
      num = num/10;
     ans = ~~rem~~ ansx10 + rem;
 }
 if ( x = = ans)
        return 1;

 if( ans > INT_MAX/10 || ans< INT_MIN/10 )
        return 0;
```

⑦ Complement of a no.

5 → 101
     010 = ② A.

13 → 1101
       0010 = ② A

27 → 11011
       00100 = ④ A.

2	27	rem	0		
2	13	1		2^0	0
2	6	1	0	2^1	0
2	3	0	1	2^2	4
2	1	1	0	2^3	0
	0	1	0	2^4	0
					④

2	13	rem			
2	6	1	0	2^0	0
2	3	0	1	2^1	2
2	1	1	0	2^2	0
	0	1	0	2^3	0
					②.

```
ans = 0, rem, mul = 1
while (n)
{
      rem = n % 2 ;
        n /= 2;
      rem = rem ^ 1;
      ans = ans + rem x mul;
      mul *= 2 ;
  }
    return ans ;
```

0 ^ 1 = 1
1 ^ 1 = 0

edge cases
n = 0 , return 1

```
  if( n = = 0)
      return 1;
```