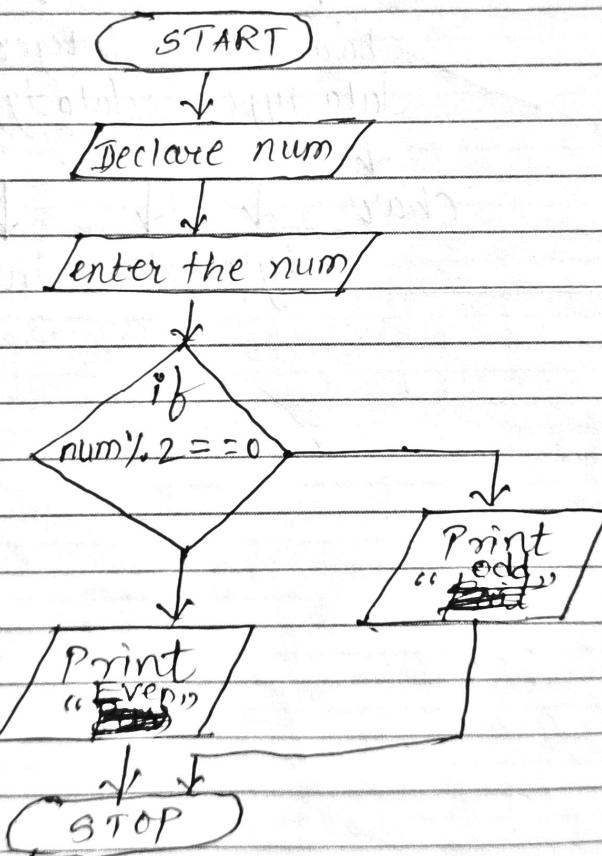


Date :



- 1) check the given number is Even or ODD.  
→ algorithm:-
- 1) Start
  - 2) Enter the number. initialize declare var, num
  - 3) Enter the number.
  - 4) if ( $\text{num} \% 2 == 0$ )
  - 5) if condn true print Even
  - 6) If condn false print odd.
  - 7) Stop.

flow chart:-



Date :



2) Factorial of given number.

→ algorithm :- 1) Start

2) Enter the number

3) Initialize  $i=1$ , fact = 1

4) if ( $i \leq num$ )

fact = fact  $\times$  i

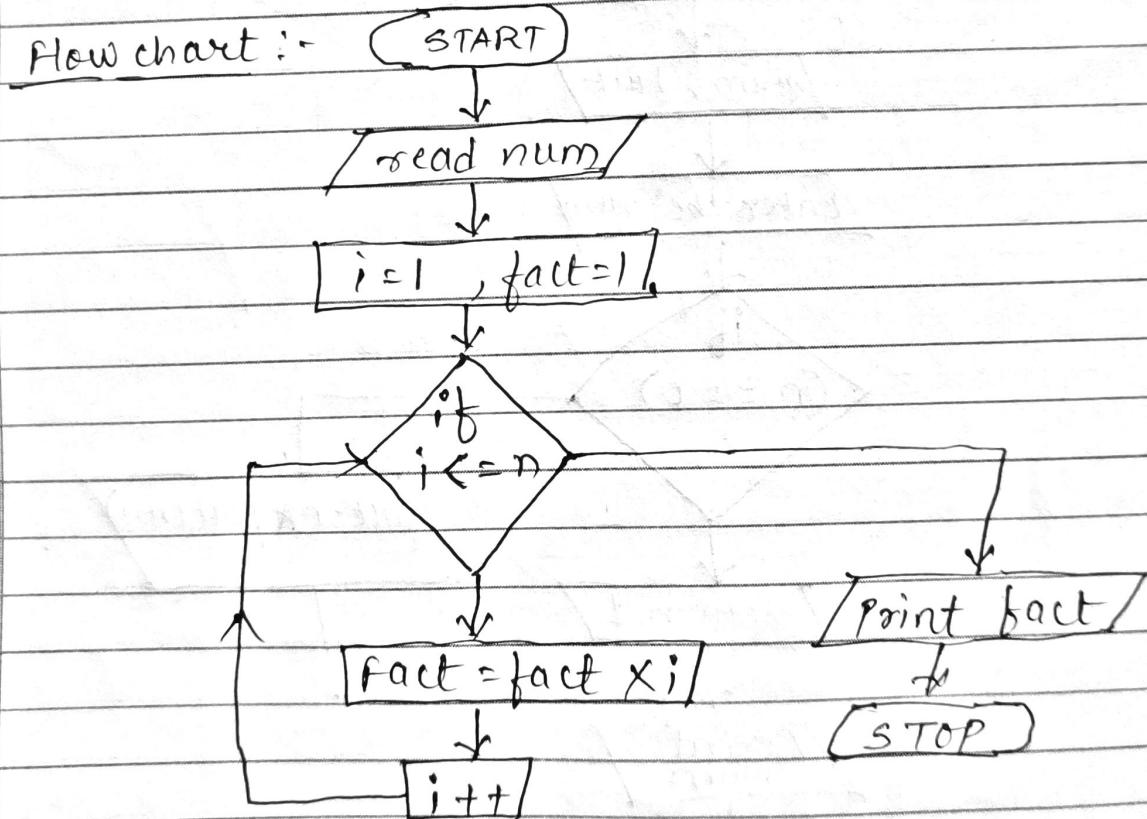
$i = i + 1$

5) Repeat step no. 4 till  $i \leq num$

6) print fact

7) stop.

Flowchart :-





Date :

\* Factorial of number using recursion.

→ Algorithm :-

1) declare num, fact.

2) Enter the num.

3) if ( $n == 0$ )

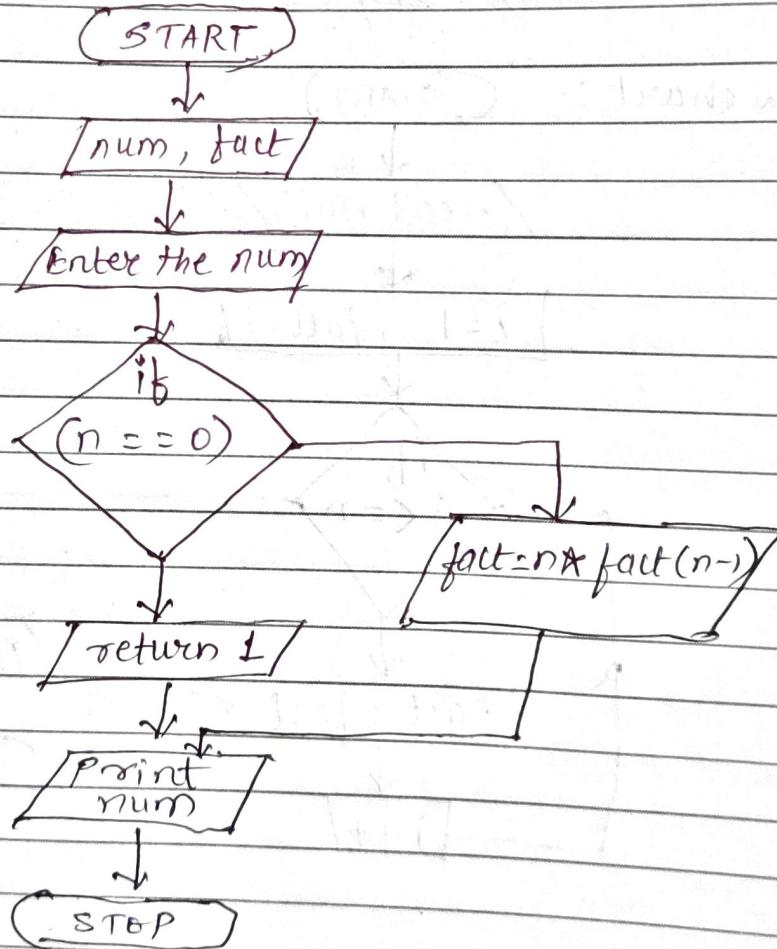
4) If cond'n true return 1

5) If cond'n false fact =  $n * \text{fact}(n-1)$

6) print the num.

7) Stop.

flowchart :-



Date :

4) swap two numbers without using third variable approach.

→ Algorithm :-

1) start

2) Initialize num1=20 & num2=10

3)  $x = x + y$     num1 = num1 + num2

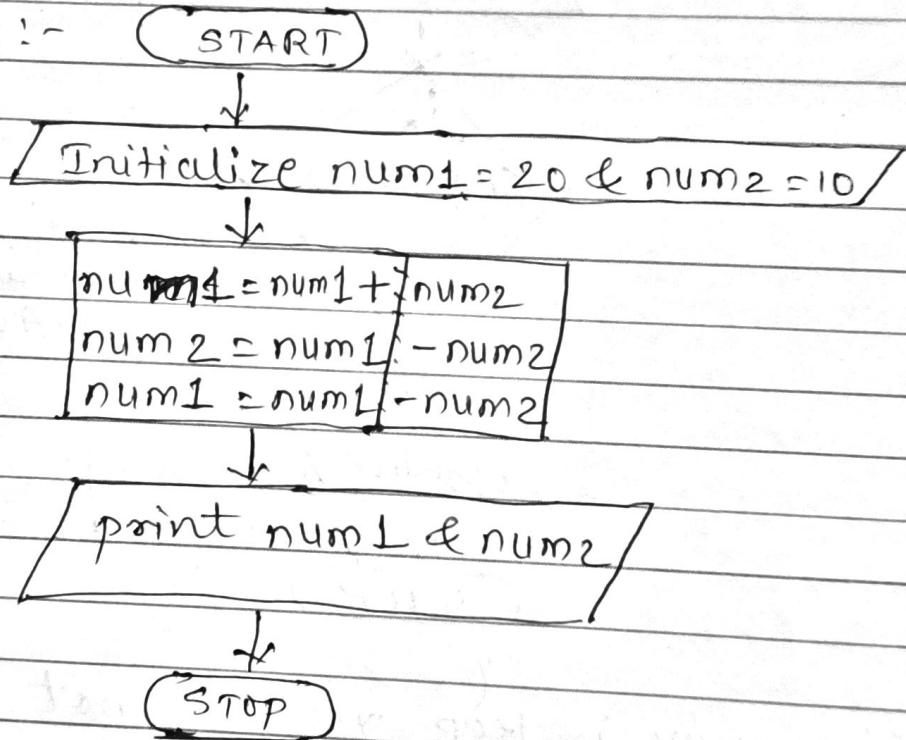
$y = x - y$     num2 = num1 - num2

$x = x - y$     num1 = num1 - num2

4) Print num1 & num2 after swap.

5) stop.

Flowchart :-





5) check the number is +ve or -ve

→ Algorithm :- → start

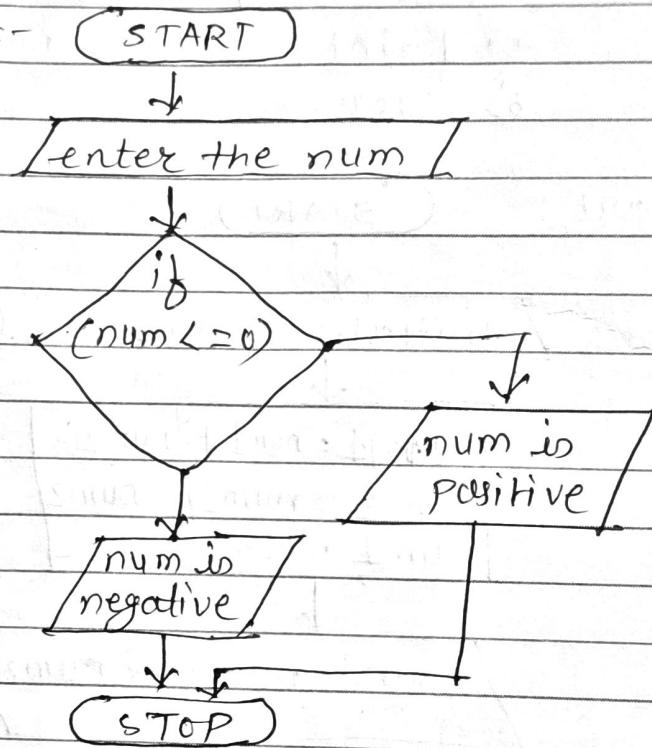
2) enter the number.

3) If ( $\text{num} \leq 0$ )

4) If condn true the num is negative.

5) If condn false the num is positive.

Flowchart :-



6) Given year is leap year or not

→ Algorithm :- → START

2) Enter any year.

3) check given year is divisible by 400 or not

4) If divisible then leap year else check for further condition.

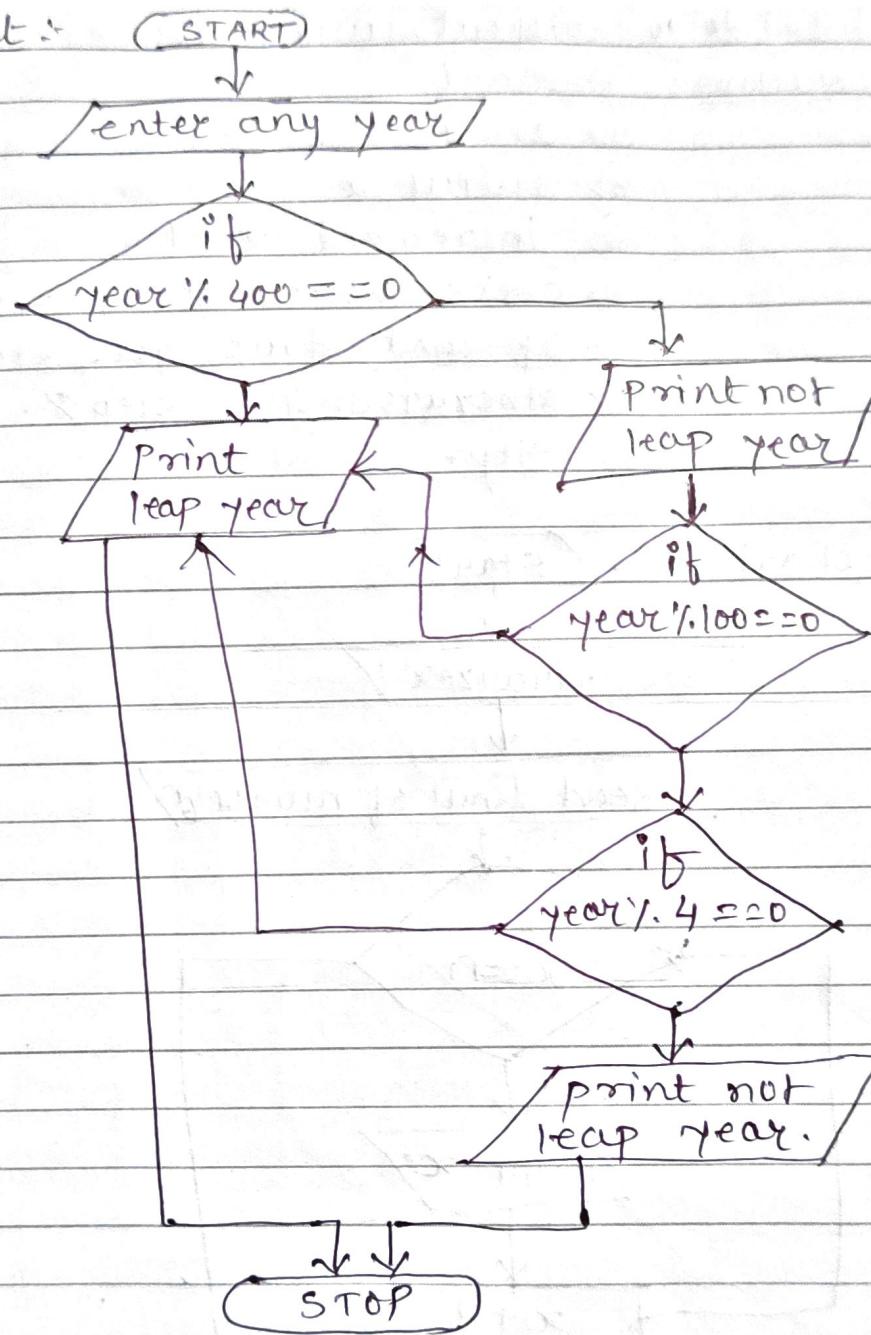
5) Now if is divisible by 100 then it is not leap year or else we further divide it by 4.

6) If it is divisible then it is leap year else its not.

7) STOP.



Flowchart :-

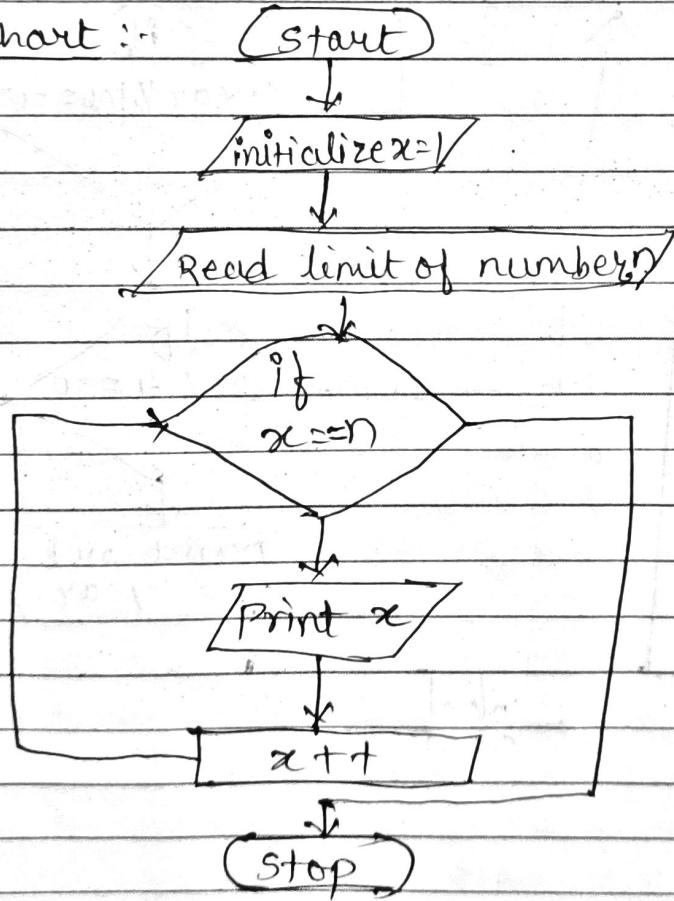




7) Print 1 to 10 without using loop.

- Algorithm :-
- 1) Start
  - 2) Initialize  $x=1$
  - 3) output  $x$
  - 4) increment  $x+1$
  - 5) check the value of  $x \leq 10$
  - 6) If cond'n true goto step 3
  - 7) otherwise goto step 8.
  - 8) Stop.

Flowchart :-



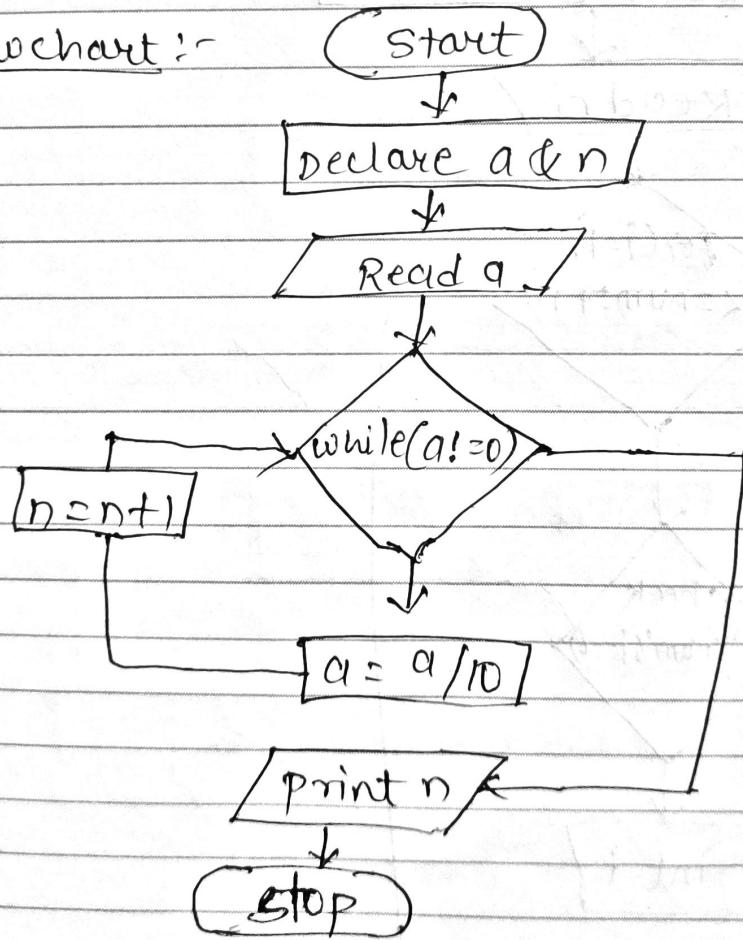


→ print digits of given number.

Algorithm :-

- 1> Start
- 2> Declare a & n
- 3> Read a
- 4> while( $a \neq 0$ )
- 5> If cond'n true print n.
- 6> If cond'n fail then  $a = a/10$
- 7>  $n = n + 1$
- 8> Stop.

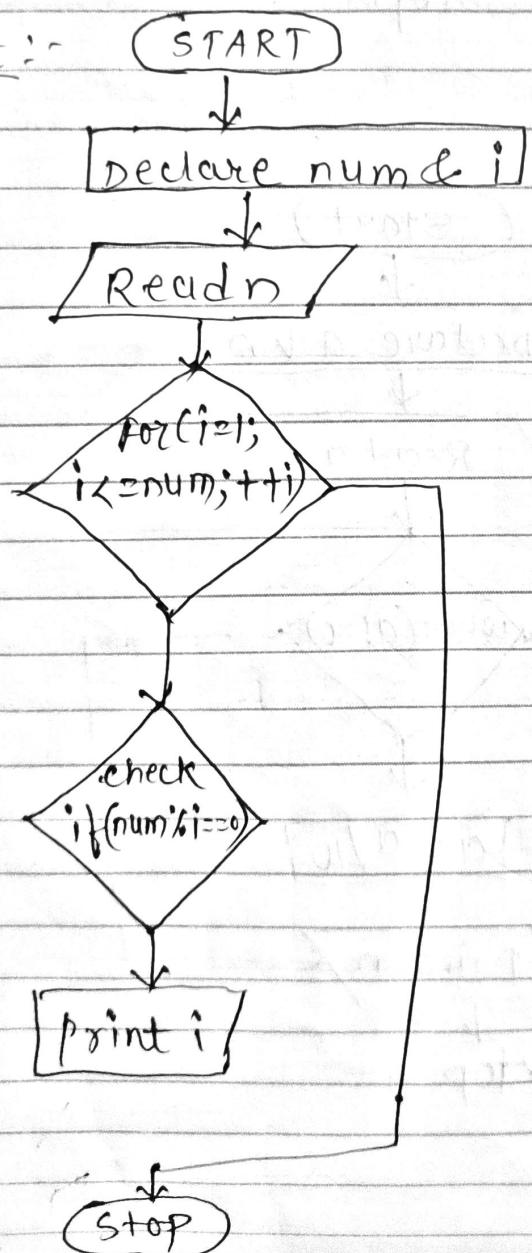
Flowchart :-





- \$ print all the factors of given number.
- Algorithm :-
- 1) Start
  - 2) Declare num, i
  - 3) Enter the num
  - 4) for ( $i = 1; i \leq num; ++i$ )
  - 5) if ( $num \% i == 0$ )
  - 6) print the factor of given number.
  - 7) Stop.

Flowchart :-



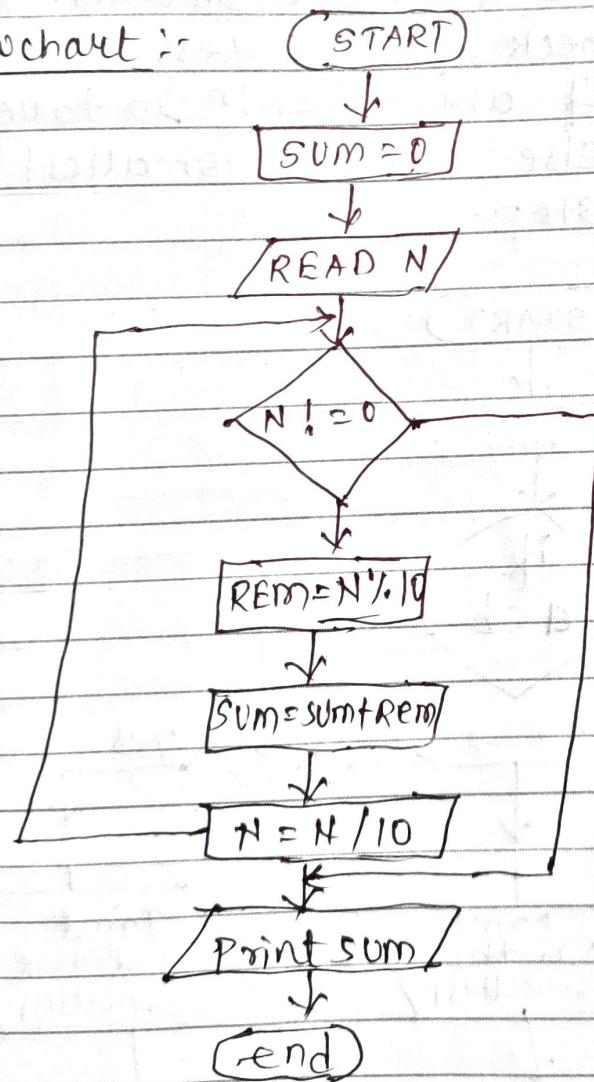
STOP



→ sum of the digits of given number.

- Algorithm :-
- 1> Start
  - 2> Input N
  - 3> sum = 0
  - 4> while ( $N \neq 0$ )
    - Rem =  $N \% 10$ ;
    - sum = sum + Rem;
    - $N = N / 10$ ;
  - 5> print sum.

Flowchart :-

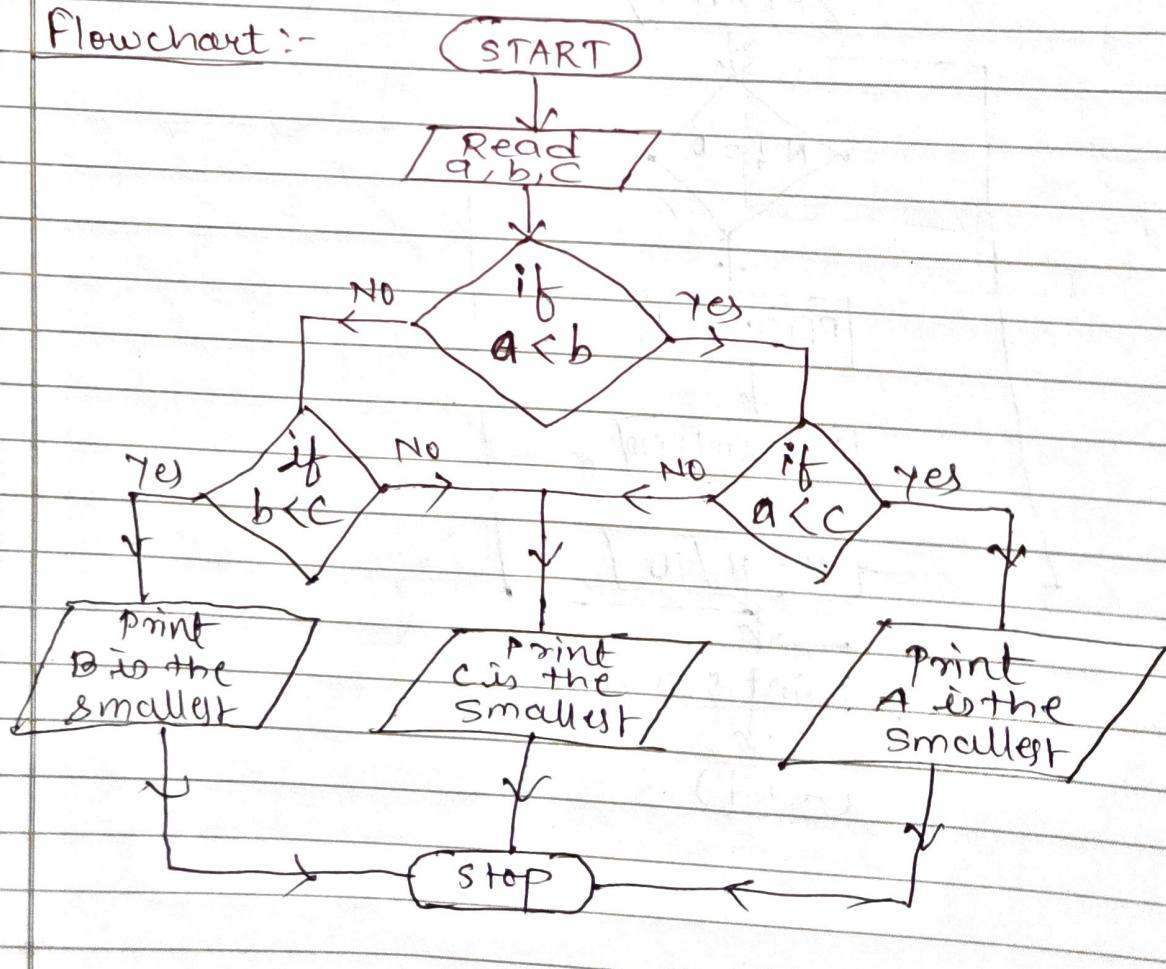




Date :

- 11) write a program to find smallest to 3 numbers
- Algorithm :-
- 1) start
  - 2) Take three numbers in  $a, b, c$
  - 3) check if  $a$  is less than  $b$ .
  - 4) If above condition is true, go to step 5,  
else go to step 7.
  - 5) check if  $c$  is less than  $a$ .
  - 6) If above condition is true,  $c$  is the  
smallest.
  - 7) else  $a$  is the smallest. Go to step 9.
  - 8) check if  $b$  is less than  $c$ .
  - 9) If above cond'n is true,  $b$  is smallest
  - 10) else  $c$  is the smallest.
  - 11) stop.

Flowchart :-





Date:

12) How to add two numbers without using arithmetic operators in Java

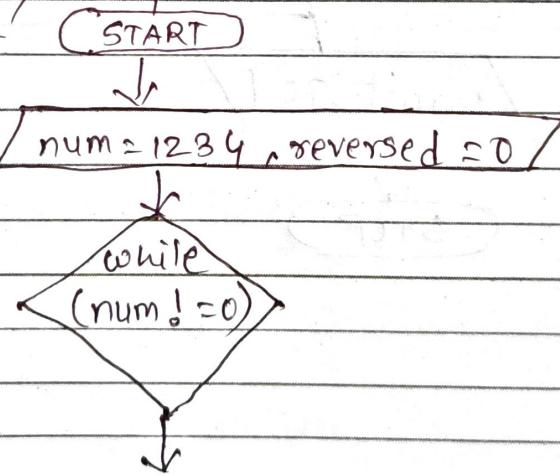
- Algorithm :-
- 1) The user inputs the first number.
  - 2) store the number in variable num1.
  - 3) The user inputs the second number
  - 4) store the number in variable num2
  - 5) For loop used for the addition of two numbers.
  - 6) Display the sum of numbers on the screen

13) Reverse given number.

Algorithm :-

- 1) Start
- 2) initialize num = 1234, reversed = 0.
- 3) while (num != 0)
- 4) digit = num % 10  
reversed = reversed \* 10 + digit  
$$\text{num} = \frac{\text{num}}{10}$$
- 5) print the num reversed num.
- 6) stop.

Flowchart:-



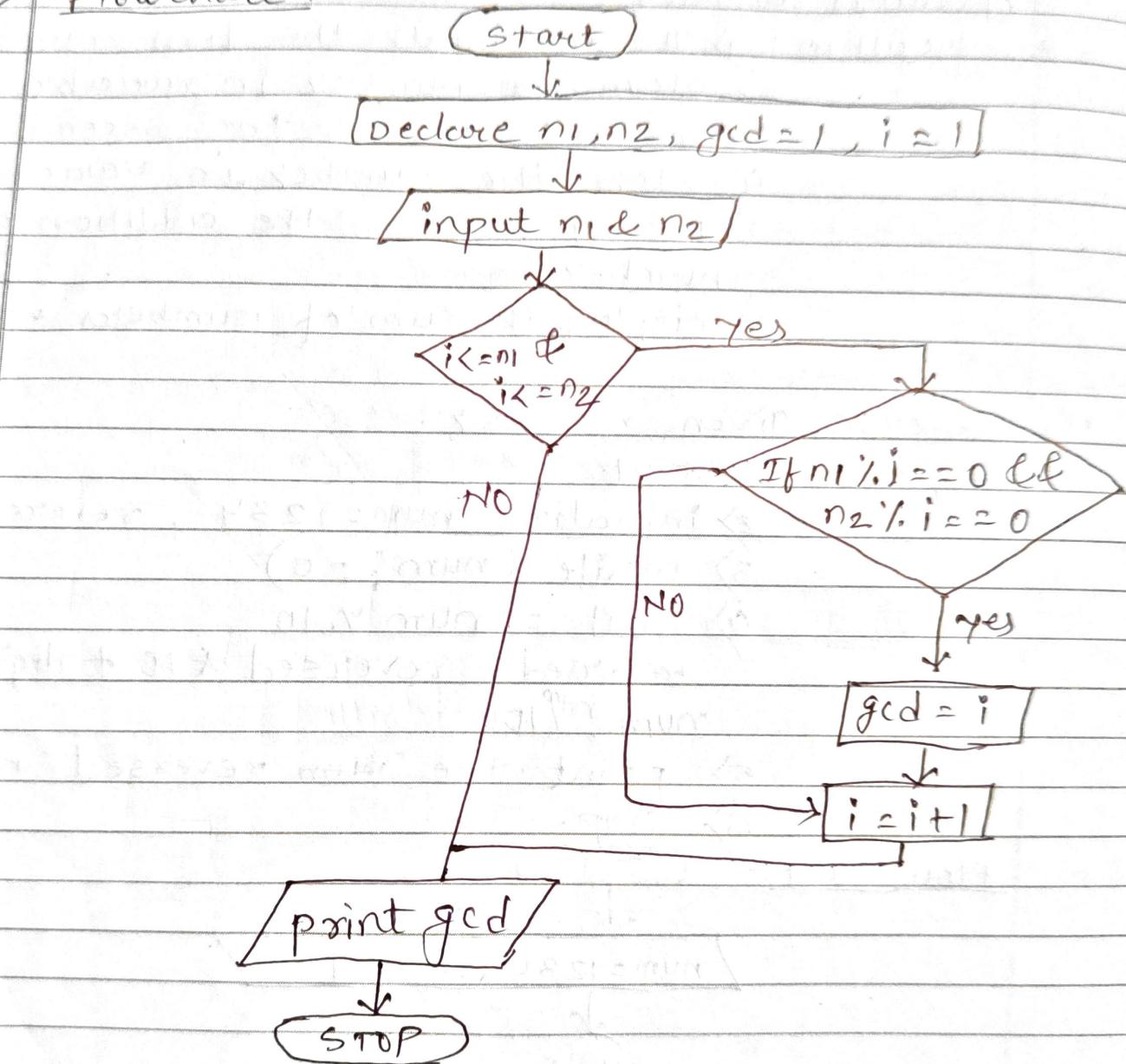
```
digit = num % 10  
reversed = reversed  
* 10 + digit, num = num / 10
```





Date :

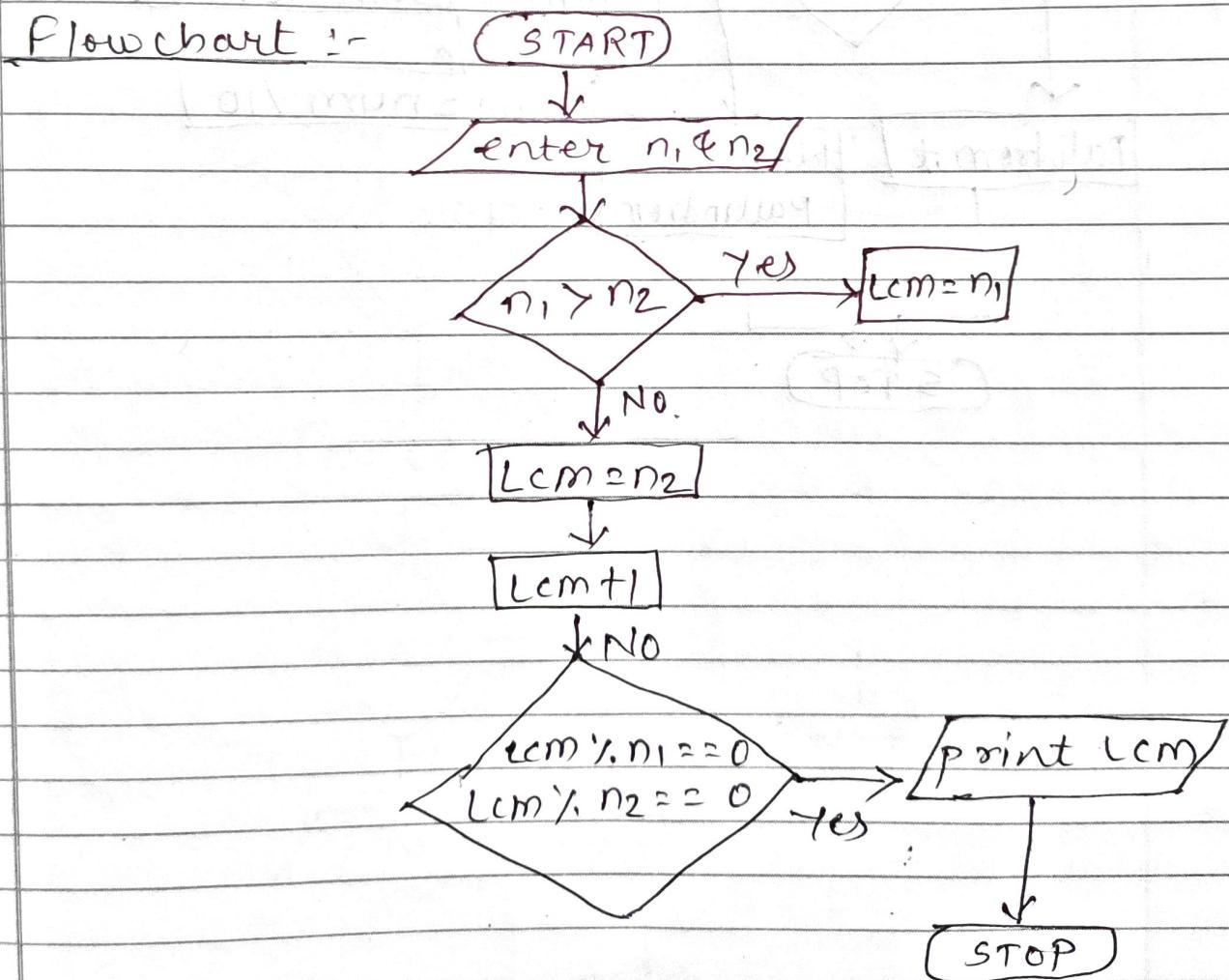
- 1\*) Find GCD of two given number.  
→ Flowchart :-





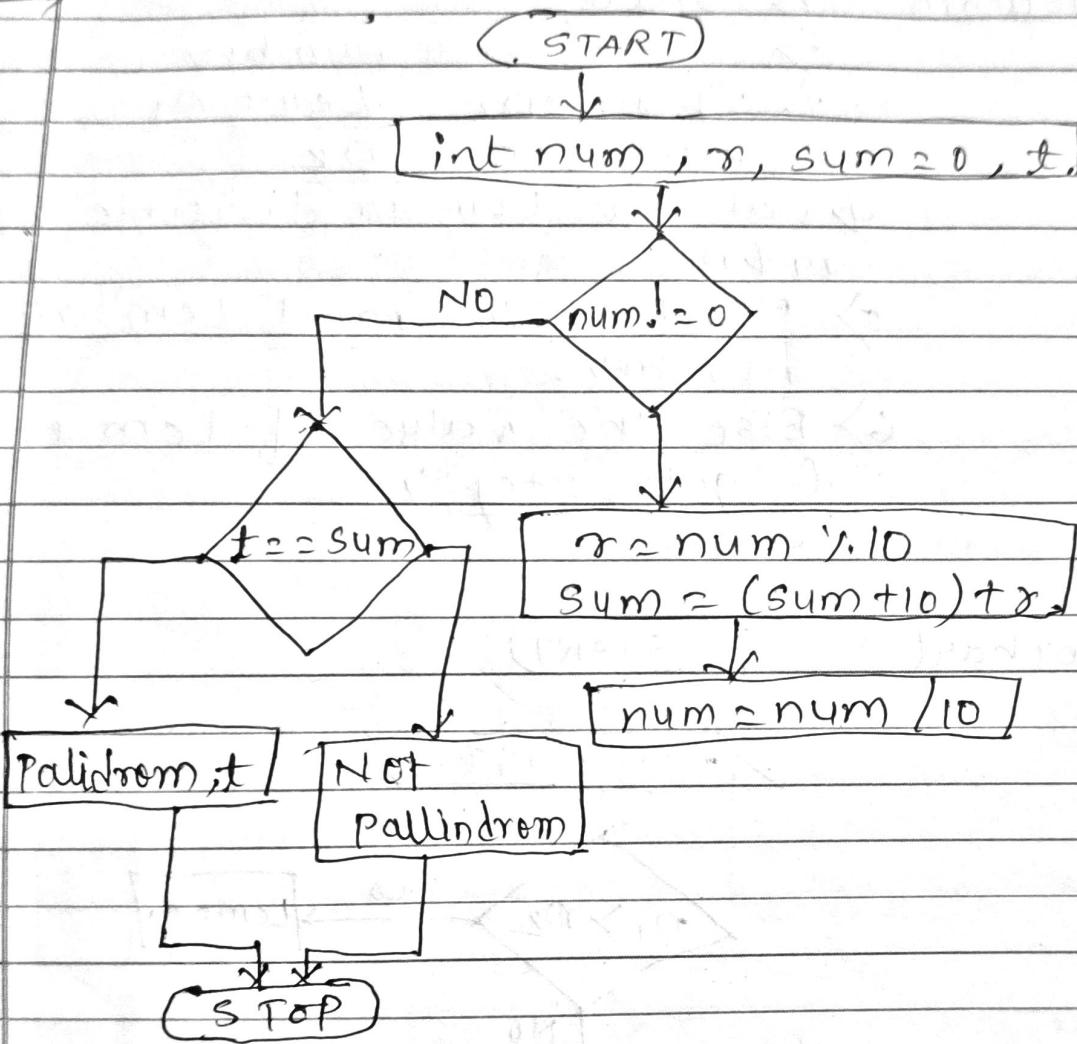
- LCM of two given number
- Algorithm :-
- 1) Start
  - 2) Accept two number
  - 3) If  $n_1 > n_2$   $LCM = n_1$   
else  $LCM = n_2$
  - 4) Validate LCM is divisible by both  $n_1$  &  $n_2$
  - 5) If divisible print LCM of two number.
  - 6) Else the value of LCM is increased & goto step 4
  - 7) Stop.

Flowchart :-





17) Given number is palindrome or NOT.

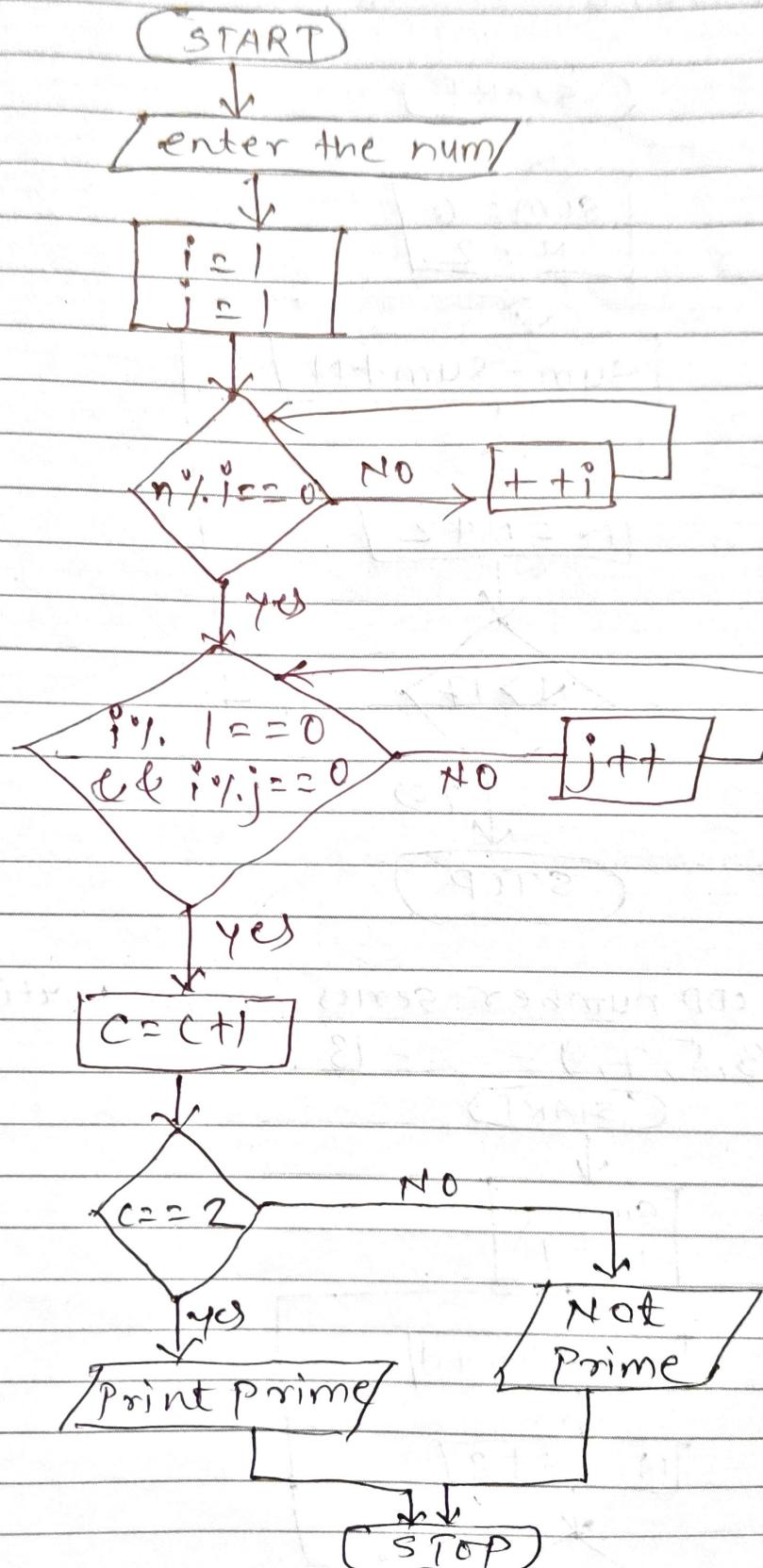




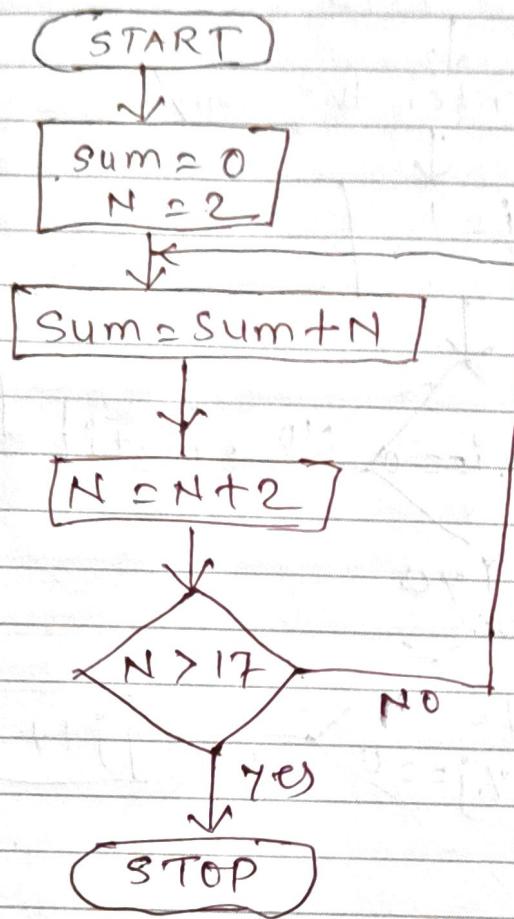
Date :

18)

print all the prime factors of given number



- 19) print even number from particular range  
2, 4, 6, 8 - - 16.



- 20) Print odd number series from particular range  
1, 3, 5, 7, 9 - - 13.

