①

(Wednesday)

→ Sum of Digits

let sum=0 + + +

① num=456

int digit= 456%10 = 6

↓

by doing this we get last digit

② num= 456/10 =45

digit = 45%10 = 5

num= 45/10 = 4

③ digit= 4%10= 4

num= 4/10=0

↳ stops

→ num!=0

↳ digit= n%10

↳ Sum= Sum+ digit

↳ num=num/10

②
→ Return the smallest digit from the Number.

I/p Num = 420   4
                2
                0

O/p: → 0

Rule:

Integer → Range ⓧ → ⓨ
min ↓ INT-MIN   max ↓ INT_MAX

5 7 9 4 6

if
① 5 < INT_MAX

int minimum = INT_MAX

= ~~5~~ ④

***

① To find out Smallest number we Intialize with INT_MAX

② To find Out Largest number we Initialize with INT-MIN

③ Include header file
#include <limits.h>

int min = INT-MAX; ← compare.

num = 5289

digit = num%10 = 5289%10 = ⑨

num = num/10 = 5289/10 = 5289

digit = 528 % 10 = 8 → [Compare with min Vae]

num = 528 / 10 = 52

digit = 52 % 10 = 2

num = 52 / 10 = 5

digit = 5 % 10 = 5

num = 5/10 = 0 → Stop

→ Inbuilt function

min (a, b);
max (a,b);
} we can use these directly to get min, max.

→ Note:

- Findout the best alternative other than '%'. because % is an heavy operation, % is not a best practice.

③ → Check if a number is Even or odd.

Even	odd
2	1
4	3
6	5
8	7
10	9

multiples of 2

12 → rem → 0

%2 == 0

12 → rem → 1

%2 == ①

```
if (num%2==0)
{
    "Even"
}
else
{
    "odd"
```

→ Alternate method

2 → 00000010

4 → 00000100

6 → 00000110

8 → 00001000

10 → 00001010

Even: last bit 0 & 1 → 0 → Even

odd: last bit 1 & 1 → 1 → odd

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

→ Note:

- The Best approach is Using Bit-wise approach. we can use this in any problem because there is no best approach other than bit wise.

★★★

→ Reverse a Number: (Asked in Online Test, Interview)

I/p → n = 123
O/p → 321

I/p → n = 421
O/p → 124

I/p → n = 479
O/p → 974

Approach:

① n = 123

n % 10 = 3

$n = \frac{123}{10} = 12$

n%10 = 12%10 = 2

$n/10 = 12/10 = 1$

n%10 = 1%10 = 1

$n/10 = 1/10 = 0$
↳stops

321

300 → 3×10^2
+ 20 → 2×10^1
1 → 1×10^0

int ans = 0;

① $0\times10+3 = 3$

② $3\times10+2 = 32$

③ $32\times10+1 = 321$

② int ans = 0;

n = 576
%10 → 6 $0\times10+6 \rightarrow 6$

n = 576/10 = 57
%10 = 7 $6\times10+7 = 67$

n = 57/10 = 5
%10 = 5 ~~[illegible]~~
$67\times10+5 = \underline{675}$

n = 5/10 = 0
↳stop

```
while(num!=0)
{
  int digit= num%10;
  ans = ans*10+digit;
  num =num/10;
}
return ans;
```

alternate:

```
for(  ; num!=0; num=num/10)
{
  int digit = num%10;
  ans=ans*10+digit.
}
return ans.
```

→ Reverse an Integer.

⑤

Algorithm:

↳ ① 0 wala case handle Kardo

② Sign ki Info flag me rakhlo

③ Sign remove Krdo [Use: abs(num) for removing sign]

④ num reverse Krdo

⑤ reverse no ke andar sign lagado

⑥ return Kardo

1→+ve

0→-ve

Ex: n=-123

flag→variable.

```
if(num<0)
  flag=0
```

$\boxed{flag = 0}$

n = 123

ans = 321

ans = -321 → return

→ Prog: (Dry Run):

```
if(n==0) return 0;                ① n = -56              n = 123456789
boolean flag = 0;                 ② flag = 0 ✓           flag = 0
if (n>0)
    flag = 1;                        Y                   flag = ~~0~~ 1
n = abs(n);                       n = abs(-56) = 56
long long int ans = reverseNumber(n);   ans = 65         ans
                                                         = 9 - - - 1
if (flag==0)                         Y
    ans = 0-ans;      } —          ans = -65
                                 ←                →
// range check →                 ●―――――――――――――――――――――●
                                 INT-MIN          INT-MAX   ans > INT_MAX
    if ( ans > INT_MAX || ans < INT_MIN)                    return
        return 0;
    return ans;          — -65
}
```

$\boxed{}$ ~~int~~ long long int

Value

abs → Exact value either It is +ve or -ve.

Ex: abs(5) = 5

abs(0) = 0

abs(-5) = 5

can we make our own function if we have any confusion.

```
int absolute value (int n)
{
    if (n>0)
        return n;
    else
        return (0-(n))
}
```

→ Palindrome Number: (Asked in Online Test, Interview)

(6)

L → R = R → L
(reverse)

Ex: $\overrightarrow{\underleftarrow{12321}}$

(1) reverse the Given number

(2) Compare with that reverse. (org == rev)
↳ then It is a palindrome.

```
String  isPalindrome (int num) {
    int orginal = num;
    int rev= revreseNumber(num);
    if (orginal = =rev) {
        retuen "Yes"
    }
    else {
        retuen "No";
    }
}
```

→ Numbee is a powee of 2 or Not

(7)

0 → doesnot have Set bit

$2^0 \longrightarrow 1 \longrightarrow$ 0000 - - . 0001

$2^1 \longrightarrow 2 \longrightarrow$ 0000 - - - - 00(1)0

$2^2 \longrightarrow 4 \longrightarrow$ 0000 - - - -0(1)00

$2^3 \longrightarrow 8 \longrightarrow$ 0000 - - -(1)000

$2^4 \longrightarrow 16 \longrightarrow$ 0000 - - -(1)0000

} 2^x Having only 1 set bit

I/p → n ↳ No of set bit = =1 → Can be represented as 2^x (or) power of 2

→ How can I find Single Set bit

$$- n \& (n-1) \text{ gives } [\overset{\text{result as 0}}{\text{[illegible]}}]. \text{ If 0 then It is power of 2}$$

$n=8 \rightarrow 00 - - - 001000$

$\& \; (n-1) = 8-1 = 7 \rightarrow 00 \cdots 0111$

$0000 \cdots 0000$

$n = 16 = 0000 - - - 10000$

$n = 15 = 0000 - - - 01111$

$0000 - - - 00000$

→ If we cant makeup formula then

Hw 32 bit
↳ count Set bit by using >> operator with O(1) time

→ Check if a Number is prime or Not.

* Prime Number

↳ have factors 1 and Itself

$13 \rightarrow 1, 13$

$2 \rightarrow 1, 2$

* 1 – Neither prime nor Composite

Prog:

```
if (num==1)
{
   return "No";
}
if (num==2 ||num==3)
{
   return "Yes"
}
int Start = 2;
int end = n-1;

for ( int i=start; i<=end; i++)
{
      if (num%i ==0)
      {
           return "No";
      }
}  return "Yes";
```