Assignment 1.

Q1) Count digits in number.

```
int count = 0;
while(n!=0) {
        // n = 101
 int remainder = n %10; // 1

 i~~nt~~ n = n /10; // 10

        count ++ //1
   }
   SOP (count)
}

while (n!=0)
        ~~rep~~ 10 ≠ 0
 int remainder = n%10; // 0
    n = n/10  ; // 1
        count ++ //2
   }

while(n!=0)
        // 1 ≠ 0.
int remainder = n %10 //1
   n= n/10 ; // 0
    count ++ //3
   }
while (n!=0)
        0 = 0
```

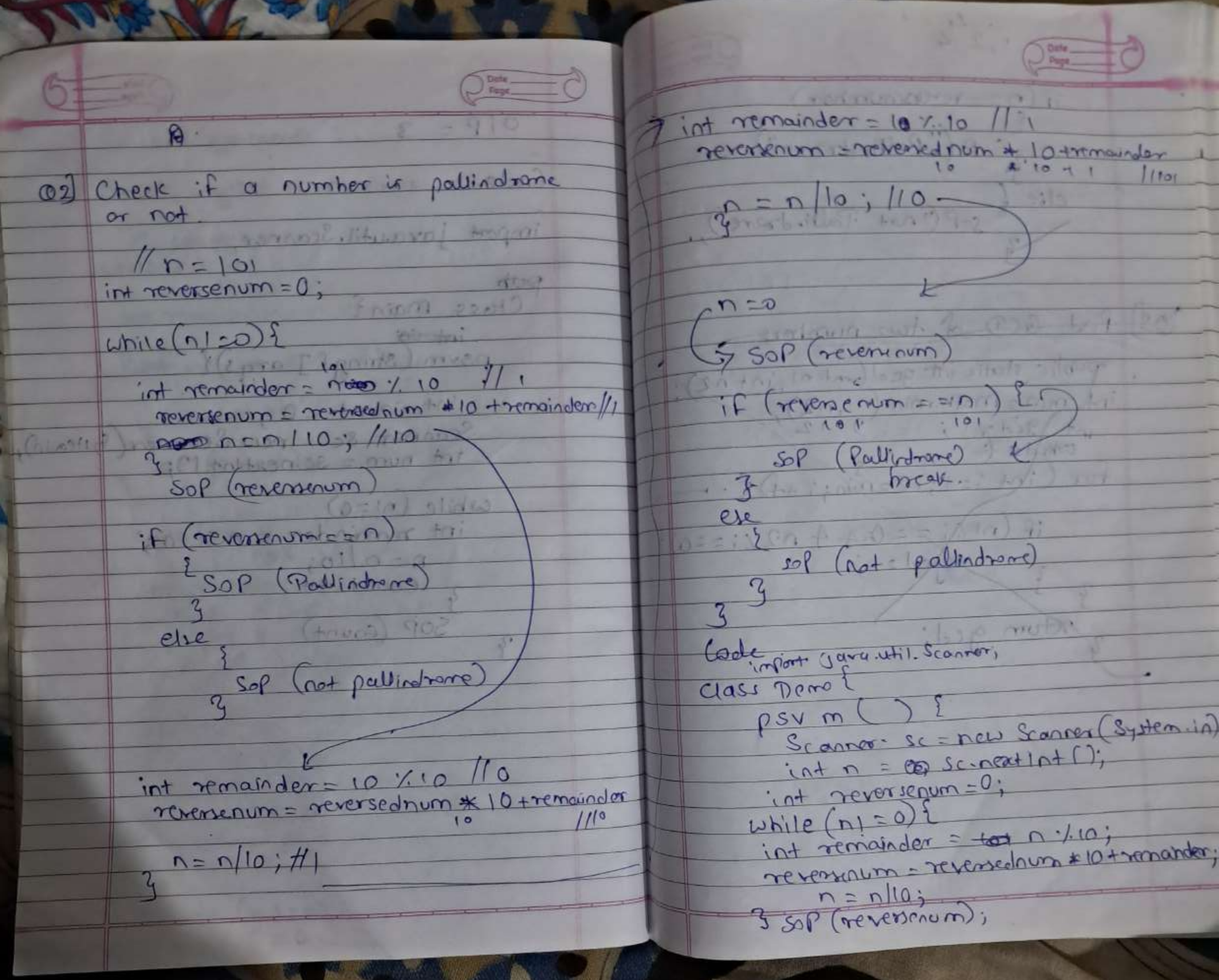
O/P = 3.

Code.

```
import java.util.Scanner;

~~pub~~
Class Main{
    ~~int C~~
    psvm (String[] args){
      int Count = 0;
     Scanner sc = new Scanner(System.in);
      int num = sc.nextInt();

    while (n!=0)
      int remainder = n % 10;
         n=n/10;
         count ++
      }
       SOP (count)
  }
```



A.

Q2] Check if a number is pallindrome or not.

```
  // n = 101
int reversenum = 0;

while(n!=0){
                      101
  int remainder = n % 10     // 1
   reversenum = reversednum *10 +remainder//1
     n=n/10; //10
  }
   SoP (reversenum)

 if (reversenum==n)
   {
     SoP (Pallindrome)
    }
  else
      {
       Sop (not pallindrome)
      }


 int remainder= 10 %10 //0
  reversenum = reversednum * 10+remainder
                          10             //10

    n= n/10; //1
 }
```

```
→ int remainder = 10 % 10 // 1
   reversenum = reversednum * 10+remainder
                        10    * 10 + 1    //101
    n = n/10; //0
  }

   n = 0

   → SoP (reversenum)

   if (reversenum ==n ) {
           101       101
         SoP (Pallindrome)
              break.
   }
   else
     {
        sop (not pallindrome)
     }
 }
```

Code
```
import java.util.Scanner;
class Demo {
    psv m ( ) {
      Scanner sc = new Scanner(System.in)
      int n = sc.nextInt();
     int reversenum = 0;
    while (n != 0){
     int remainder = n %10;
     reversenum = reversednum * 10+remander;
          n = n/10;
    } SoP (reversenum);
```

1,2,4
4

Date
Page

```
if(n==reversenumber)
   {
    SoP ("Pallindrome")
   }
else {
       soP ("not Pallindrome"),
   }
```

Q3] Find GCD of two numbers.

```
public static int gcd(int n1, int n2);
int min = Math.min (n1, n2);
                    9   12
  int gcd = 1;
  ~~while {~~
for (int i=1; i<=min; i++) {
          if (n1%i==0 && n2%i==0) {
                gcd = i;
          }
   }
    return gcd;
}
```

Q3 Find GCD of

```
while (n1>0
     if (n1>
       n
     }
     else
       n
  if (n1==
     gc
  else
     gc

  if (n1
     n1
  }  n1
  else
     n

  if(

  else
```

Q3 Find GCD of two numbers.

```
         9            12
while (n1 > 0) && n2 > 0) {

    if (n1 > n2) {               9 ≯ 12
         n1 = n1 % n2;
    }         9 % 12       =
    else
         n2 = n2 % n1      = 3
         n2 = 12 % 9       = 3
          n2 = 3
  if (n1 == 0)
       gcd = n2;
  else
       gcd = n1;
                                   3

  if (n1 > n2) {
      9    3
       n1 = n1 % n2;
    }  n1 = 9 % 3  = 0
  else
       n2 = n2 % n1                  n1 = 0
                                     n2 = 3

     if (n1 == 0)
          gcd = n2
     else
          gcd = n1                 3

                 3
```

Code:-

```
import java.util.Scanner;

~~po~~
Class Demo {

    public static void main (String args[])
    {
     Scanner sc = new Scanner ();
     int n1 = sc.nextInt ();
     int n2 = sc.nextInt ();

   while (n1>0 && n2>0) {

       if (n1 >n2)
       {
         n1 = n1 % n2 ; }
      else {
         n2 = n2 % n1 ,
       }

     if (n1==0) {

         return gcd = n2 ;
      }
    else
        return gcd = n1;

    }.
}
```

7. Check if a number is Armstrong Number

~~int dig~~

```
count = by function calling
int sum =0.
          153
while (n > 0) {
   int remainder = n % 10   // 3
   sum = sum + remainder ^power(count)
   n = n /10;  //15   //0 + 3^3 = 27
}
if (sum == n)
     Sop ("Armstrong");
else
     SOP (" no' Armstrong);

        15
while (n >0)
   int remainder = n % 10       15 % 10 = 5
   sum = sum + remainder ^power (count)
// = 27 + 5^3 = 125+27 = 152
     n = n / 10;  //1

While (1>0)
   int remainder = n%10 = 1 % 10 = 1
   sum = sum + remainder ^3
      // 152 + 1^3 = 153
      n = n / 10 ; //0
while (n>0)   0 ≯ 0
```

```
if (sum == n)
    153   153
Sop (Armstrong)
```

Code.

```
import java.util.Scanner;

Class Demo;


public static void main() {
    Scanner sc = new Scanner;
    int n = sc.next int
      count (n)
      isArmstrong (n)

    }

           int
static ~~void~~ count (int n) {
  int Count = 0;

     while (n>0) {

       int remainder = n % 10;
       n = n/10;
       count ++;
     }
~~sop (count)~~
     return count;
static int isArmstrong (int n) {
int digits = count(~~int~~ n);
int sum = 0;
while (n > 0) {

    int remainder = n % 10;
    sum = sum + (int) math.pow(remainder,
                               digits);
    n = n / 10;
  }
  if (n = = sum) {
       sop ("is Armstrong")
   }
  else
       sop ("not Armstrong");
   }
}
```

5. Print all Divisors of a given number.

```
                 n = 36
                        6
for (int i=1, i<=sqrt n; i++){
        36% 1 = 0
     if (n%i==0){
          SOP(i);  1 ~~36~~
     if (i != n/i){
          print (n/i);        36/1 = 36
     }
  }
}
                              1+1, 36

         n %i
if (36%2==0){
     SOP(i)  // 2
   if (i != n/i)
   sop (n/i)  //18                2+1 = 3

                                  18
     36%3
if (n%i ==0){                       n×i = n
    sop (i) // 3                      12
  if (i != n/i)  36/3
    sop(n/i) // 12
```

Date
Page

```
      36 % 4
→ if (n % i) {
     sop (i) // 4
     if (n != n/i)
     {
        sop (n/i) ~~36~~ 9
     }

      36 % 5
   if (n % i) {   X
      sop (i)
      if        i++

   if (36 % 6) {
       n   % i
      sop (i) = 6 →
      if (i != n/i)
         6 ≠ 6
```

o/p
1
36
2
18
3
12
4
9
6

Code:

```
import java.util.Scanner;

public static void main (){
  Scanner sc = new Scanner(System.in).
   int n = sc.next int ().
 factors ~~factorial~~ (n);
}
static void ~~int~~ factors (int n){
    for (int i=1; i<=Math.sqrt(n); i++){
       if (n%i==0) {
          sop(i).
       }
       ~~else~~ if (n!=n/i){
          sop(n/i);
       }
    }
  }
}
```

O/P

```
3|6
 2
1 8
 3
1 4
19 7
 6
```

```
1
36
2
18.
3
12
4
9
6
```

Q8 Check
int c
for(i

```
){
(Systemin).
{
nt(n); i++){
```

Q8 Check if a number is prime or not.

```
 int count =0
                                  9
for(int i=1; i<=Mathsqrt(n); i++){
          9%1==9
     if (n%i == 0){
       Sop ~~Sop(i)~~ 1
       
       if (n% = n/i) {
         Sop ~~Sop(n/i)~~; 9
         }
     }
  }
  count ++
  if (count ==1)
    print(prime)
  else
    print(not prime).

will eterate
3 times
if n=9
√n= 3

      9%3 ==0
  if (n% i7 ==0)

   if(n=       4:1)

   [Count == 3]
   if (count(==1)
       prime
   else (not prime)
```

Code:-

```
import java.util.Scanner;
class Demo{
    psvm () {
    Scanner sc = new Scanner(System.in);
     int n = sc.nextInt();
      factors (n);
    }

 static void factors (int n){
        int count = 0;
    for(int i = 1; i<= math.sqrt(n); i++){
            if(n%i==0') {
              ~~sop(i)~~

            if (n != n/i){

               }
            }
        Count ++;
    }
}
   count ++;
}
```

```
if(count==1){
 sop("is prime");
}
else {
  sop("not prime");
 }
```