20) To print Odd number series.

→

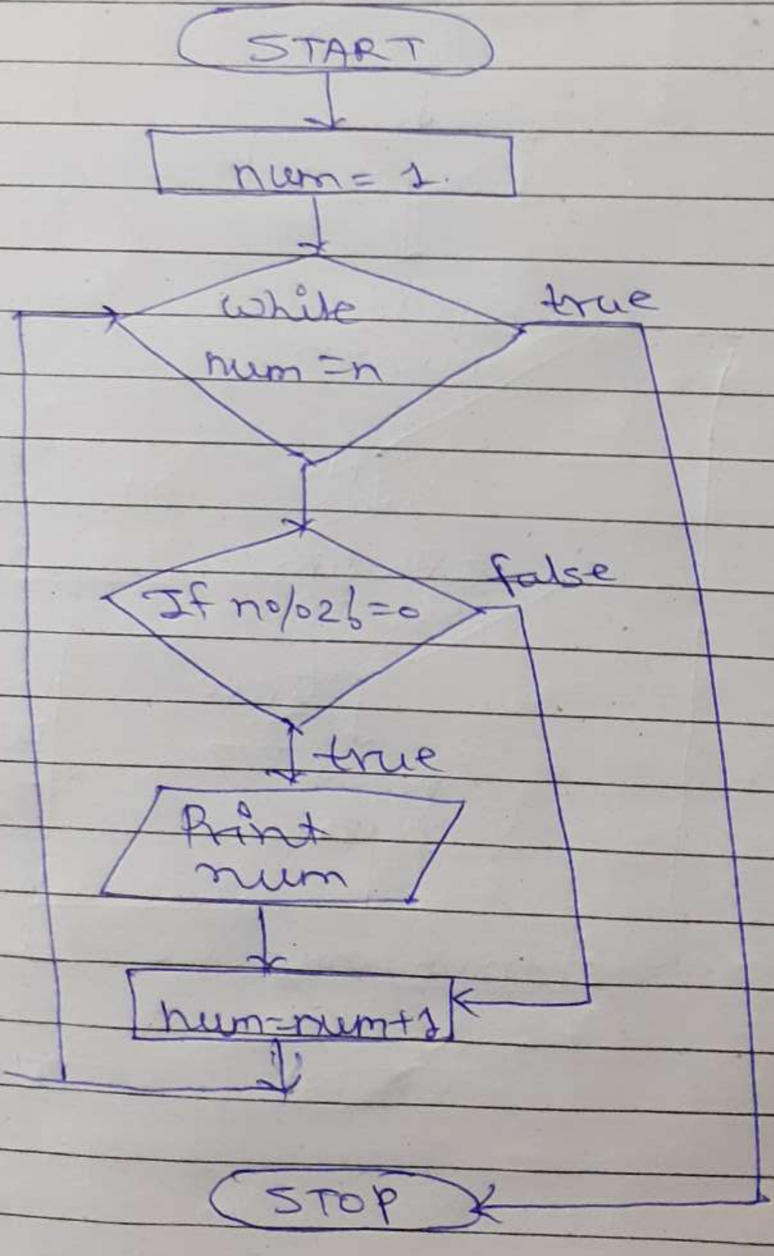
Step 1 : Start

Step 2 : Read input.

Step 3 : Iterate while loop according to user input

Step 4 : Inside loop If n%2!=~~0~~0 print n.

Step 5 : Stop.

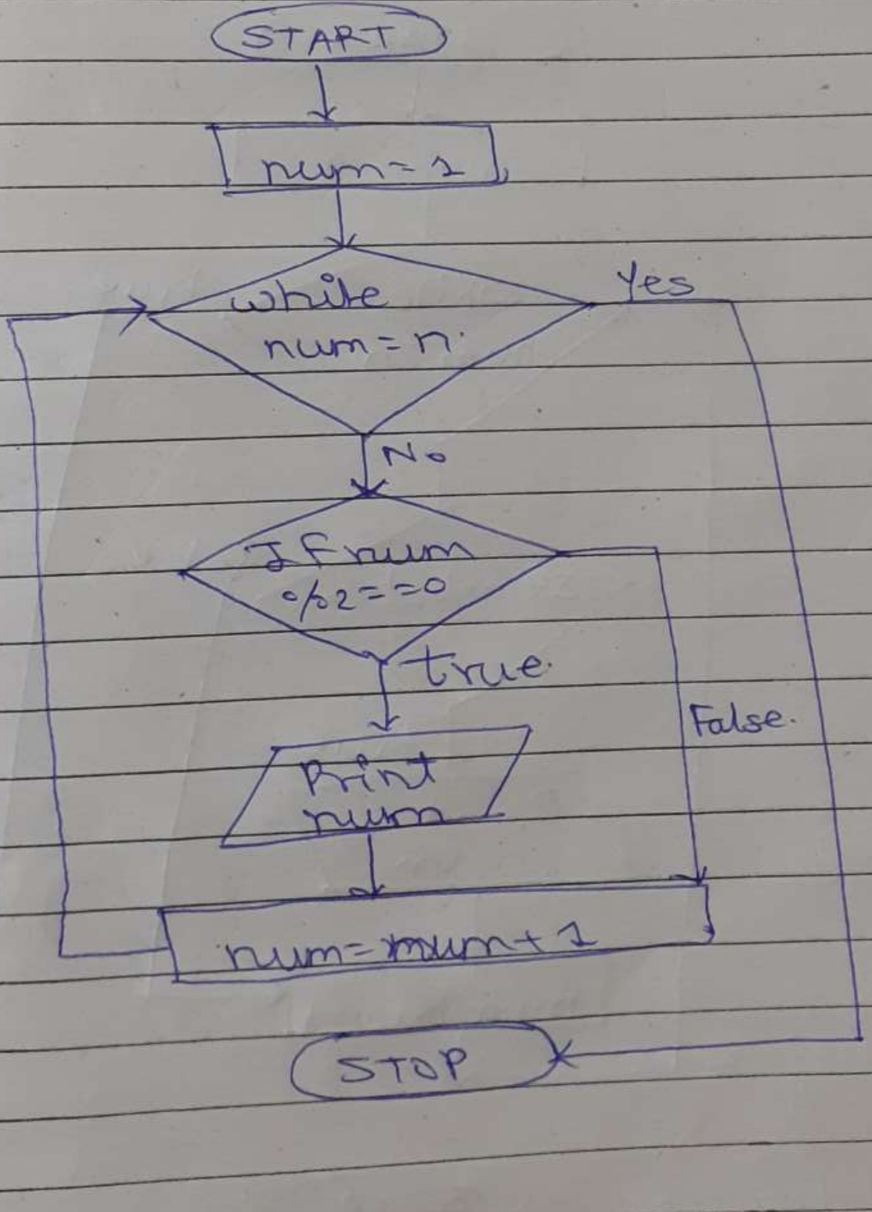


Page No.
Date

19.) To print Even number series.

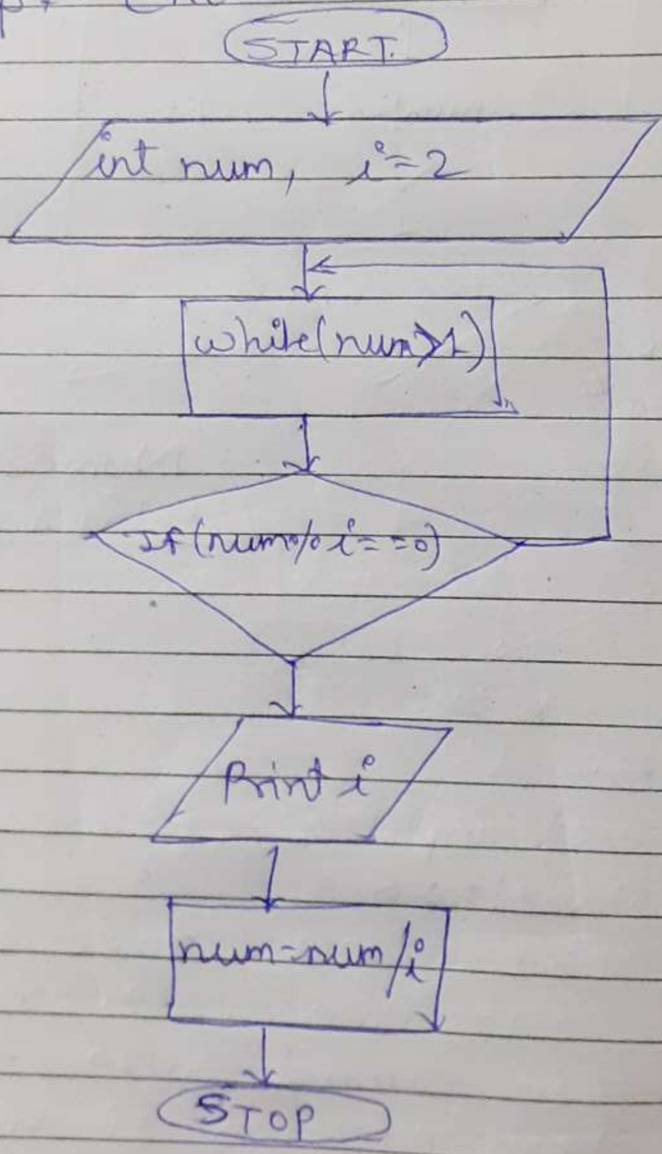
→

Step 1 : Start
Step 2 : Read input a
Step 3 : Iterate for while loop according
    to user input.
Step 4 : Inside loop If n%2 ==0 print
    even number.
Step 5 : Stop

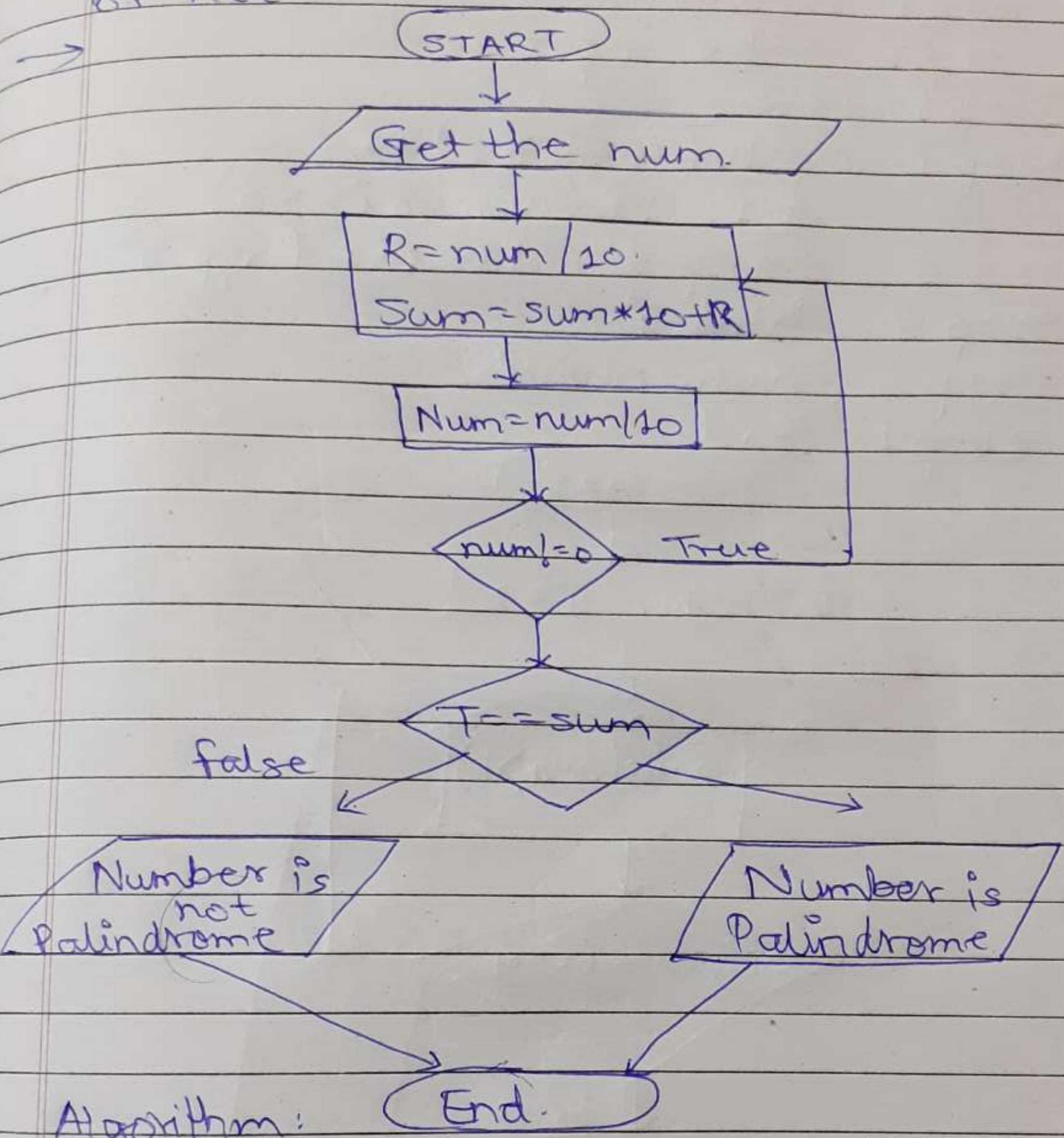


18.) To print prime factors of given Number

```
→ Step1: Start
  Step2: Take input
  Step3: i=2.
  Step4: while(num>1) →
           4.1: If (num%i==0) go to
                step 5. else i++ & repeat step4
  Step 5: Print i.
  Step 6: num=num/i.
  Step7: End.
```



17) Check whether given No. is Palindrome or not.



Algorithm:

Step 1: Start:

Step 2: Get input & read input.

Step 3: while!=10

3.1: r=num/10

3.2: sum=sum*10+r.

3.3: num=num/10.

Step 4: Check T==sum If true go to step 5 else step 6.

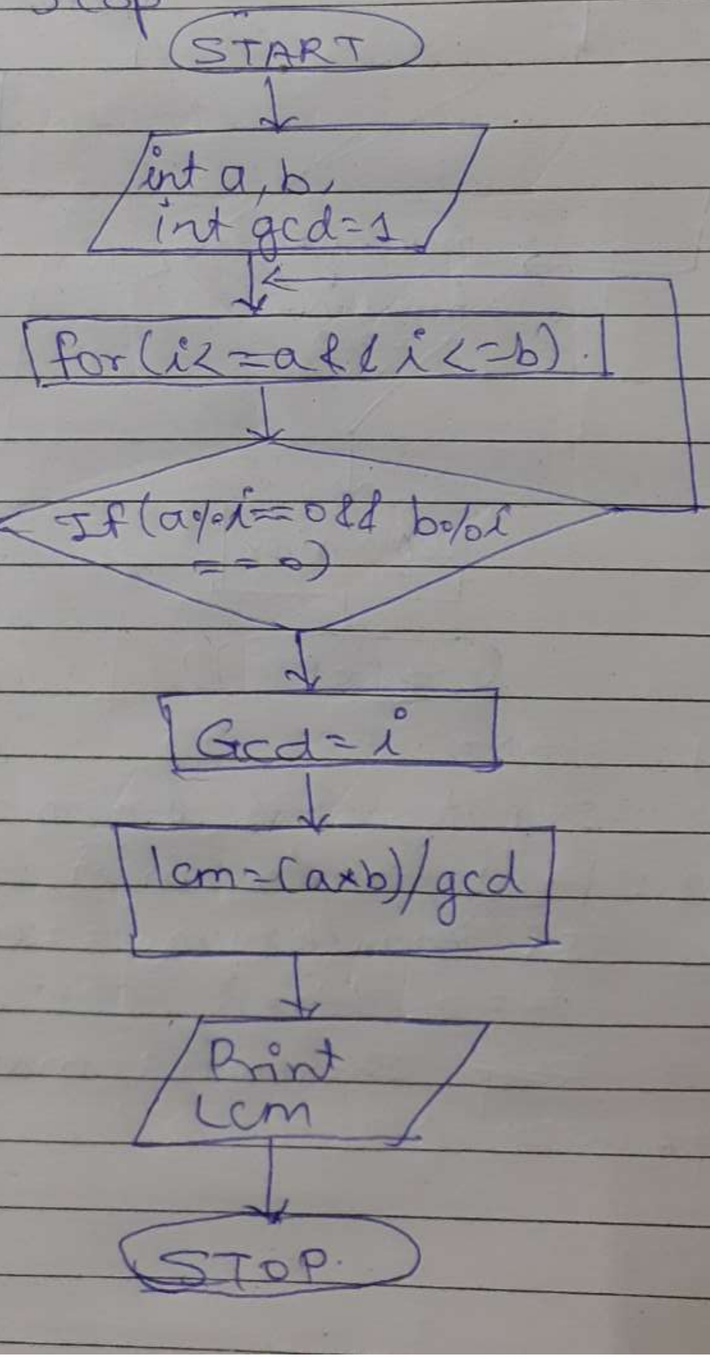
Step 5: Print number is Palindrome

Step 6: Print Not Palindrom.

Step 7: Stop

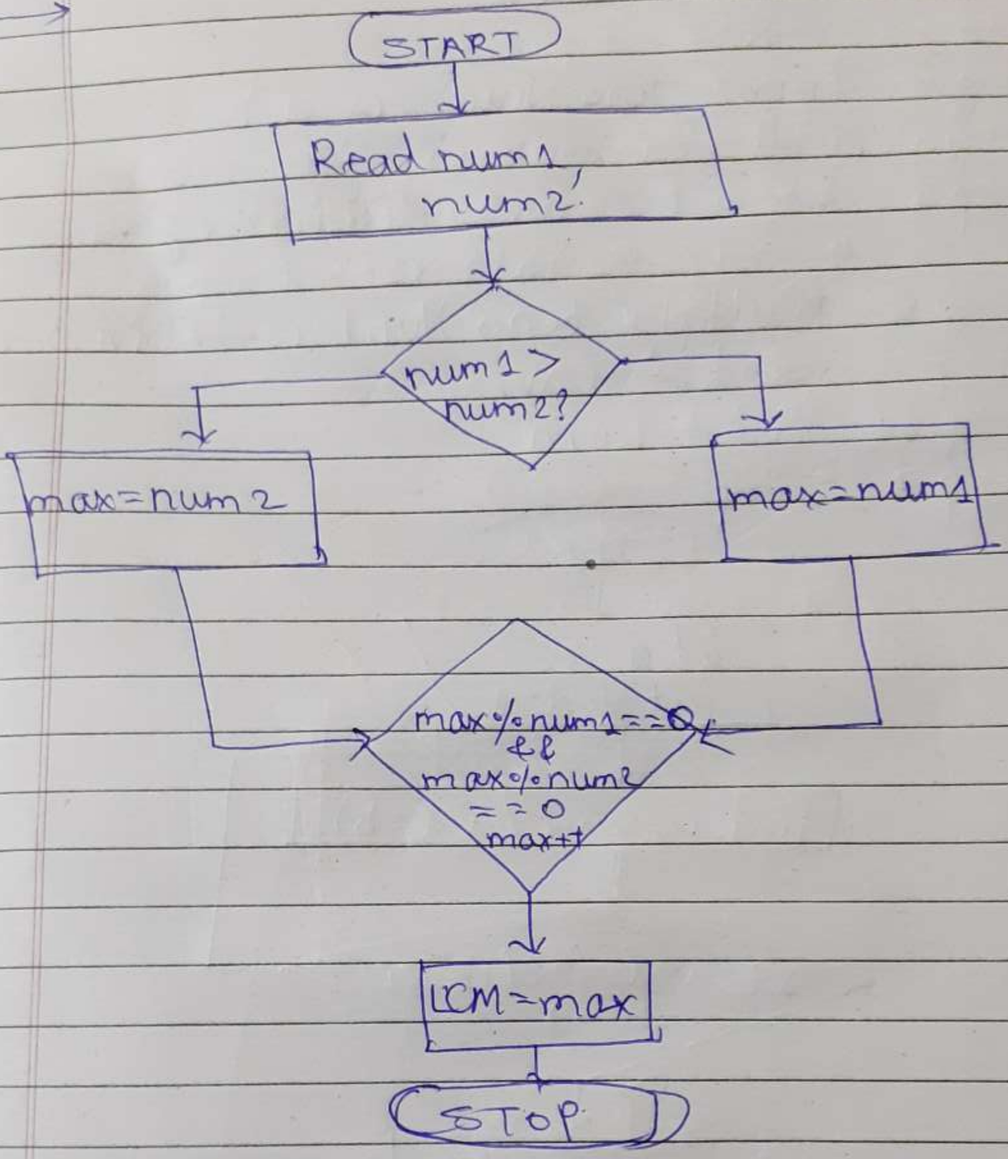
16) LCM of two numbers using the Prime factor method.

→ Step 1: Start
Step 2: Input Two Numbers a, b
Step 3: Find factors of given numbers
Step 4: If a & b are divisible by Gcd
equals to factors
Step 5: Multiply two Numbers → Divide by
Gcd = LCM
Step 6: Print LCM
Step 7: Stop



Page No.
Date

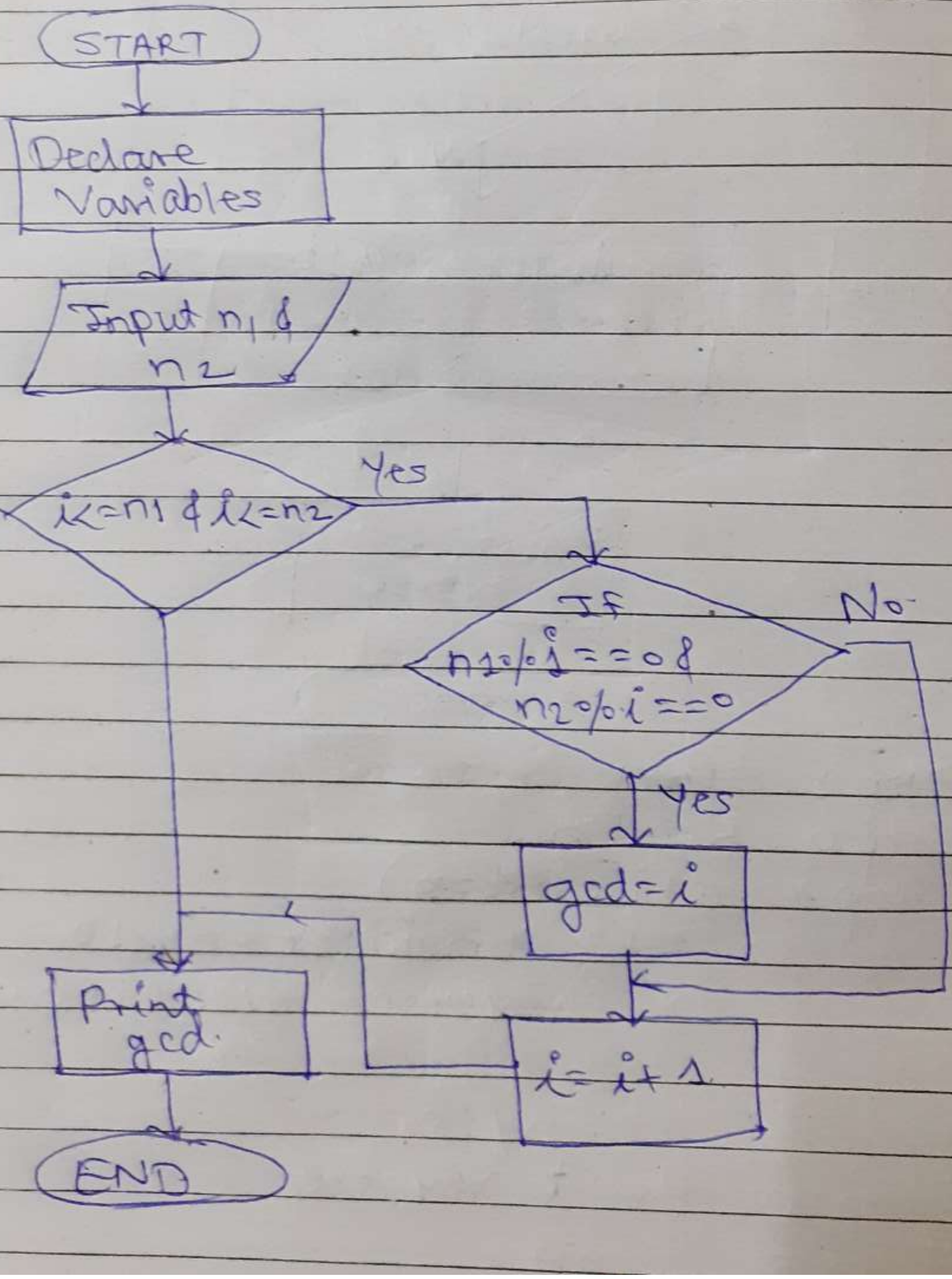
15.) LCM of two Numbers



Step 1: Start
Step 2: Input num1 & num2
Step 3: Find maximum of num1 & num2
Step 4: If max%num1==0 & max%num2 ==0 then LCM=max.
Step 5: ~~Stop~~. Increment max.
Step 6: Stop

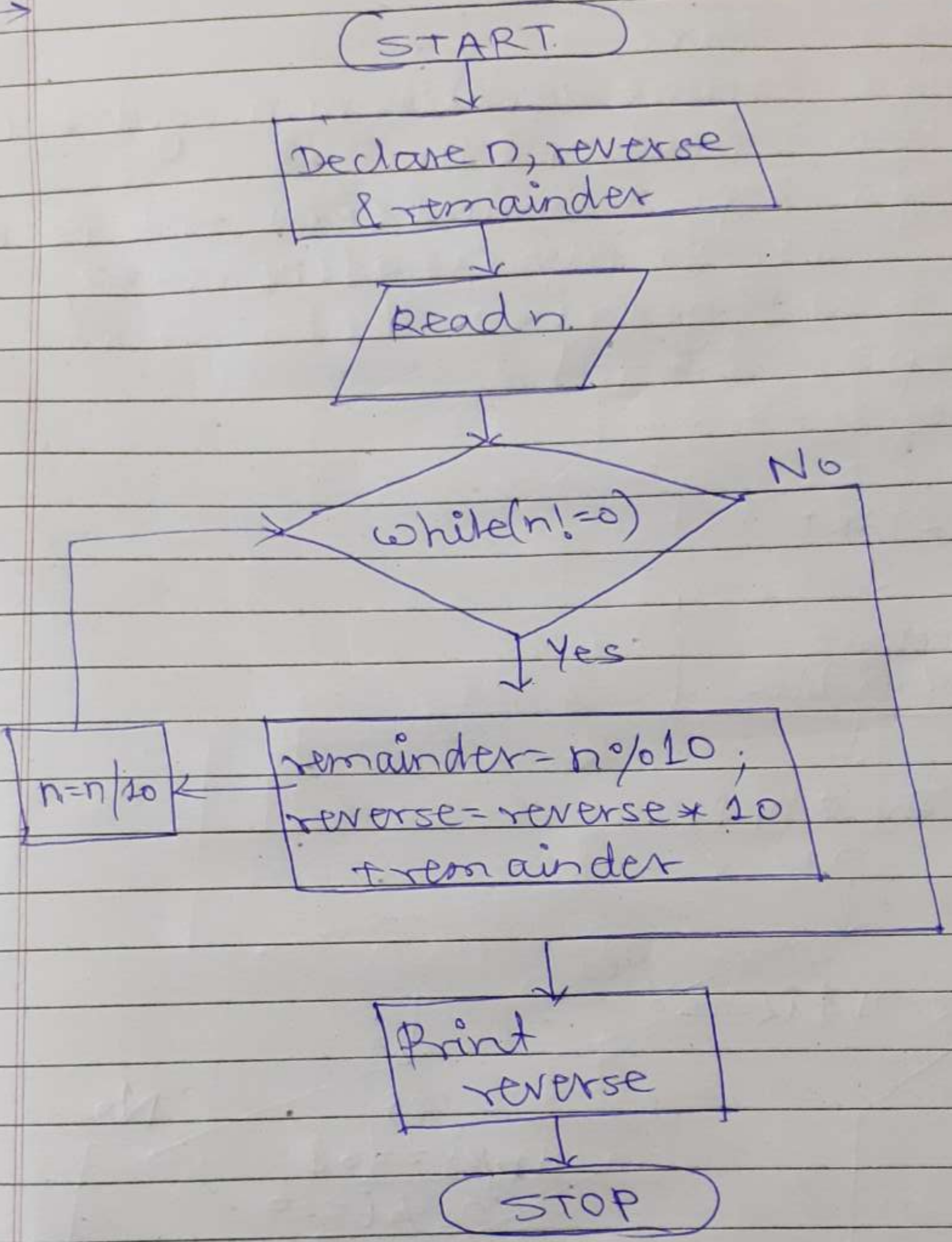
14.) GCD of two given numbers
→

```
Step 1: Start
Step2: Declare variables n1, n2, gcd=1, i=1.
Step3: Input n1 & n2
Step4: Repeat until i<=n1 and i<=n2
        4.1: If n1%i==0 && n2%i==0;
        4.2: gcd=i
Step5: Print gcd
Step 6: Stop.
```



15.)
→

13) Reverse digits of number
→

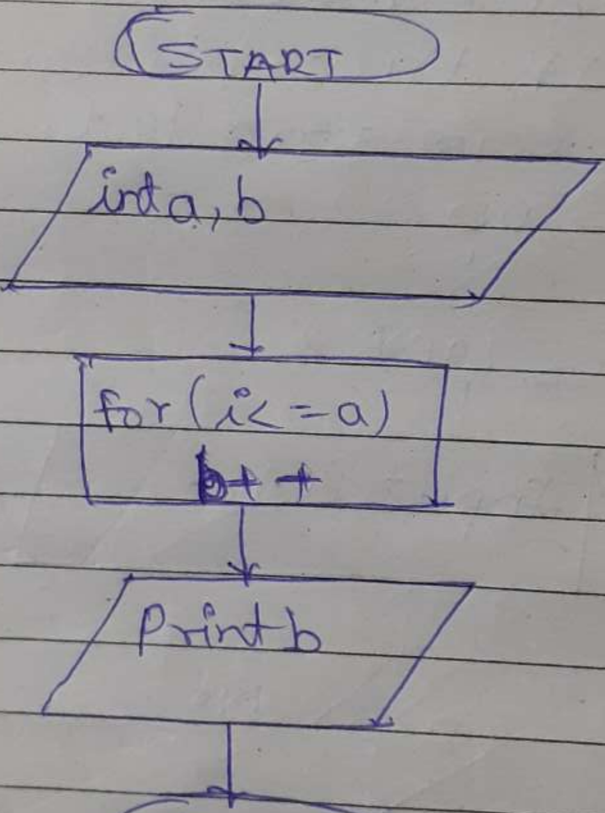


```
↳ Step 1: Start
  Step 2: Read n.
  Step 3: while(n!=0)
          3.1: remainder=n%10
          3.2: reverse=reverse*10+remaind.
                         or.
          3.3: n/10;
  Step 4: Print reverse
  Step 5: Stop
```

12.> Add two numbers without using arithmetic operators.

→

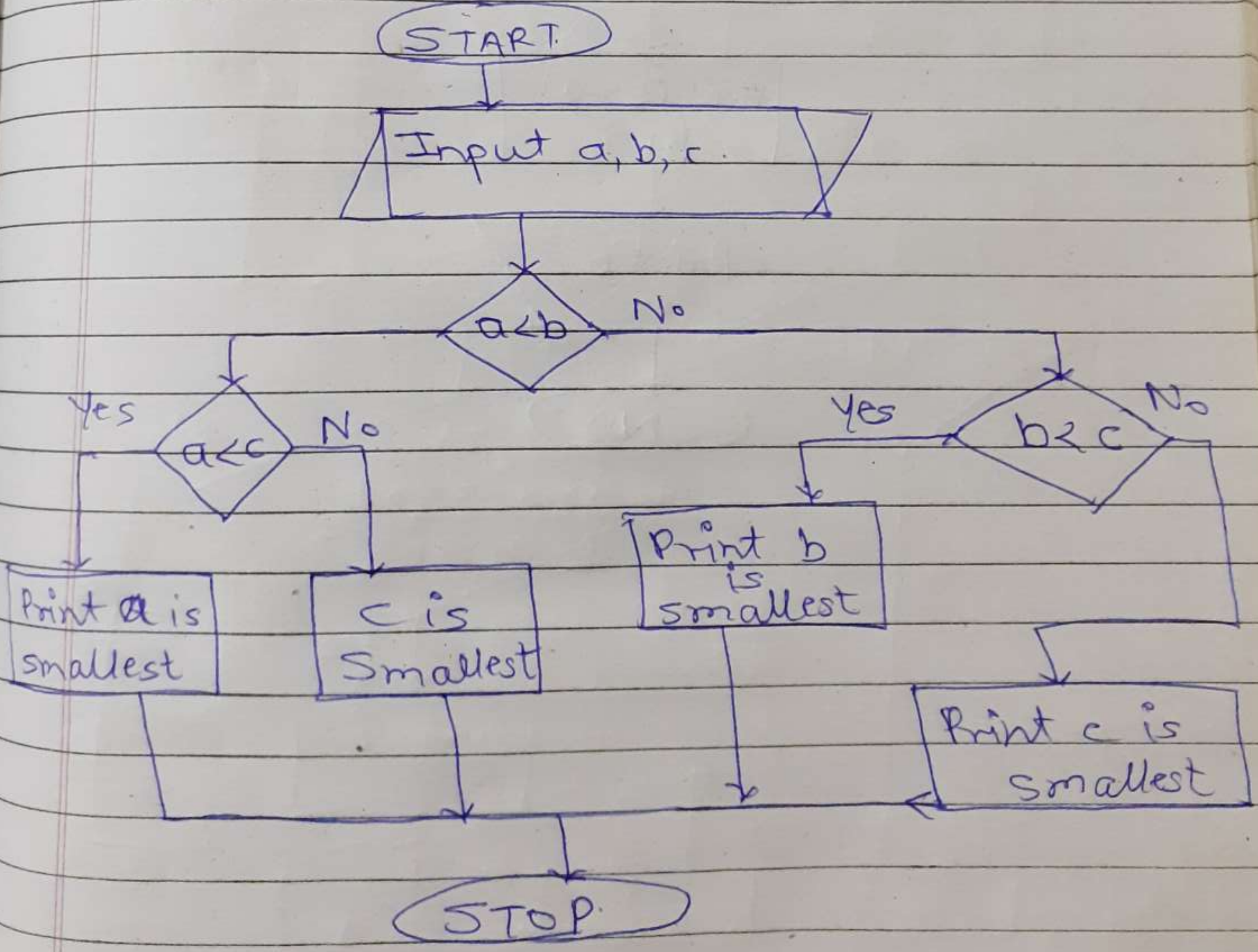
```
Step 1: START
Step 2: Take two i/p
Step 3: for loop (i<=num1)
                num2 ++
Step 4: Print num2
Step 5: End.
```



Page No.
Date

11) Smallest of three numbers:-

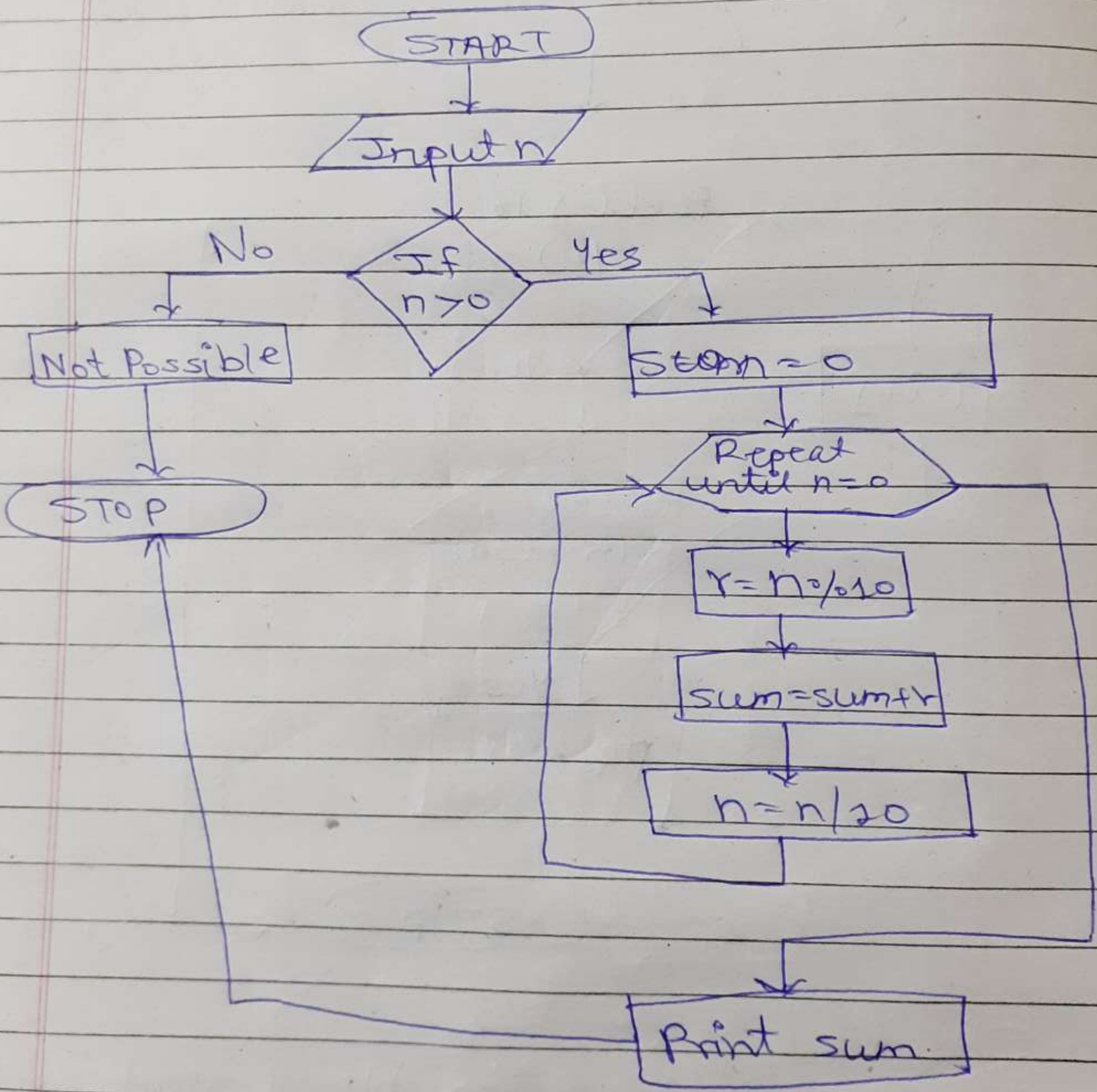
→ Step 1: Start
Step 2: Take three numbers a, b, c.
Step 3: Check if a<b
Step 4: If true go to step 5 else go to step 7.
Step 5: Check if c<a
Step 6: If true c is smallest else a is smallest. Go to step 9.
Step 7: Check b<c
Step 8: If true, b is smallest else c is smallest.
Step 9: Stop



10) Sum of digits of number

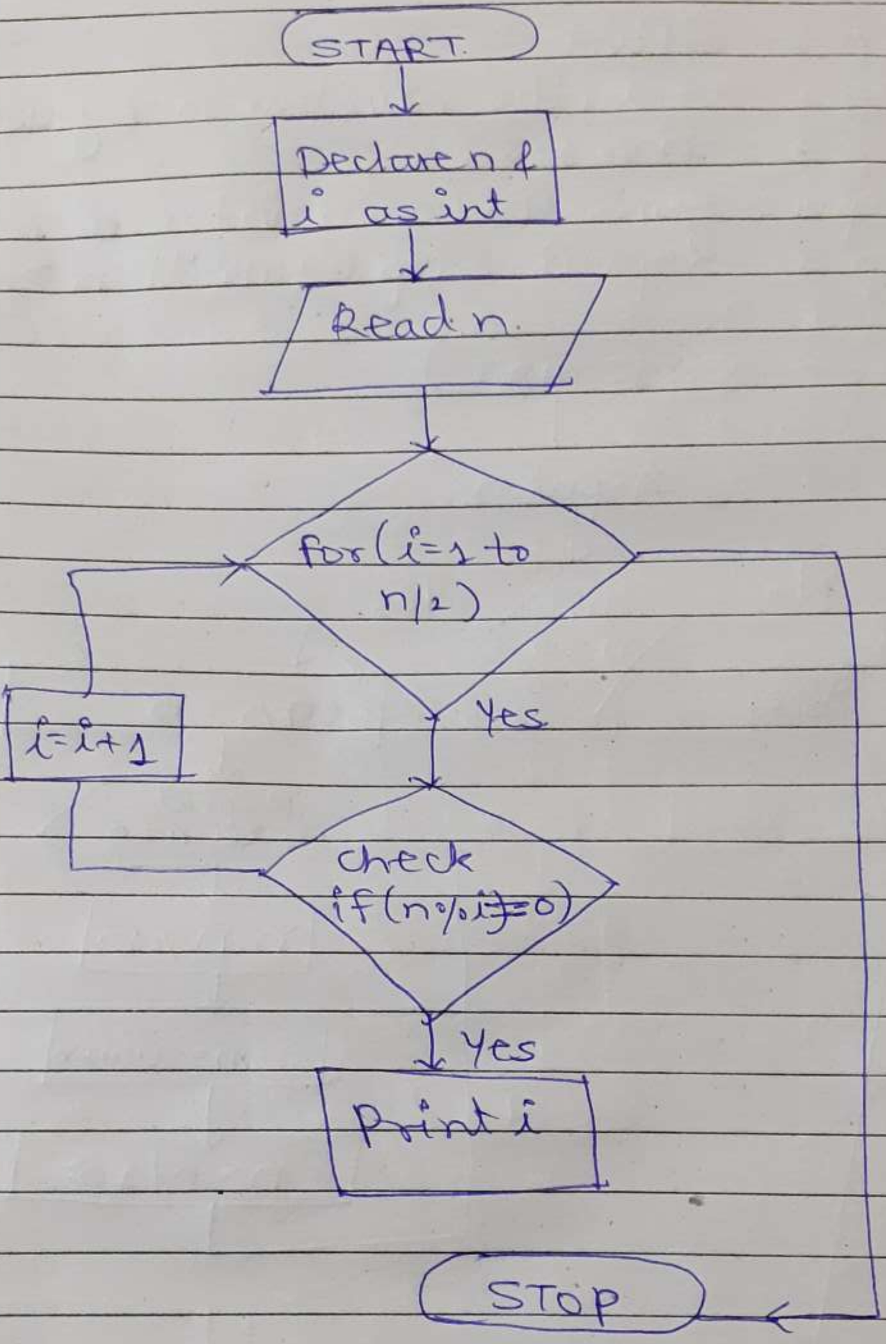
→

Step 1: Start
Step 2: Get the number n by user.
Step 3: n%10
Step 4: Sum the remainder of number
Step 5: Repeat step 2 ~~un~~till n > 0.



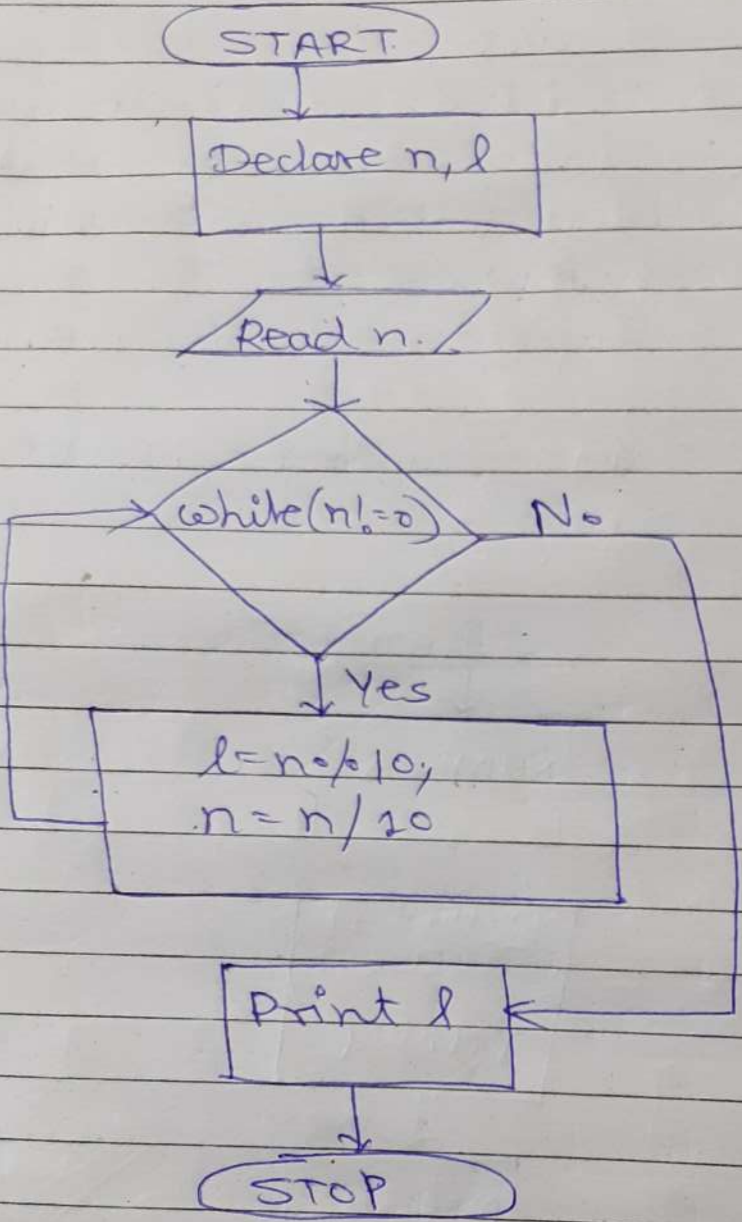
9.) To print factors of number

→



Step 1 : Start
Step 2 : Declare Variable & read input
Step 3 : for i=1 to n/2 in
3.1: Check If (n%1==0)
3.2: Increment i by 1.
Go to step 4
Step 4 : Print i
Step 5 : Stop.

8) Print digits of Number
→



```
Step 1: Start
Step 2: Take input n.
Step 3: Read n
Step 4: while(n!=0)
            Step 4:1: l=n%10
            Step 4:2: n=n/10
Step 5: Print l.
Step 6 : Stop
```

Page No.
Date

7) Print 1 to 10 without loop.

→

Step 1: Start

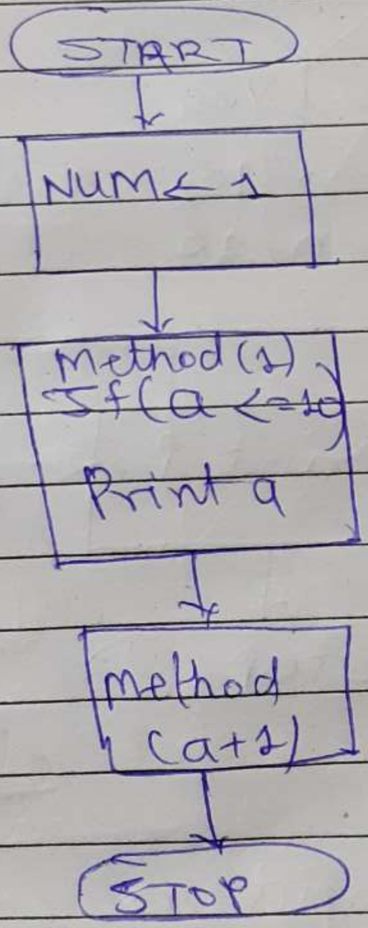
Step 2: Initialize variable number as integer number(n) = 1

Step 3: Read & print the value of number.

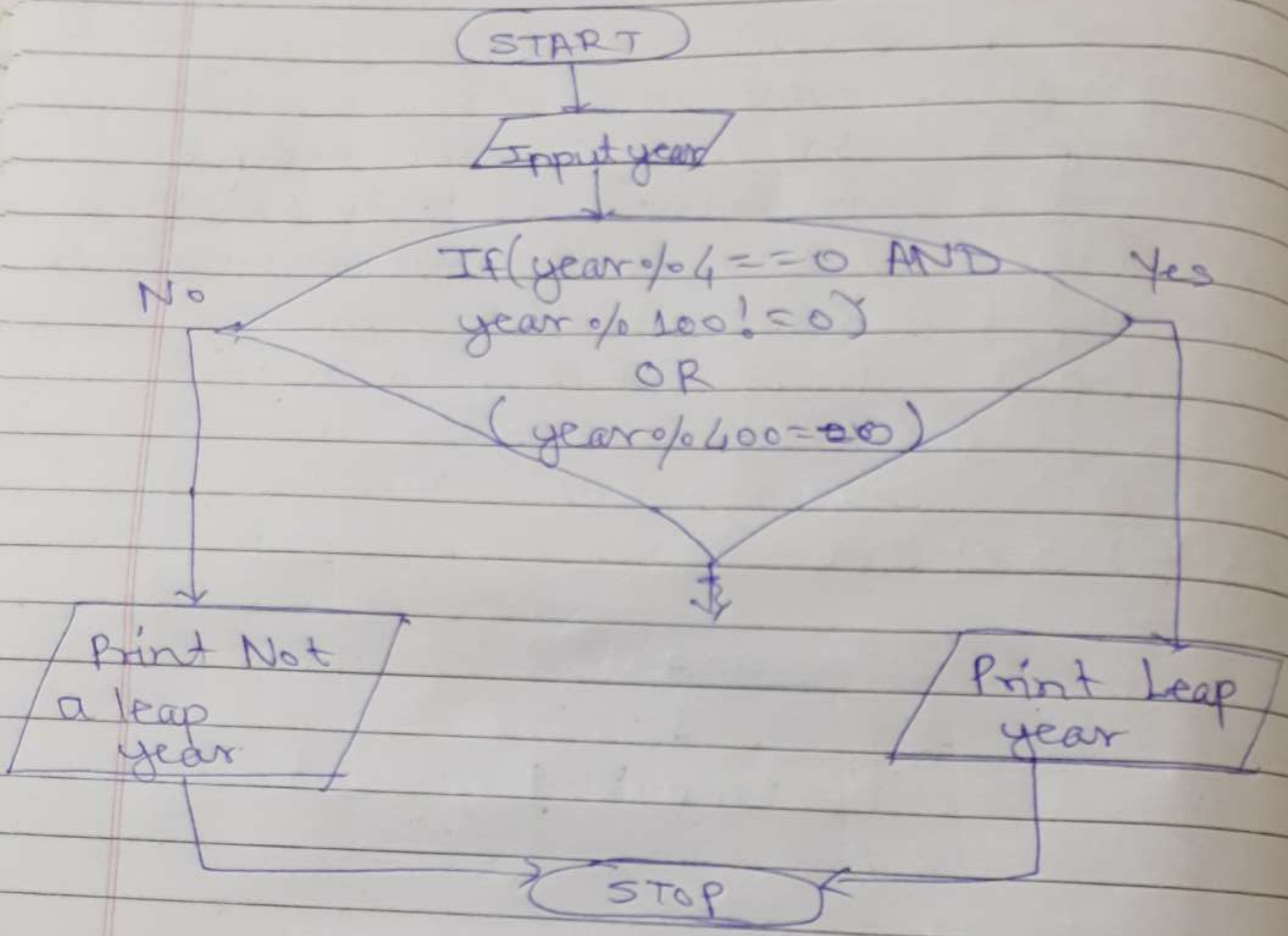
Step 4: Repeat step 3 until number < 10

4.1: number = number(n) + 1

Step 5: Stop



6) Leap year



↳ Algorithm

Step 1 : Start

Step 2 : Input year

Step 3 : If year%4 = 0 & year%100!=0
or year%400 = 0 then go to
step 4
else go to step 5

Step 4 : Print leap year

Step 5 : Print not a leap year.

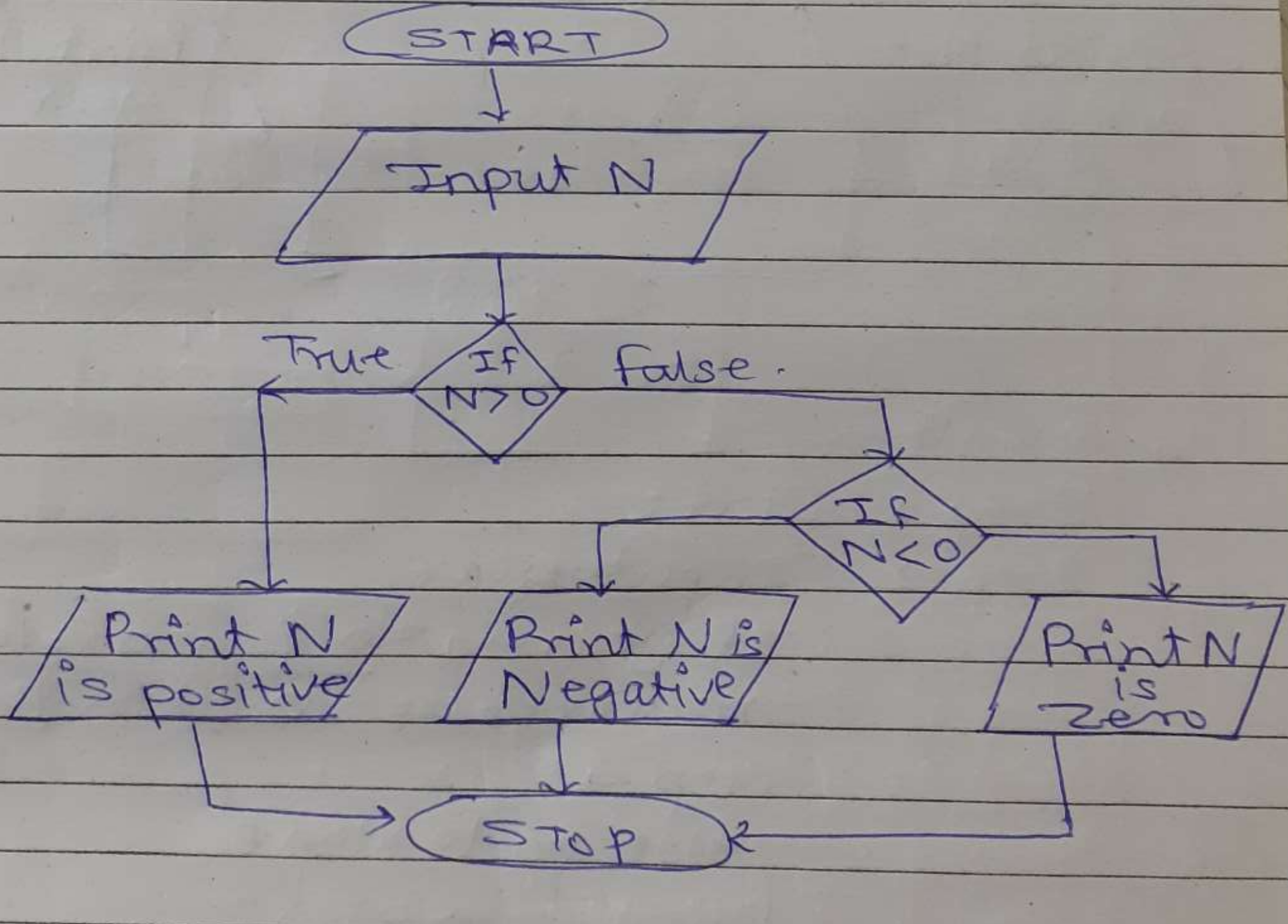
Step 6 : Stop

5.) Positive/Negative number :

→

Step 1 : Start
Step 2 : Enter input n
Step 3 : IF n > 0 to go step 6
Step 4 : Else if n == 0 go to step 7
Step 5 : Else Print Number negative
Step 6 : Print Positive number
Step 7 : Print number is 0.
Step 8 : End.

↳ Flow chart :

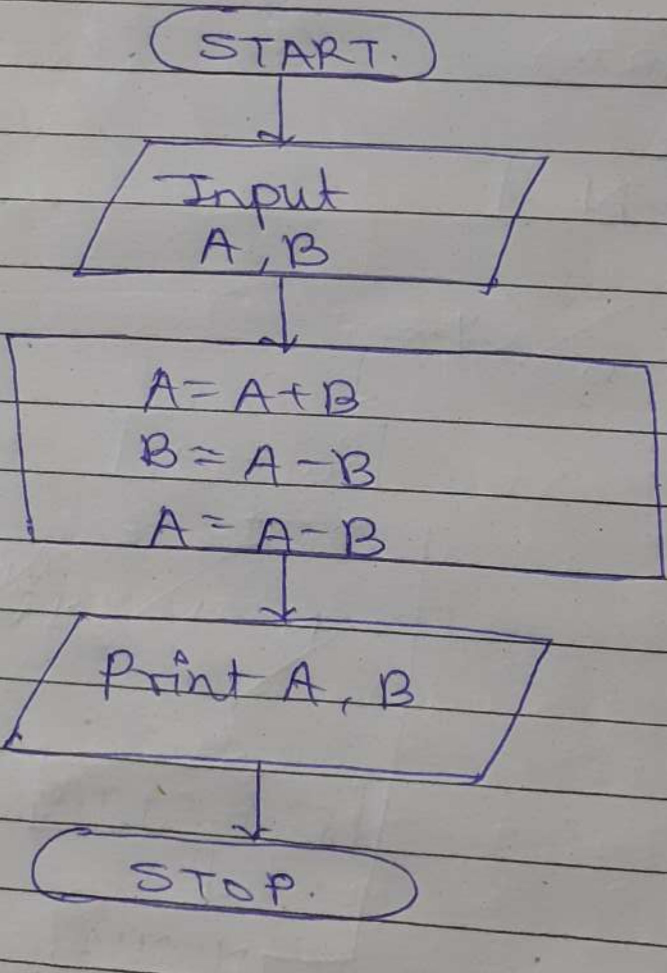


4.) Swap two number without using third variable:

→

Step 1 : Start
Step 2 : Enter A, B
Step 3 : A = A+B
Step 4 : B = A-B
Step 5 : A = A-B
Step 6 : Print A, B
Step 7 : End.

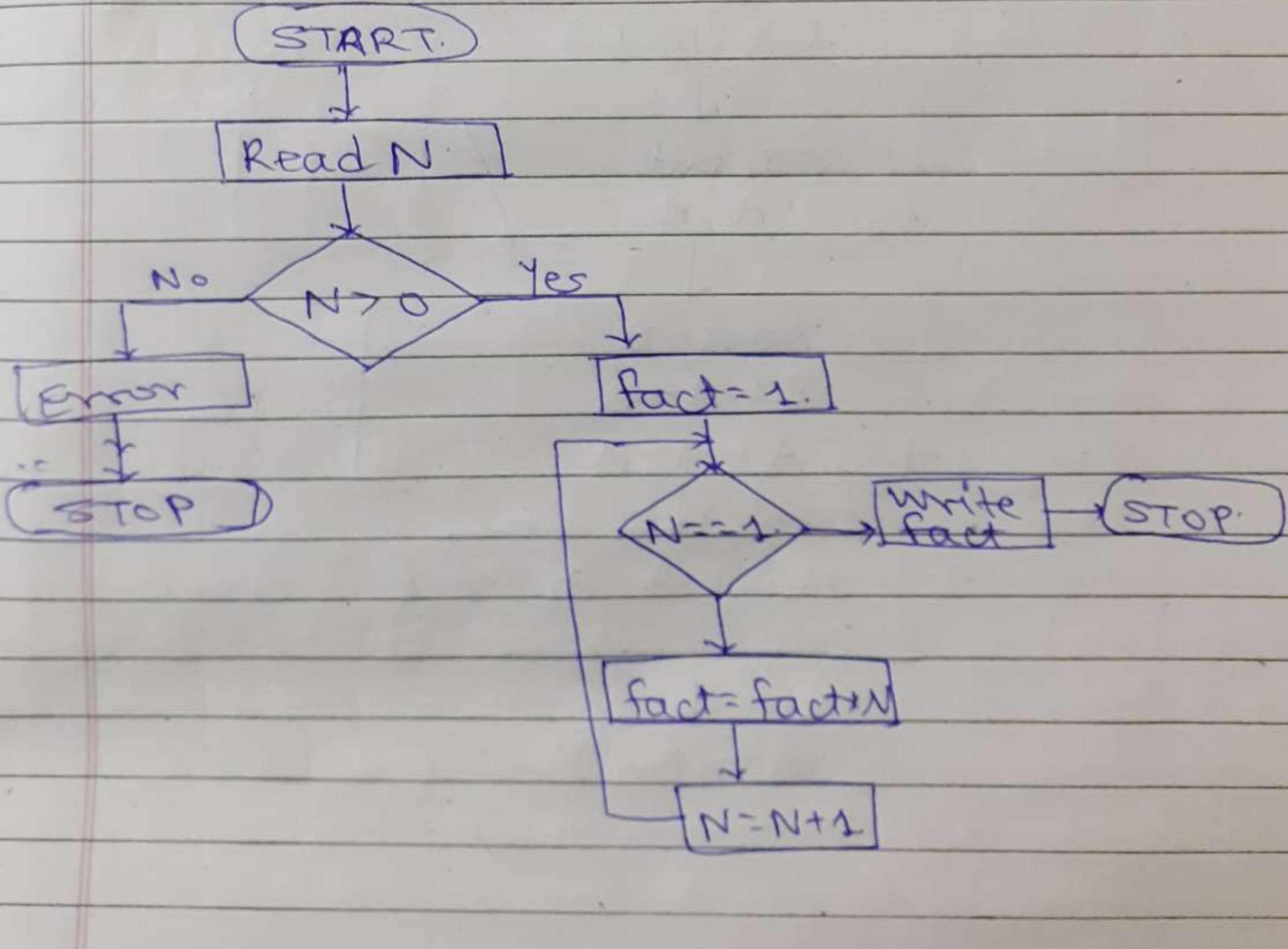
↳ Flowchart:



3) Factorial using recursion.

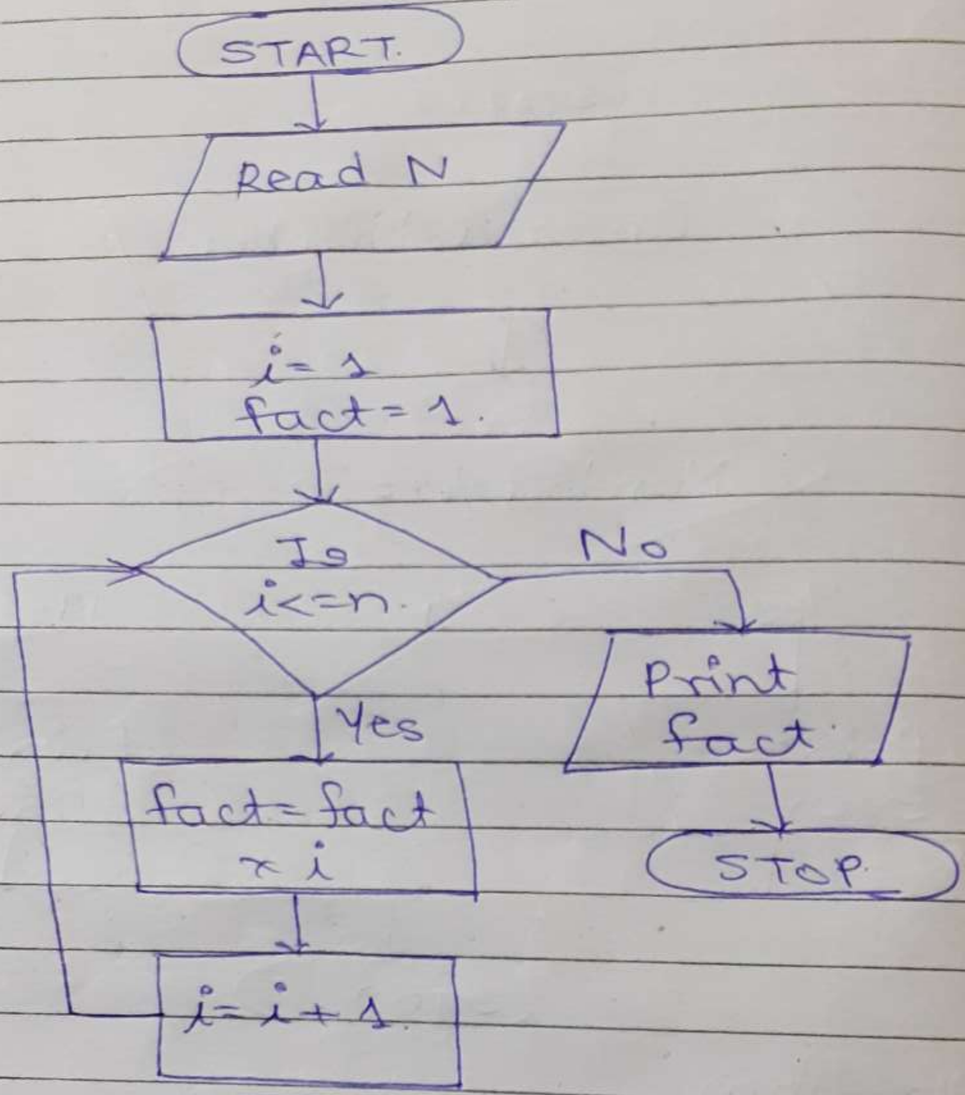
→

```
Step 1 : Start
Step 2 : Read number n.
Step 3 : Call factorial(n)
Step 4 : Print factorial f.
Step 5 : Stop
factorial(n)
Step 1 : If n==1 then return 1
Step 2 : Else
            f=n*factorial(n-1)
Step 3 : Return f.
```



2> Factorial:

→



↳ Algorithm:-

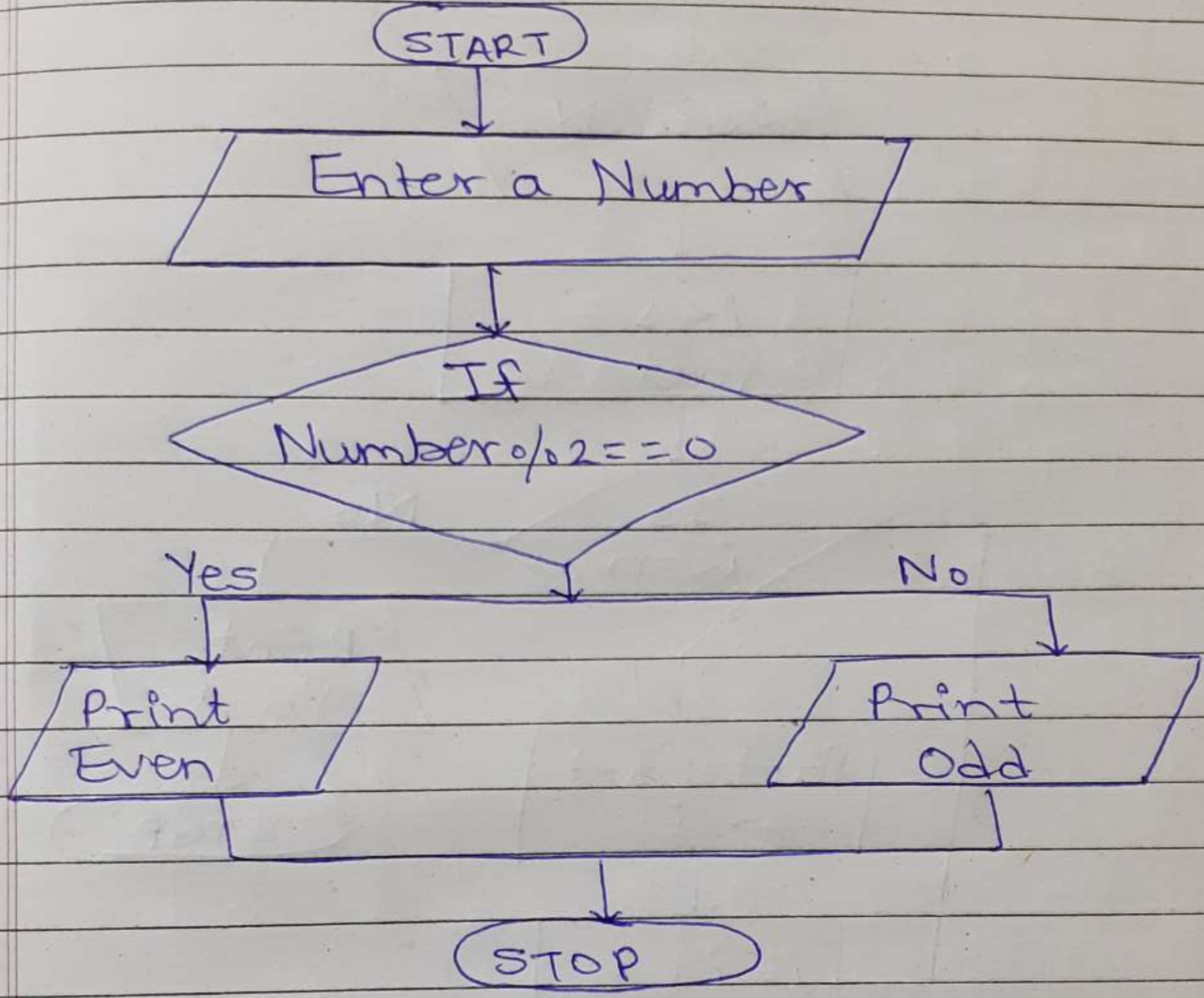
```
Step 1: Start
Step 2: Enter input
Step 3: Read number from user.
Step 4: Initialize variable fact=1 & i=1
Step 5: Repeat until i<=number
~~Step 6~~:  5.1. fact = fact * i
             5.2. i = i+1
Step 6 : Print fact
Step 7 :  Stop
```

Page No.
Date

Assignment - 1.

1) Even or Odd.

→



↳ Algorithm

Step 1 : Start
Step 2 : Read a number to N
Step 3 : Divide number by 2 & store
remainder in R.
Step 4 : If R=0, Go to step 6
Step 5 : Print "N is odd" to go step 7.
Step 6 : Print "N is even"
Step : Stop