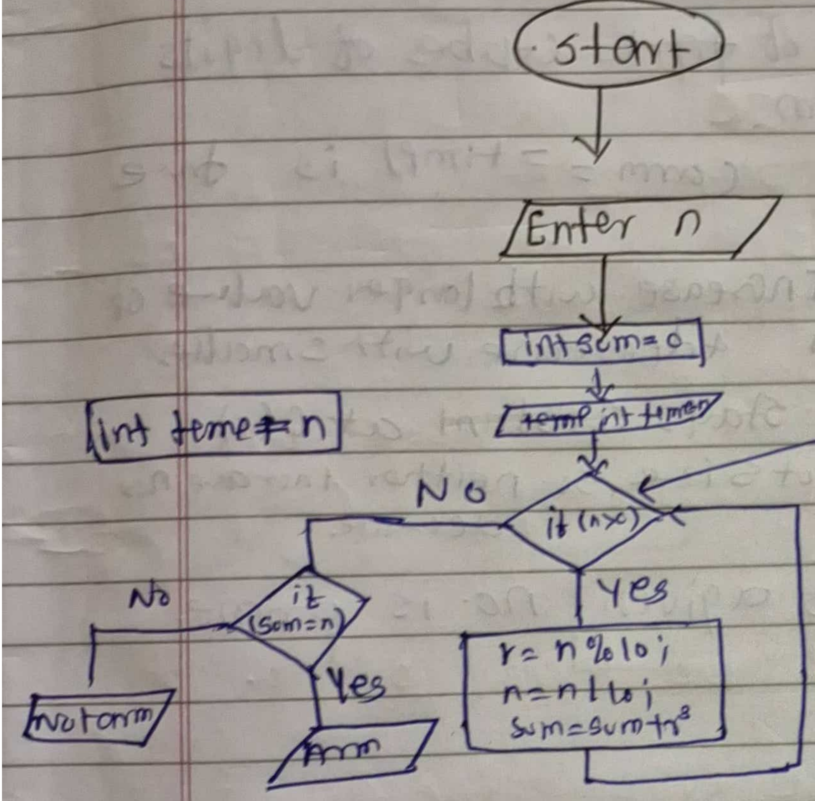
* Assignment - 1 *

1) Write java Program to check if a given no. is armstrong no.

→ ~~progr~~ Flowchart - :



Program -

```
public class ArmNo{
psvm (String[] args) {
   int n,  arm=0, rem;
while(n>0)
{       rem= n%10 ;
       arm = (rem*rem*rem) + arm
       n= n/10;
}
if (c==arm)
       S.o.pln( "Arm Strong No.");
else
       S.o.pln( "Not Arm No.");
   }
 }
```

Explanation - 3 digit no. is eaqual to sum of Cubes of its digits. 153

$1^3+5^3+3^3 = 1+125+27 = 153$

in Loop while runs as long as n is greater than 0. %10 extract the last digit of n & store it in rem.

Arm contains sum of ~~digits~~ cubes of digits & add result to arm.

Program checks if (arm = =tmp) is true otherwise false.

Time Complexity - Increase with larger value of n & decrease with smaller values.

Space Complexity - stays Constant at O(1) regardless of input size, so neither increase nor decrease.

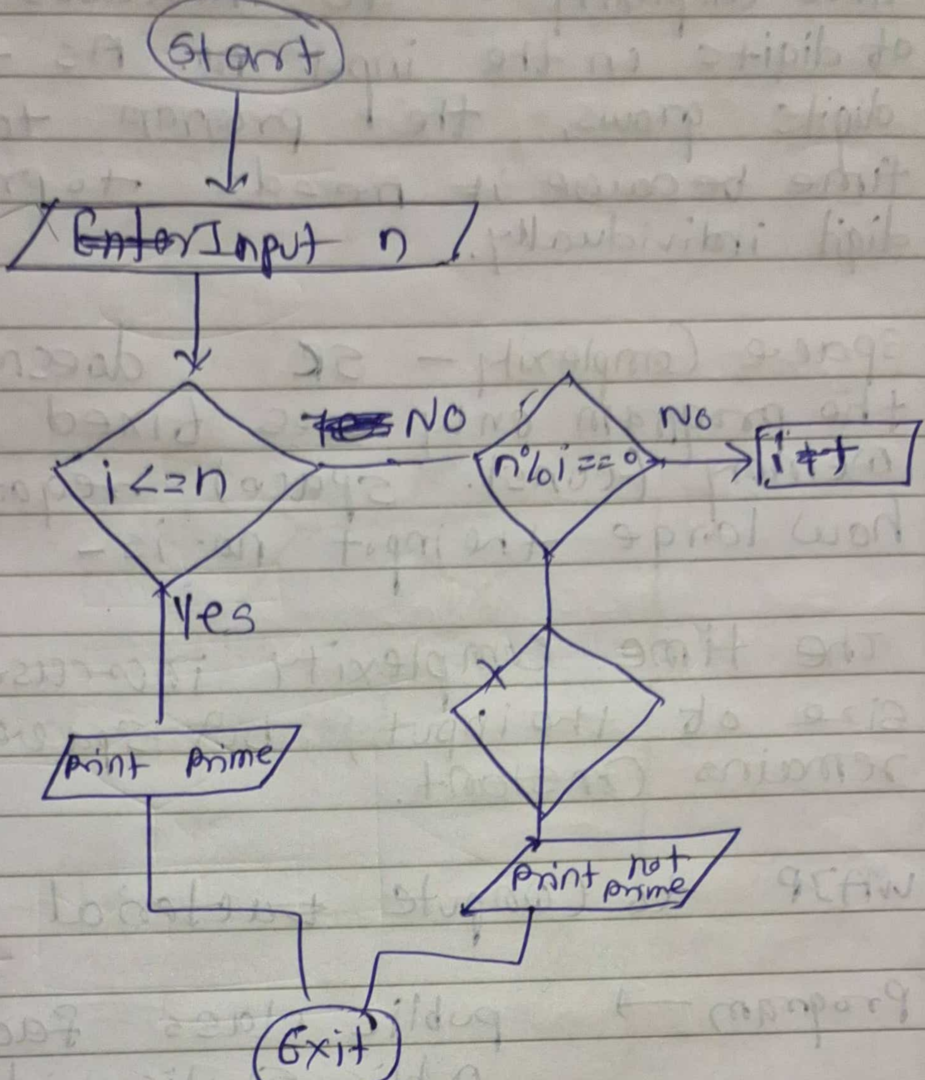
Q.2) Prime number

WAP to check if a given no. is prime.

Program —

```
public class ArmNo. {

  public static void main<

          int  n = (153);

          for (i = 1 ; i < n ; i++) {

                if (n%2 == 0) {

                    system.out.println("No. is Prime

                }

          }

          if (i = n) {
          System.out.println("prime");
```

Flow chart



* Explanation — prime no. is is divide by only itselb or one(1). In this program I took one variable n 4 strore value 153 I use for loop 4 check first Condition, if no. modute by 2 it gives non prime no. after loop Completion if no. equel to given no. print prime.

i

Output 153 is prime no.

Time Complexity - TC increases with the no. of digits in the input no. As the no. of digits grows, the program takes more time because it needs to process each digit individually.

Space Complexity - SC doesn't increase the program only uses fixed amount of memory (const. space) regardless of how large the input no. is -

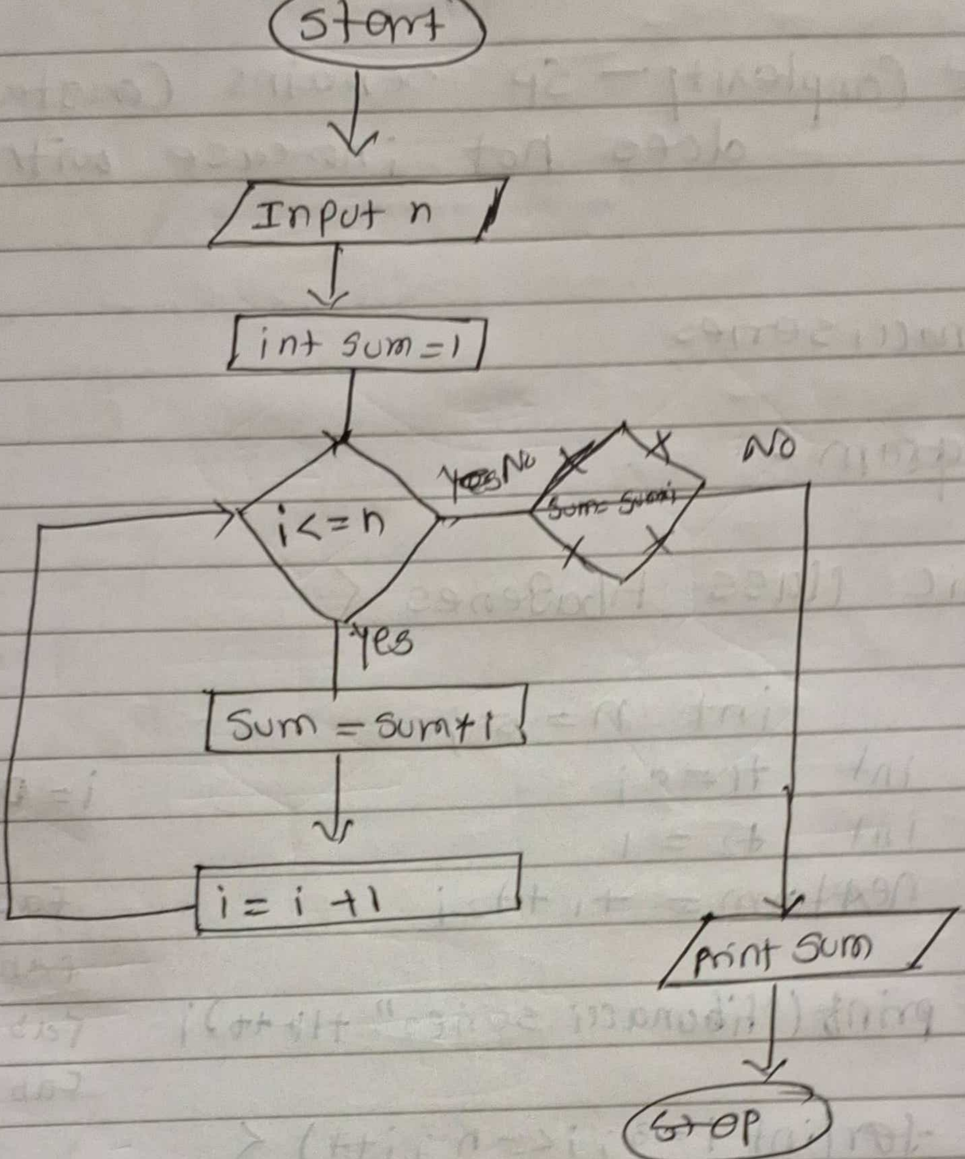
The time complexity increases with size of the input, but space complexity remains Constant.

(Q.3) WAJP to Compute fuctorial of no.

Program →

```
public class fac {
 public Static void main(String args[]
 {
    int n = 5 ;
    int sum = 1 ;
    for(i=1 ; i< ;i++){

       sum = i x sum ;

    }
    System.out.println("Fac is" + sum

 }
}
```

Flow chart



Explanation

factorial means product ~~of~~ all 1 to n is called factorial. I used for loop to multiply one by one all no. from 1 to n 4 store all sum in one variable after completing the loop print the variable which store all multiplacation.

Time Complexity - TC increase because loop runs more times.

Space Complexity - SP remains Constant & does not increase with larger inputs

Q.4) Fibonacci series

→ program -

```
public class FiboSeries {

            int  n = 5;
      int   t1 = 0;
      int   t2 = 1
      nexterm. = t1 + t2;

      print ("Fibonacci series" +t1+ +t2);

      for(int i = 3; i<= n; i++) {

            nextterm = t1+t2;

            t1 = t2;
            t2 = nexterm;
         }

 }
```

i = 0, fab = 0

i + fab

fab = 0+0 = 0

fab = 1+0 = 1

fab = 2+1 = 3

fab = 3+ ~~[illegible]~~ 6

5

0

1

t1 t2

next = 0+1 = 1

~~t~~ t1 = t2

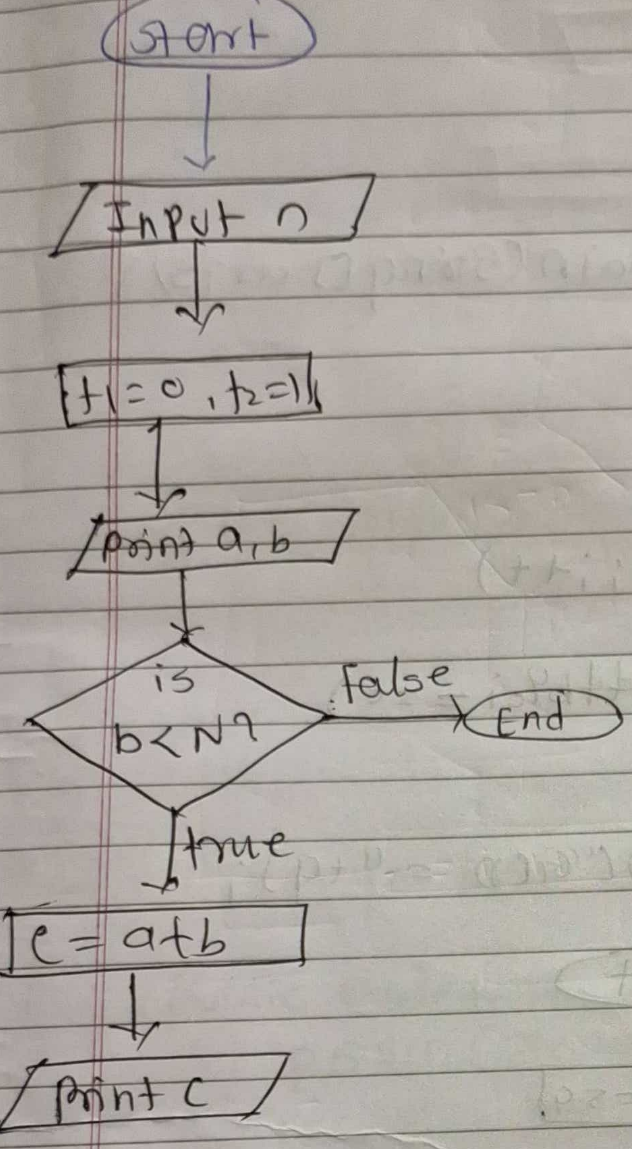
t1 = 1

t2 = 1

2^{n1}

~~next~~ = 2

Flow chart -



Time Complexity -

O(n)

space Complexity -

O(n)

Explaination

In this program, I declare sum variable & use for loop, in fibanacin Serier 0 & 1 t1 & t2 I declare ~~or~~ first then use for loop in this when i is eaul to 3 then add addition on ~~2~~ starting term to variable & assign value of ~~o~~ t2 to t1 & assing value of nextterm to t2 repeat is until condn false.

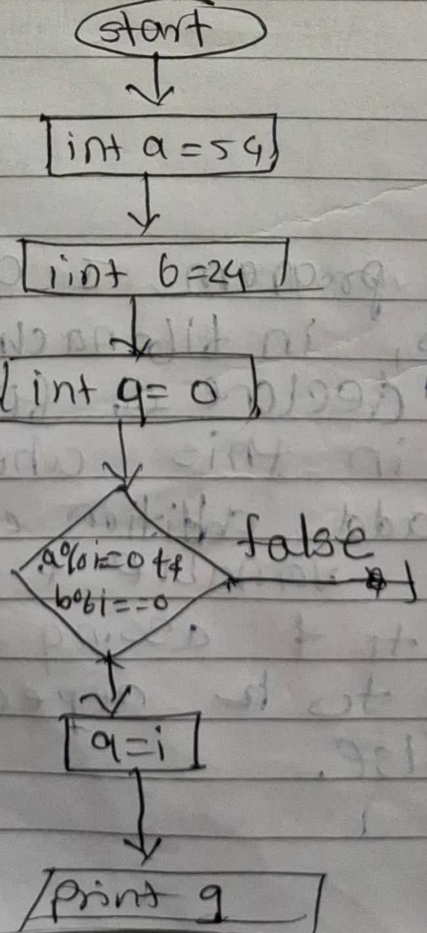
```
if(a%i=0t a [illegible] %0 b
    {
    }
```

Q.5] Find GCD →

WAJP to find the Greatest Common Divisor (GCD) of 2 nos.

```
public Class Gcd {

public static void main(String[] args){
        int a=54
        int b=24
        int g=0;
      for(int i=1 ; i<a ; i++)
        {
            if (a%i==0 &&b%i==0)
                g=i ;
        }
   }    System.out.println("GCD ="+g);
}
```

Flow chart



output→

Explanation- In for loop i willinsement upto the value of a, thereis on Condition ~~a~~ if value of a modulus to value of ~~B~~i is qual to zore then it will assina value of i to q & print acd no.

Time Complexity o(n)

space Complexity →o(1)

Q.5) WAJP to find the square root of a given no.

```
public ~~Code with~~ Square {
      psvm(String args[]) {
            int ~~a~~ num;
            ~~num = sc.nextInt~~
            int sqrt =0;
         for (int i=1; i<num; i++) {

            if (num%i ==0) {

                  if(i*i= =num) {
                  sqrt =i;
                  }
            }
      System.out.println("Square root is "+sqrt);
} } }
```

Q.9) Integer Palindrome -

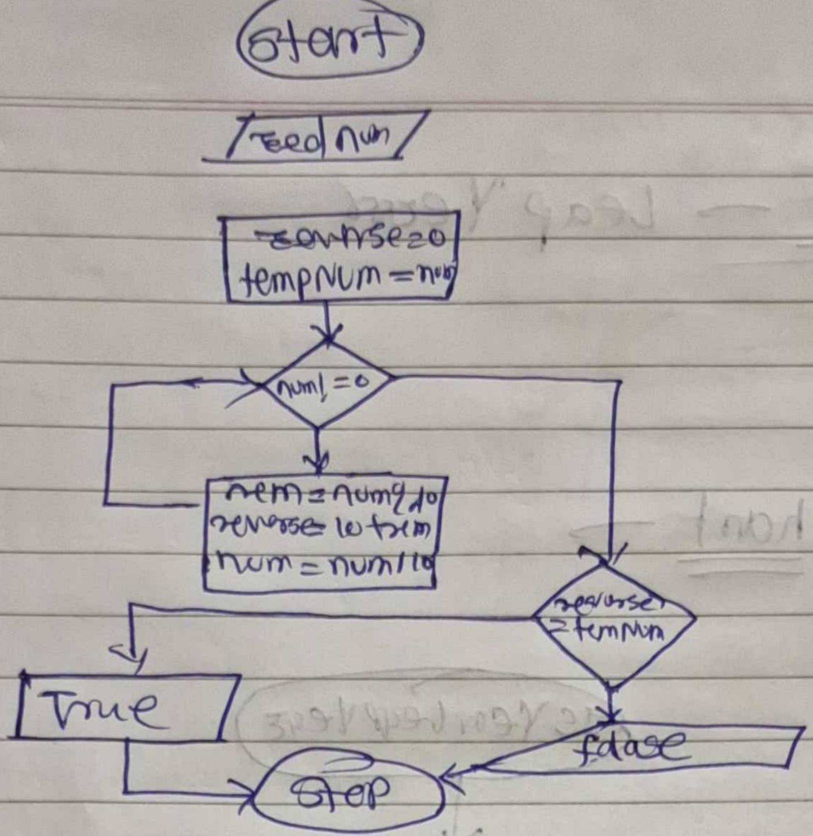
Program-

```
public class palin {

  psvm(String args[]) {
        int n=121;
        int s=0;
intc ;  ~~n = reb. nex~~
intr;     c=n;
      while(n>0)
      {
          r=n%10;
       s=(s* 10)+r;
       n=n/10;
      }
      if(c==s)
       s.o.pln(" Palindrom no.);
      else
       s.o.pln("Not palindrome no.");
    }
  }
}
```

Output → True

Radar

11qqeq



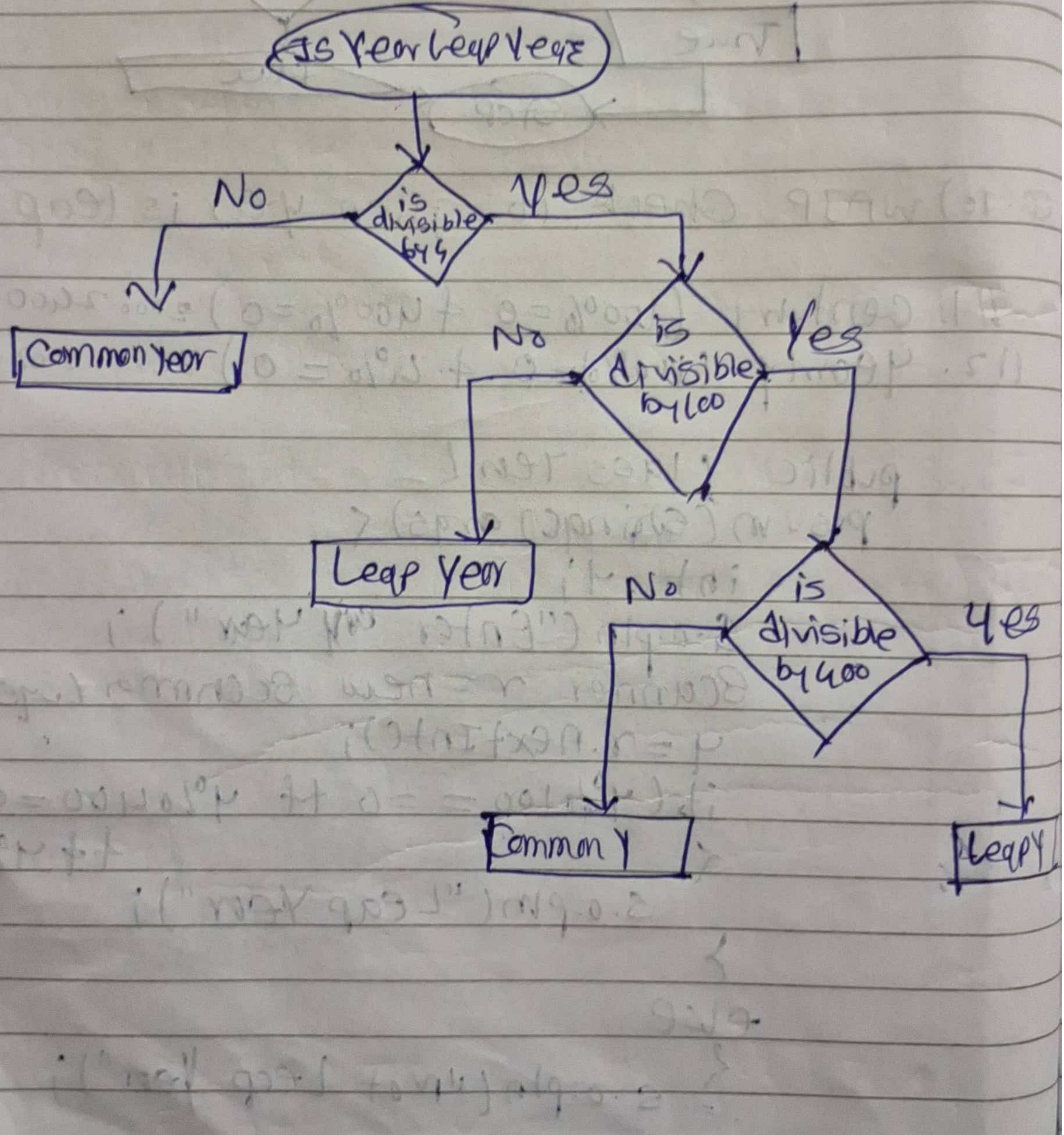
Q.10) WAJP Check ib. given year is leap yer.

#1. century (100% = 0 & 400% = 0) 2000 2400
#2. yearly (100%!= 0 & 4% = 0) 2020 2024

```
public class Yeer {
 p.s v m (String[] args) {
      int y;
      s.o.pln("Enter any Year");
      Scanner r=new Scanner(system.in);
      y=r.nextInt();
      if(y%100==0 && y%400==0 || y%100 != 0
                              && y%4==0)
      {
         s.o.pln("Leap Year");
      }
     else
      {  s.o.pln("Not Leep Year");
      }
 }
}
```

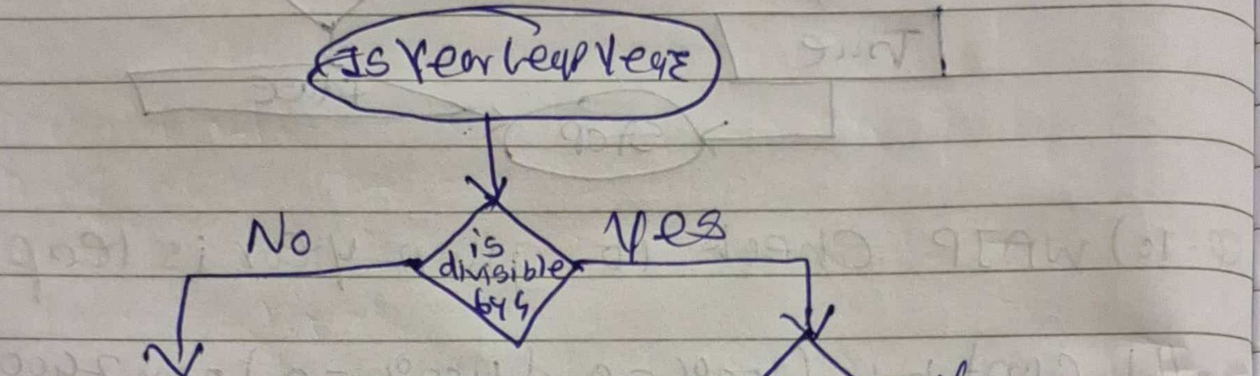
Output → Leap Year

Flow chart →



Output → Leap Year

Flow chart →


Is Year Leap Year
is divisible by 4
No
Yes
Common Year
is divisible by 100
No
Yes
Leap Year
is divisible by 400
No
Yes
Common Y
Leap Y