Assign-1.

1.] count digits in a number
given an integer N return the no. of digits in N.

⇒
```
Public void count_all_digit( int num){
        int count = 0;
        while(num >0){
           int rem = num % 10;
           ~~num = num/10 count = count + rem~~
           count ++;
           num = num/10;
        }
    Public static void main (String args[]){
        int n = 12345;
        count_all_digit (n);
    }
```

⇒ ~~For~~ i) num > 0
12345 > 0
rem = 12345 % 10 = 5
count = 1
num = 1234

ii) num > 0
1234 > 0
rem = 1234 % 10 = 4
count = 2
num = 123 ;

iii) num = 123
123 > 0
rem = 123 % 10 = 3
count = 3
num = 12

iv) num = 12
12 > 0
rem = 12 % 10 = 2
count = 4
num = 1

v) num = 1
1 > 0
rem = 1 % 10 = 1
count = 5 ;
num = 0

vi) num = 0
0 > 0   condition false while loop
ends here.

Date 1/2/1

2). check if number is palindrome or not.
A Palindrome number that reads same backward as forward e.g 121.

algorithm:-

```
public void check_palindrome ( int num) {
    int temp = num; int rem ;
    ~~while (num >0)~~
   int count = 0 ;
    while( num>0) {
    ~~int~~ rem = num% 10
    Count = count *10 + rem;
    num = num / 10
    }
  if ( count == temp) {
    S.OP (" Palindrome number");
   }
  else{ print("Not Palindrome");
    }
```

logic:-

```
i)  num= 4554             2) 455>0               3) 45>0
temp = 4554   ───→        temp = 4554   ───→      temp=4554
~~while~~( 4554 >0         (455>0 )               (45 >0)
 rem= 4554%10 ⇒ 4         rem= 4554%10 ⇒ 5       rem= 454%10 ⇒ 5
 count= 0*10+4 ⇒ 4        count= 4*10+5 ⇒ 45     count= 45*10 ⇒ 5
 num= 455                 num= 45                count= 455
                                                 num= 4

  4) 4 >0                 num=0
   temp= 4554
    (4>0)
   rem= 4%10 ⇒ 4
    count= 455*10+4
    count = 4554
```

24/04/2025

3) Find GCD of two numbers.

problem statements:- given two integer N_1 & N_2 find their greatest common divisor.

algorithm:-

```
public void find_gcd (int n1, int n2){
        int temp=0;
    for(int i=1; i<n1/2 ; i++){

        if( n1%i== 0 && n2%i==0){
            temp=i;
        }

        print(temp);

    }
```

logic:-

```
n1=9, n2=12
temp=0;
for(int i=1; i<(9/2)=4 ; i++){
    }
    ↓
for(int i=1; i<4; i++){

if(n1%i==0&& n2%i==0){
    temp=i
}
```

```
→ for i=1;
  (9%1 ==0 && 12%1=0)
     temp=1;
     i++;

  i=2
  (9%2==1 && 12%2==0)
     temp=1;
     i++;

  9%3==0 && 12%3==0
     temp=3;
```

∴ greatest common factor is 9&12 is 3

EDMI 13C 5G 24/04/2025 19:1

4) check if a number is Armstrong ~~Number or Non.~~

```
Public void ~~boolean~~ is_armstrong( int num) {
    int count = 0; ,int rem; ~~int num2~~
    while (num > 0) {
        rem = num % 10 ;
        count ++;
        num = num / 10;
    }

    ~~check_armstrong(int num)~~
    check_armstrong(num, count)
~~if~~

}
Public boolean ~~void~~ check_armstrong (int num, int count) {
    int temp_num = 0; int rem;
    int Original_num = num;
    while( num > 0) {
        rem = num % 10
        temp_num = tempNum + Math.Pow(rem, count)
        num = num/10 ;
    }
    if ( temp_num == original_num) {
        return True
    }
    else {
        return False
    }
}
```

```
logic = 153.                          => while(15>0){
  is_armstrong(153)                      rem= 15%10=5
  {int count=0, rem;                     c ~~count+~~ count=2
  while(153>0){                          num=1
  rem = ~~num%10~~ 153%10                  ↓
     rem  =3                             (1>0){
     count =1;                           rem= 1%10=1
   num=15,                                 count=3
                                           num=0

   ↓
// call check_armstrong() function
  check_armstrong (153, 3){
   int temp_num=0; rem;
   int original_num= num;
   while(num>0){
    rem= 153 %10=3
   temp_num= 0+ Math.pow(3, 3)
   temp_num= 0+27
     temp_num=27
     num=15,
while(15>0){                          if(temp_num
                                          == original_num)
   rem= 15%10=5                       {
   temp_num= 27 + Math.pow(5,3)        return True;
             = 27+125                  }
  temp_num= 152                       else{
   num=1
                                        return false;

   while(1>0){
   rem= 1%10=1
    temp_num= 152+ Math.pow(1,3)
```

GoodLuck Page No.
Date

Q5] print all divisiors of a given Number.

given an integer N return all divisiors OF N.

```
public void divisiors_of_num( int num)
{
        for(int i=1; i<=num/2 ; i++){
            if(num % i == 0){
            print(i+" ");
            }
        }
        Print (num);
}
```

logic = if Num=12

```
for( i=1; i<=12/2=>6 ; i++)
    ↓
for( i=1; i<=6 ; i++){
    if(12%1==0){
        -> print(i) => 1
    }
    ↓
i=2; i<=6 ; i++{
(12%2)==0 {
    print(2)
}
(i=3; i<=6 ; i++){
    (12%3==0) {
    Print(3);
    }
}
```

```
(i=4 ; i<=6; i++){
    (12%4==0){
        Print (4);
    }
(i=5; i<=6; i++){
    (12%5 == 1) -> false
    // don't print
}
(i=6 ; i<=6 ; i++) {
    (12%6==0) {
        Print(6)
    }
```

24/04/2025 19:15

Scanned with OKEN Scanner

6) Check number is Prime or not.

```
⇒  public boolean check_Prime(int num){
        int count=0;
      for(int i=1; i<=num/2; i++){        → if(num==2){
           if(num%i==0){                      return True;
               count=count+1;                 }.
           }
       }

      if(count>=2){
           return false;
      }
      else{
           return True;
```

```
logic= If Num=2
  For(i=1; i<=2; i++){
    if(2%1==0){
        count=1
    }
    ↓
  for(i=2; i<=2; i++){
    if(2%2==0){
        count=2
    }
  if(count>=2){
     return false;
  }
  else{
     return True
```

```
1) Num=2
⇒ if(num==2)
     (2==2){
       return true;
     }

2) Num=10;
   (10%1)==0{
     count=1;
   }
   (10%2)==0{
     count=2;
   upt:..
   (10%5)==0{
     count=3;
   }
```

24/04/2025 19

2/2/25