Assignment - 1

Write Algorithm or Flowchart

① Check if the given number is EVEN or ODD.

Algorithm :-

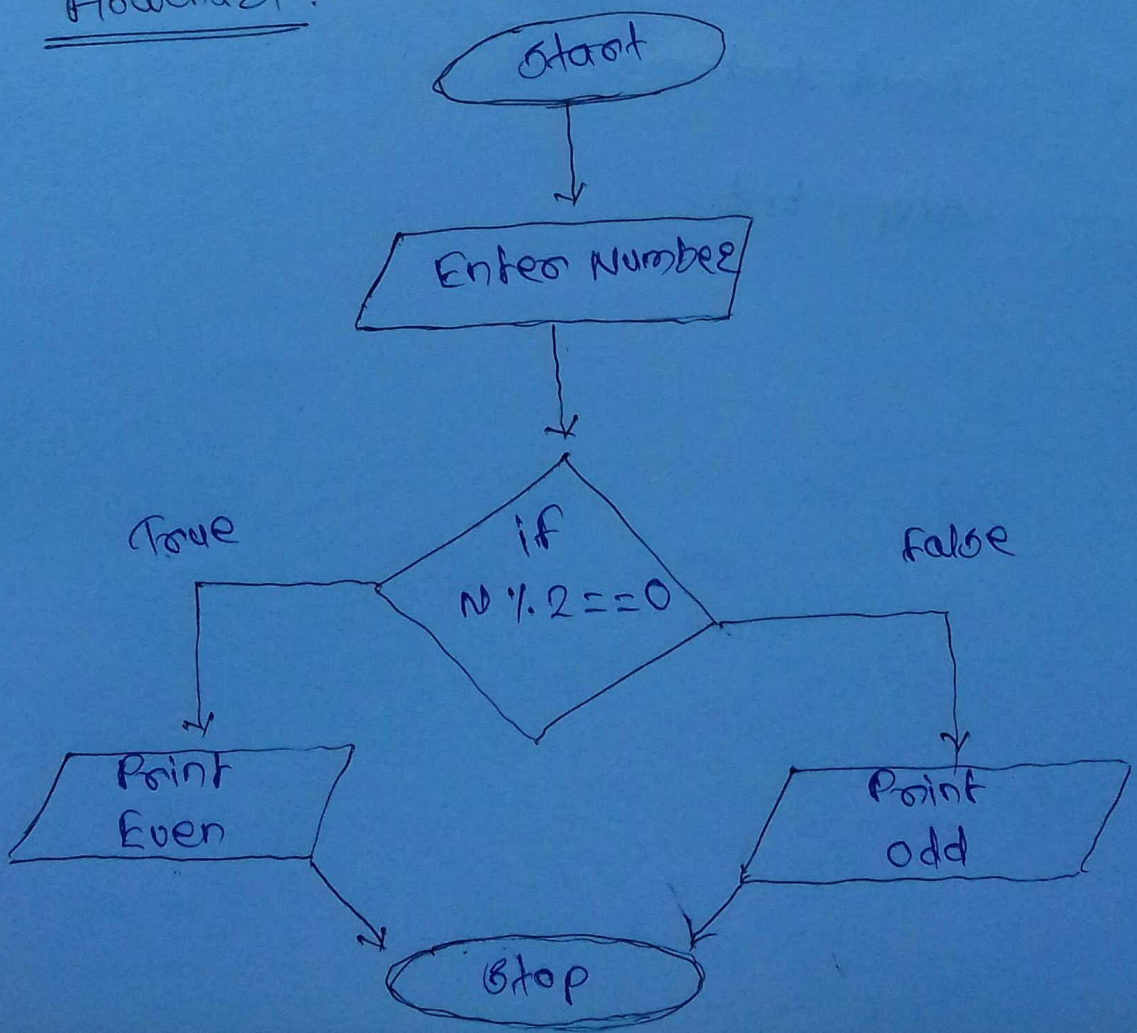
Step 1 :- Start the program

Step 2 :- Read / Input the number

Step 3 :- if n % 2 == 0 then print number is even

Step 4 :- else ... print number is odd.

Flowchart :-



② Find the factorial of a given number.

Algorithm :-

Step 1 :- Start

Step 2 :- Read a Number. n

Step 3 :- Initialize variables

i = 1, f = 1

Step 4 :- if i <= n go to Step 4
otherwise go to Step 7

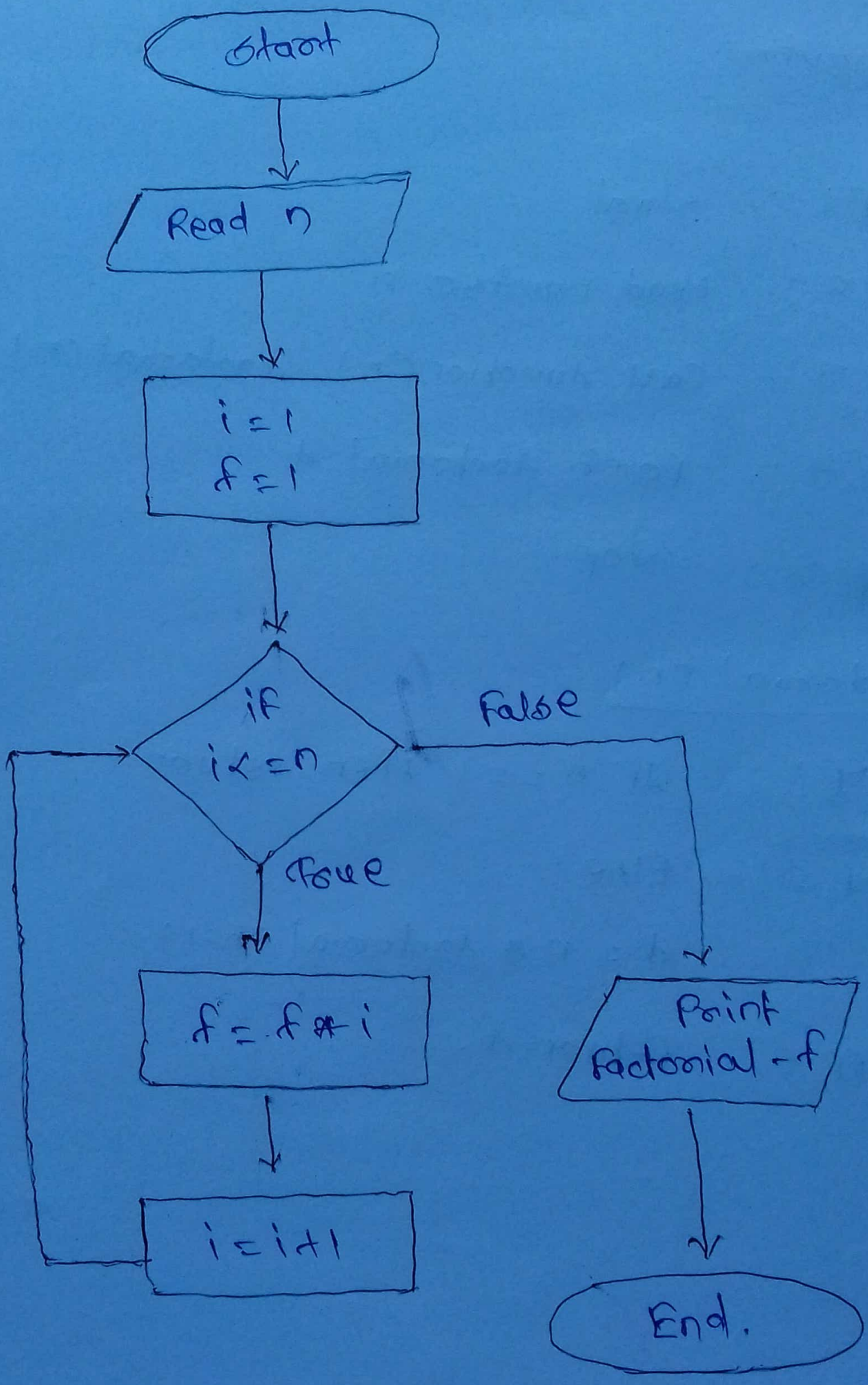
Step 5 :- Calculate f = f * i

Step 6 :- Increment the i by 1 (i = i+1)
and go to Step 4

Step 7 :- Print factorial 'f'

Step 8 :- Stop / End.

Flowchart :-


start
Read n
i = 1
f = 1
if
i <= n
False
True
f = f * i
i = i + 1
Print
factorial - f
End.

(3) Find the factorial of a number using Recursion.

Algorithm:-

Step 1 :- Start

Step 2 :- Read number n

Step 3 :- Call ~~function(n)~~ 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.

(4) Swap two numbers without using the third variable approach.

Algorithm

Step 1 :- Start

Step 2 :- Read a, b

Step 3 :- a = a+b

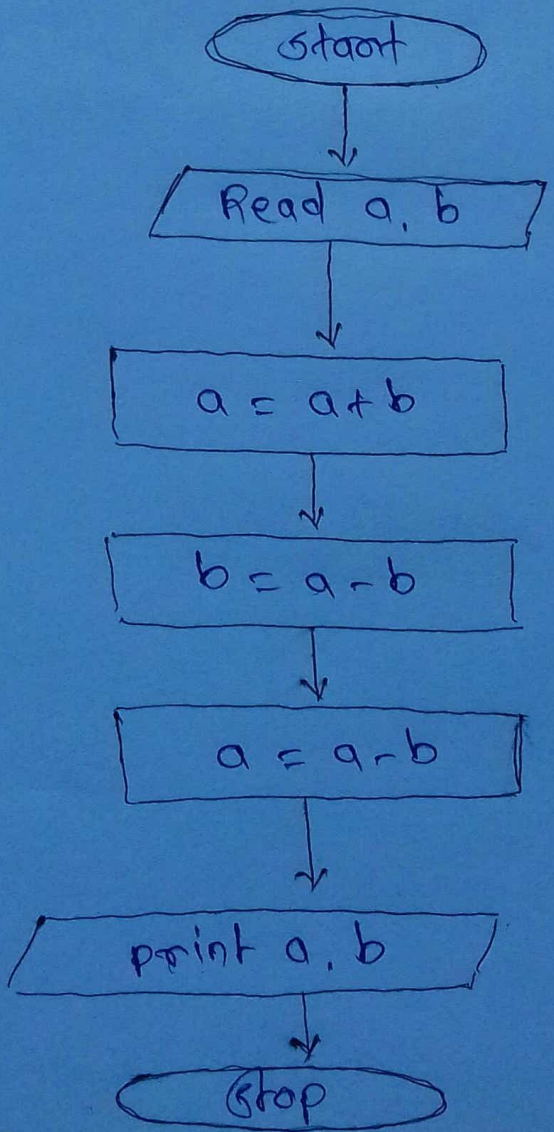
Step 4 :- b = a-b

Step 5 :- a = a-b

Step 6 :- print a, b

Step 7 :- Stop.

Flowchart



(5) check whether the given number is Positive or Negative.

Algorithm :-

Step 1 :- Start

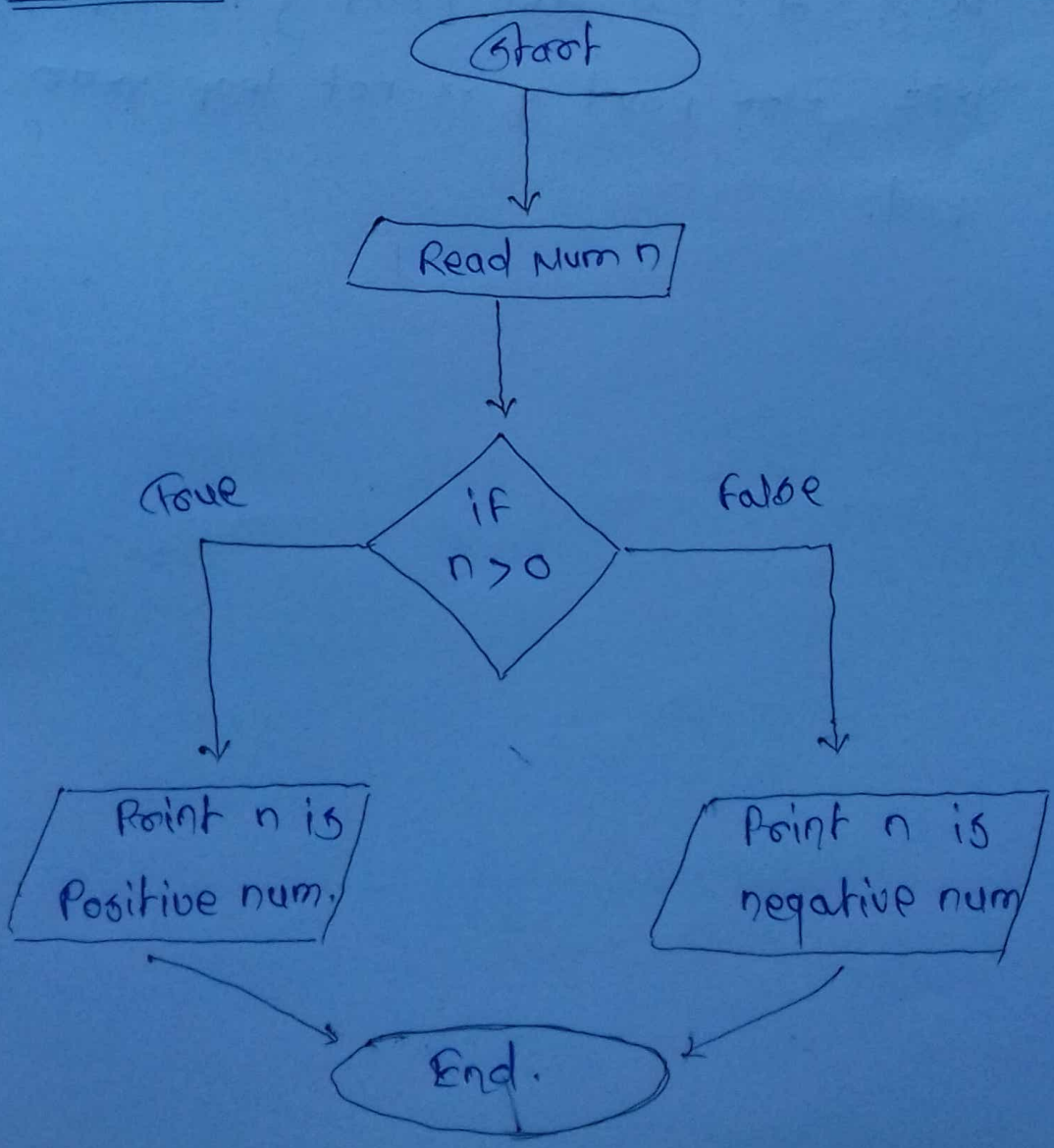
Step 2 :- Read Number n

Step 3 :- if n > 0 ; print positive number

Step 4 :- else, print negative number

Step 5 :- Stop / End.

Flowchart :-



⑥ find whether a given number is Leap year or NOT.

Algorithm :-

Step 1 :- Start

Step 2- Read / Enter year y.

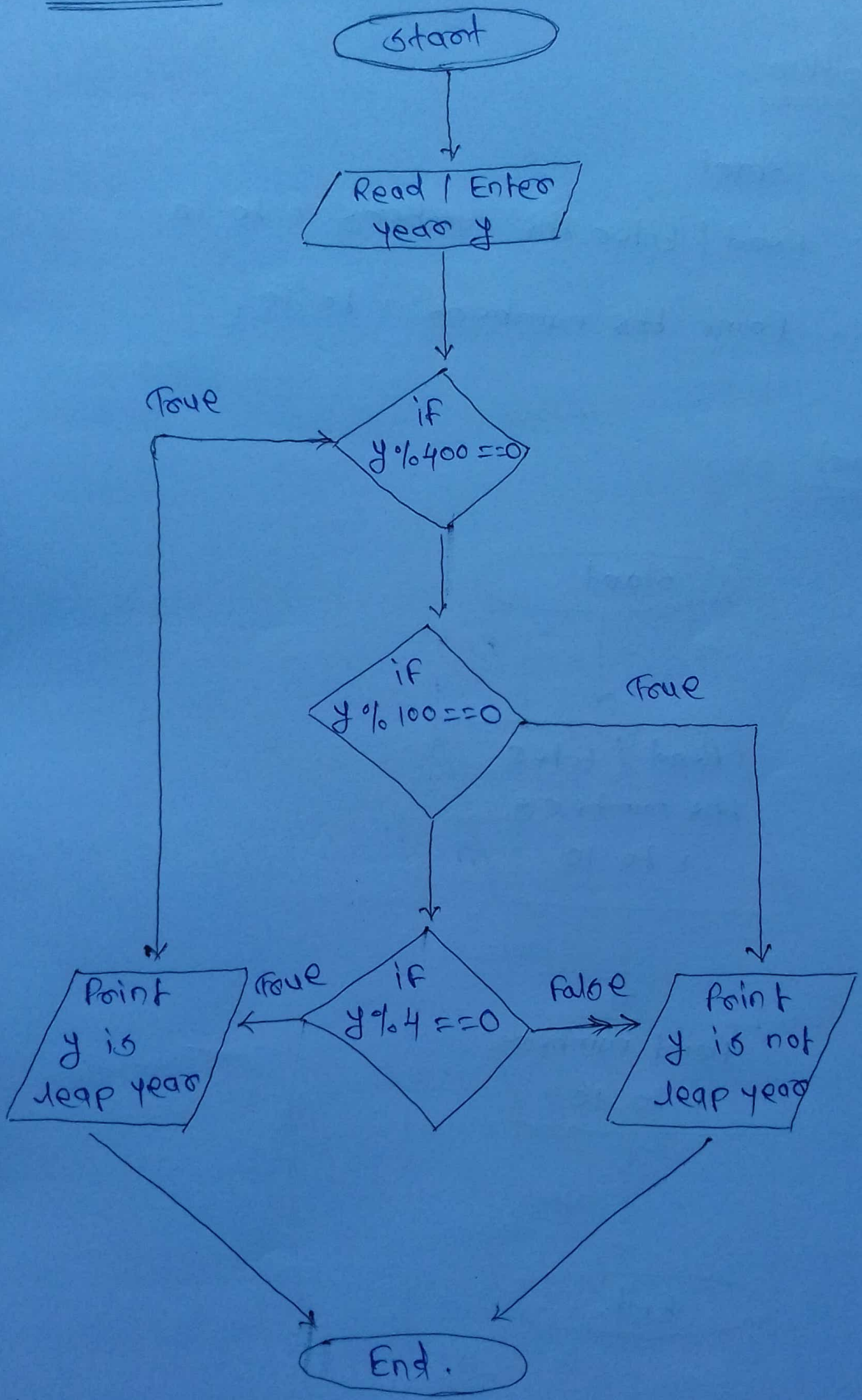
Step 3 :- if y %. 400 == 0 then print y is leap year
else go to Step 4

Step 4 :- if y %. 100 == 0 then print y is not leap year & go to Step 5.

Step 5 :- if y %. 4 == 0 then print y is leap year. else print y is not leap year.

Step 6 :- End.

Flowchart :-


Start
Read / Enter year y
if y%400 ==0
Toue
if y%100==0
Toue
if y%4==0
Toue
False
Print y is leap year
Print y is not leap year
End.

⑦ Print 1 to 10 - without using loop.

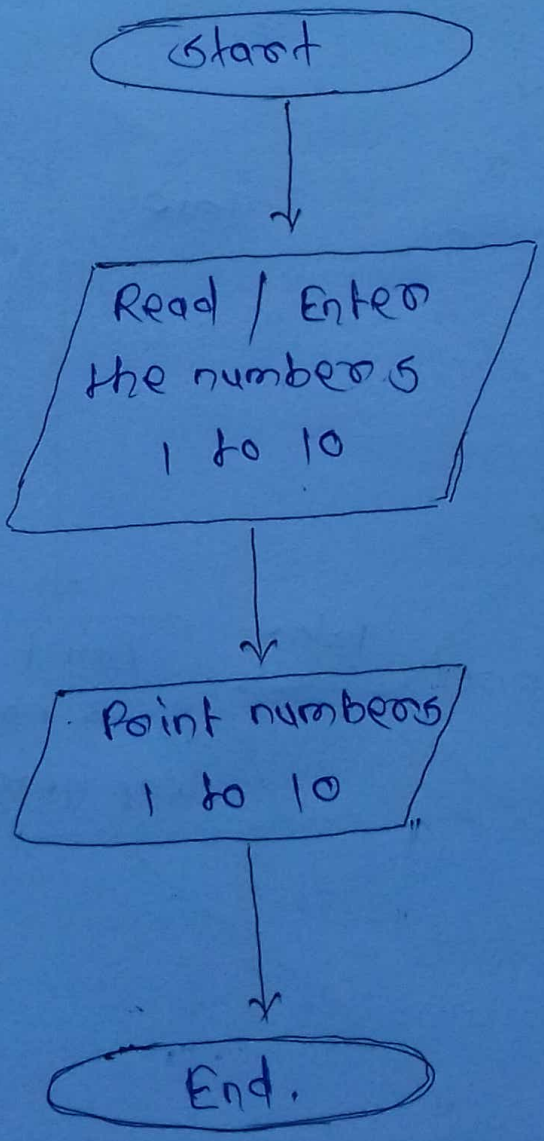
Algorithm:-

Step 1 :- Start.

Step 2 :- Read / Enter the numbers 1 to 10.

Step 3 :- Print the numbers 1 to 10.

Flowchart :-



(8) To print the digits of a Given Numbers.

Algorithm :-

Step 1 :- Start

Step 2 :- Read / Enter the number n

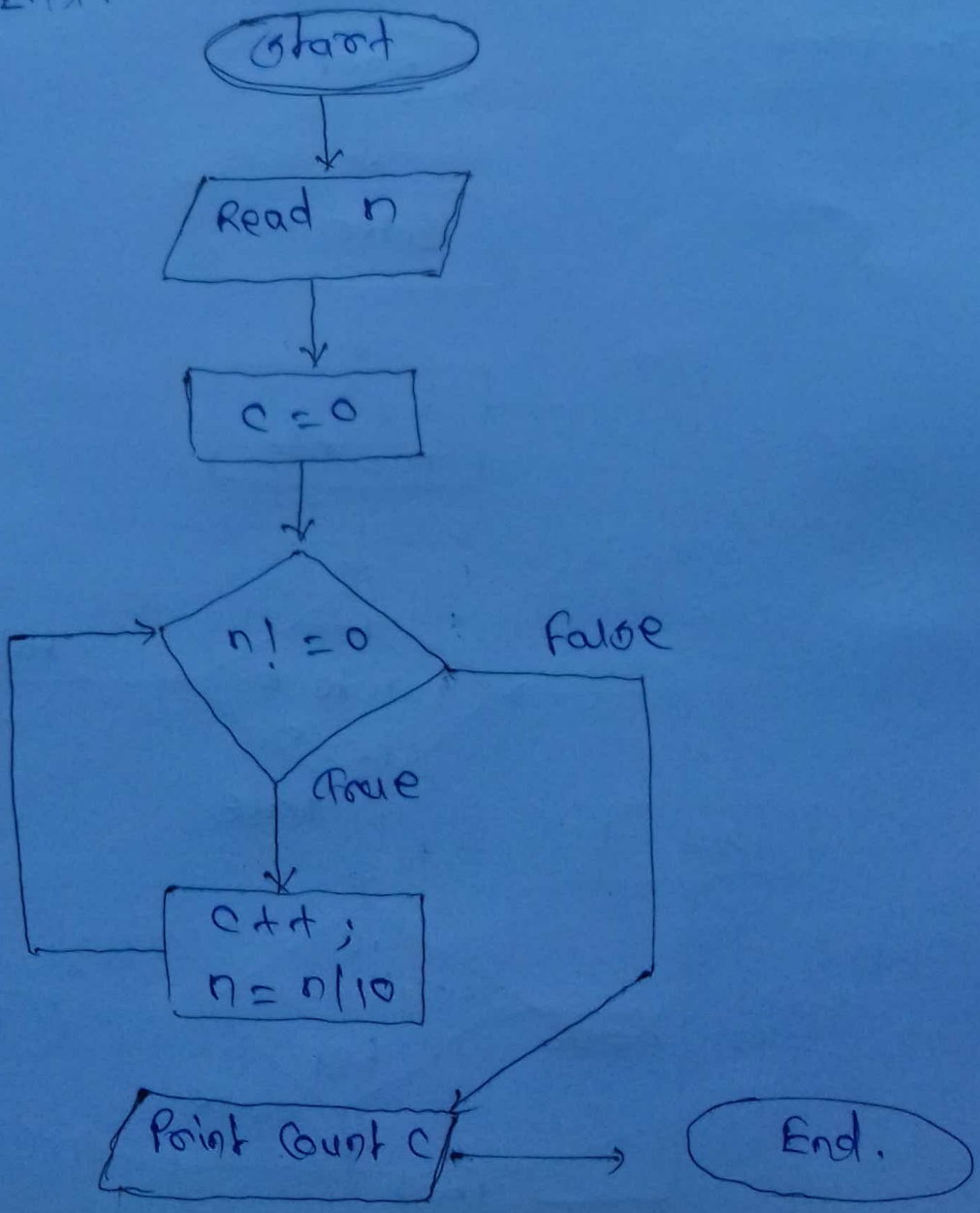
Step 3 :- Set count c = 0

Step 4 :- while (n != 0) Increase count c++
& divide number n = n/10;

Step 5 :- Print count

Step 6 :- End.

Flowchart :-



⑨ To print all the factors of the given numbers.

Algorithm :-

Step 1 :- Start

Step 2 :- Read / Enter the number n

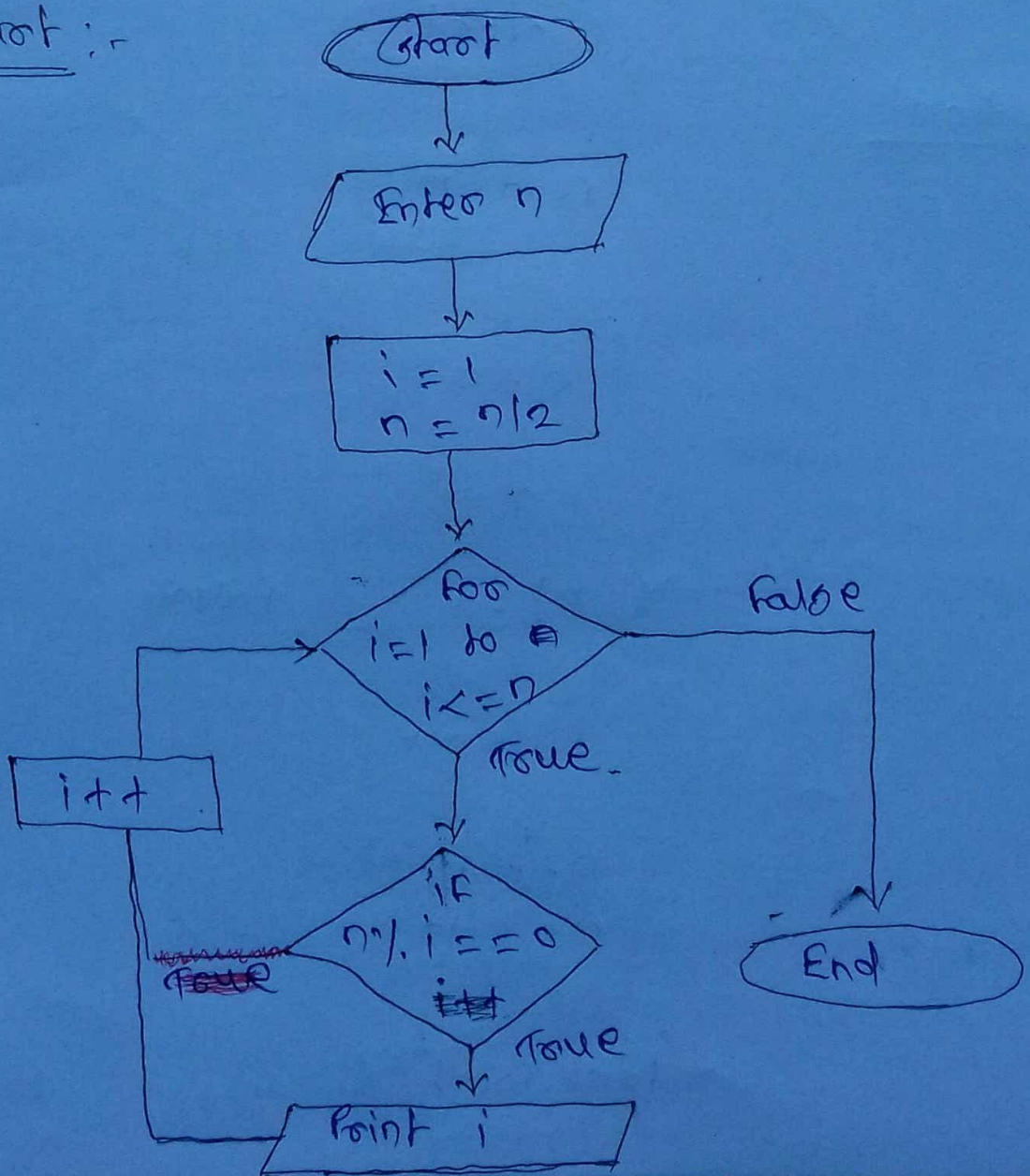
Step 3 :- Set i=1 & n = n/2

Step 4 :- if (n %. i == 0) then i++ (for i=1 to n)

Step 5 :- Print i

Step 6 :- End.

Flowchart :-



(10) Find the sum of the digits of a given number

Algorithm :-

Step 1 :- Start

Step 2 :- Read Number n

Step 3 :- Set s=0;

Step 4 :- while(n!=0), r= n % 10

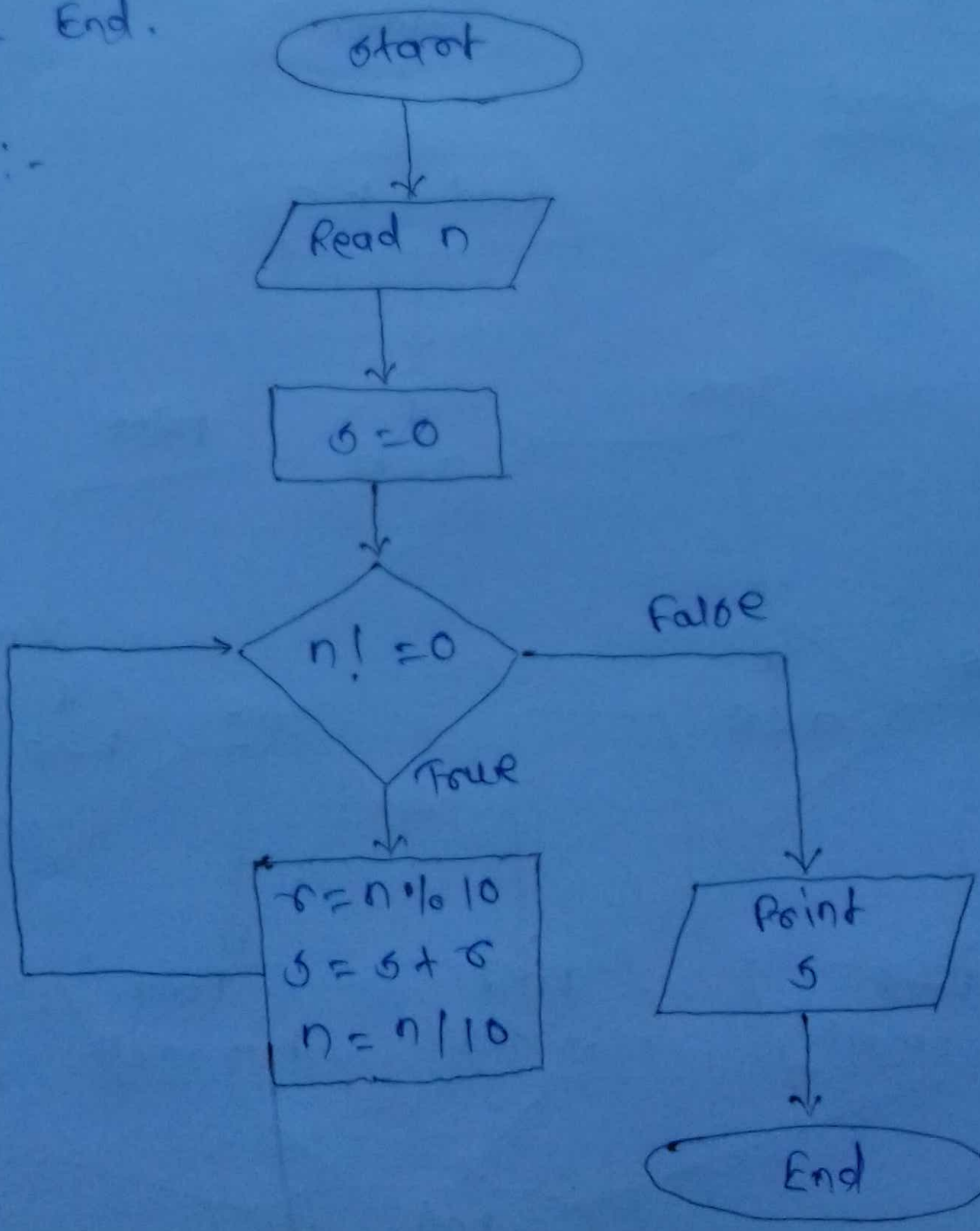
s = s+r

n = n/10

Step 5 :- Print s

Step 6 :- End.

Flowchart :-



(11) Find the smallest of three numbers :-

Algorithm :-

Step 1 :- Start

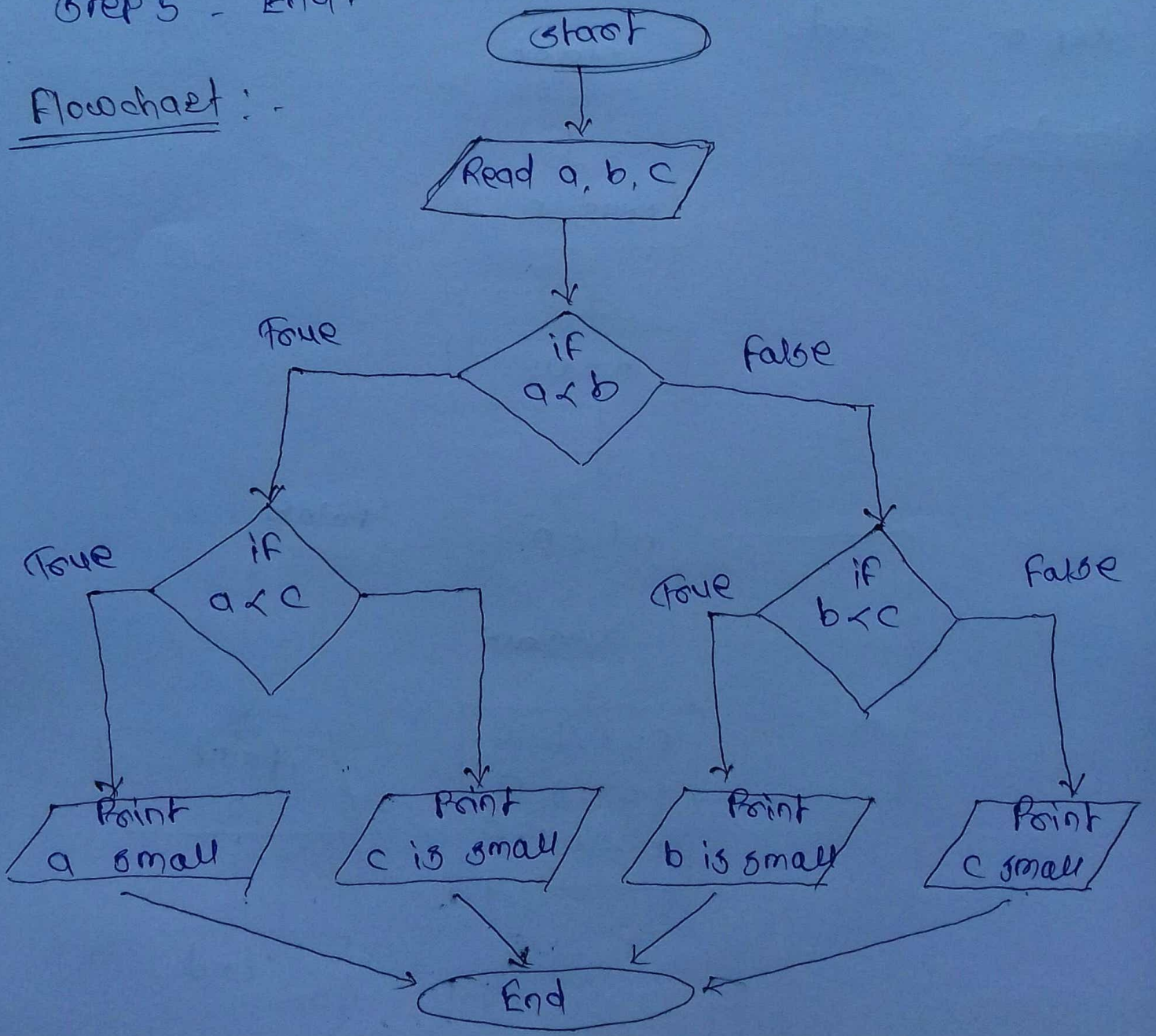
Step 2 :- Read Numbers, a, b, c

Step 3 :- check if $a < b$ and $a < c$, print a small
else print c small.

Step 4 - check if $a < b$ False, check $b < c$ true
print b small else a-small.

Step 5 - End.

Flowchart :-



(12) Add two numbers without using arithmatic operators.

Algorithm :-

Step 1 :- Start

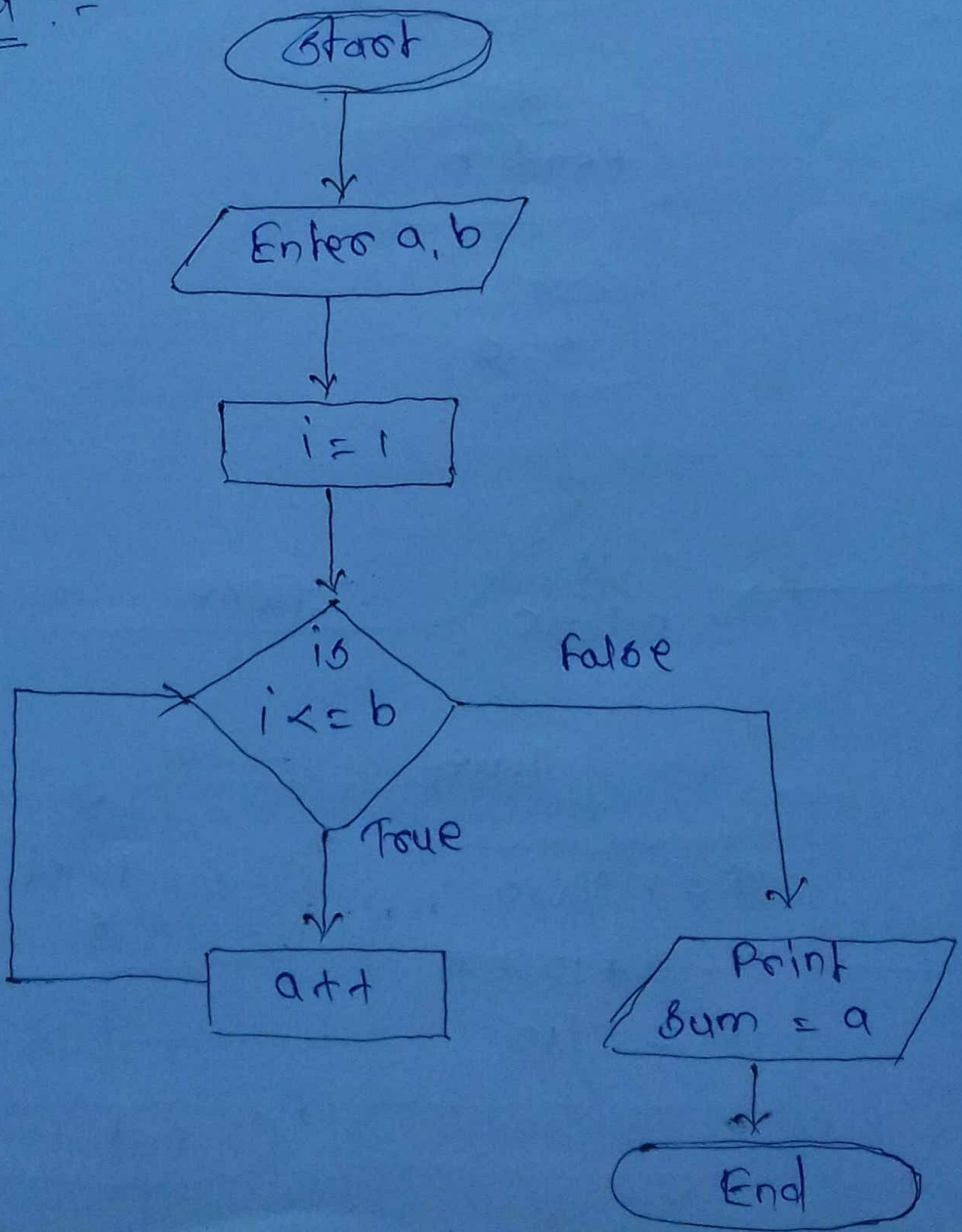
Step 2 :- Read / Enter a, b

Step 3 :- Set i = 1

Step 4 :- for (i=1 ; i<=b ; i++)
a++ ;

Step 5 :- When step 4 false print sum = a.

Flowchart :-

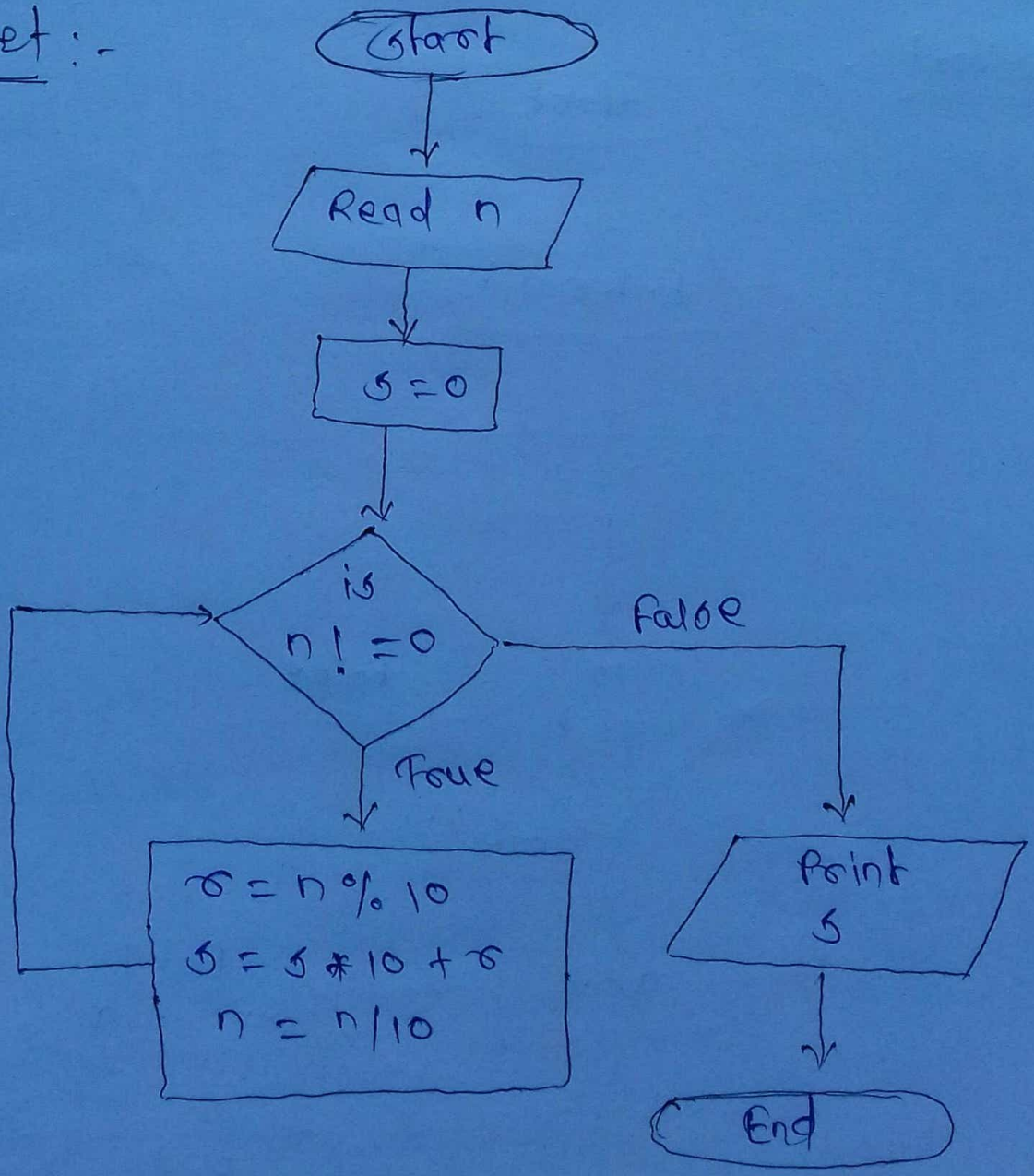


(13) Program to Reverse a given number.

Algorithm :-

Step 1 :- Start
Step 2 :- Read n
Step 3 :- s = 0
Step 4 :- while(n!=0) → r = n % 10;
s = s * 10 + r;
n = n / 10;
Step 5 :- Print s
Step 6 :- End.

Flowchart :-



(14) To find GCD (Greatest Common Divisor) (HCF) of two given numbers.

Algorithm:-

Step 1 :- Start

Step 2 :- Read n_1, n_2

Step 3 :- if ($n_1 = n_2$) then Go to Step 5

Step 4 :- if ($n_1 > n_2$)

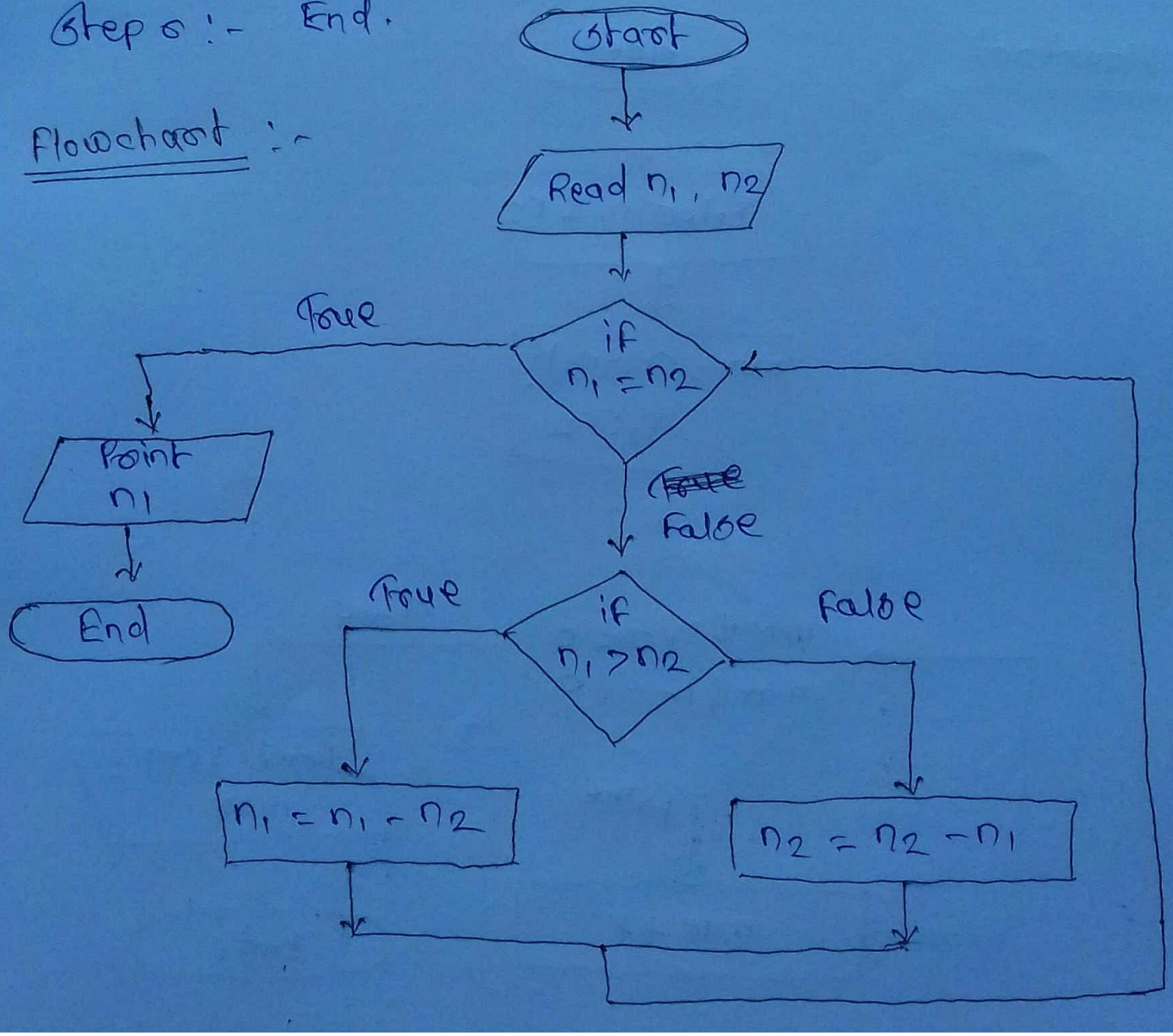
then $n_1 = n_1 - n_2$ Go to Step 3

else $n_2 = n_2 - n_1$ Go to Step 3

Step 5 :- Print n_1

Step 6 :- End.

Flowchart :-



(15) To find LCM of a two given numbers.

Algorithm :-

Step 1 :- Start

Step 2 :- Enter n_1 & n_2

Step 3 :- Find max ($(n_1 > n_2)$? $n_1 : n_2$)

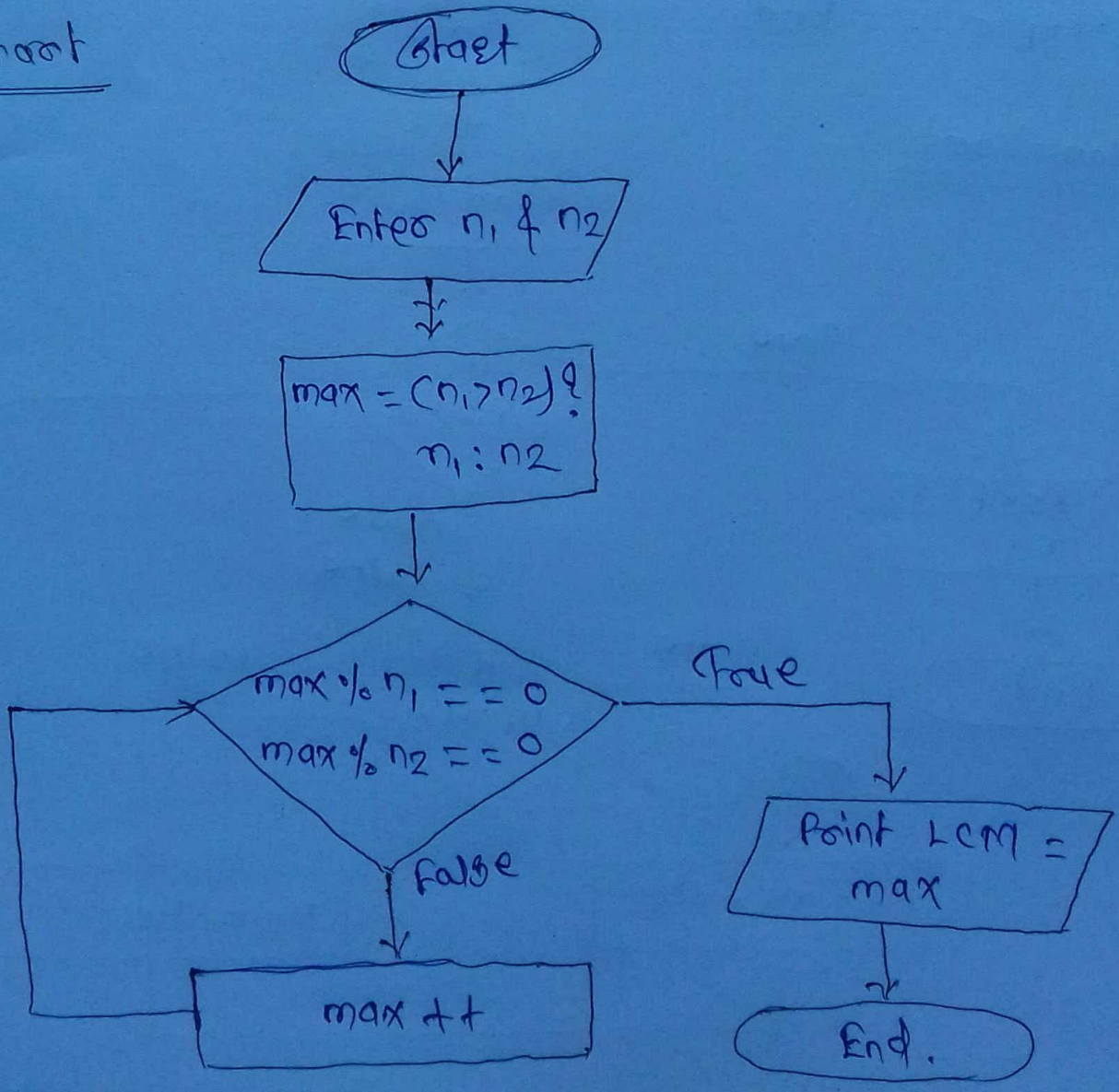
Step 4 :- if max % n_1 == 0 & max % n_2 == 0

print LCM = max

Step 5 :- else increment max by 1 & go to Step 4.

Step 6 :- Stop / End.

Flowchart



(16) Program to find LCM of two given numbers using Prime factors.

Algorithm :-

Step 1 :- Start

Step 2 :- Read / Enter the numbers n_1 & n_2;

Step 3 :- find max → ($n_1 > n_2$? n_1 : n_2)

Step 4 :- while (max % n_1 != 0 || max % n_2 != 0)

max = max * i

i++

Step 5 :- Print the value of max

Step 6 :- End.

(17) check whether the given number is Palindroome or NOT.

Algorithm:-

Step 1 :- Start

Step 2 :- Read / Enter number n

Step 3 :- temp = n , s = 0

Step 4 :- while (temp ! = 0)

r = temp % 10 ;

s = s * 10 + r ;

temp = temp / 10 ;

end while

Step 5 :- if (s == n)

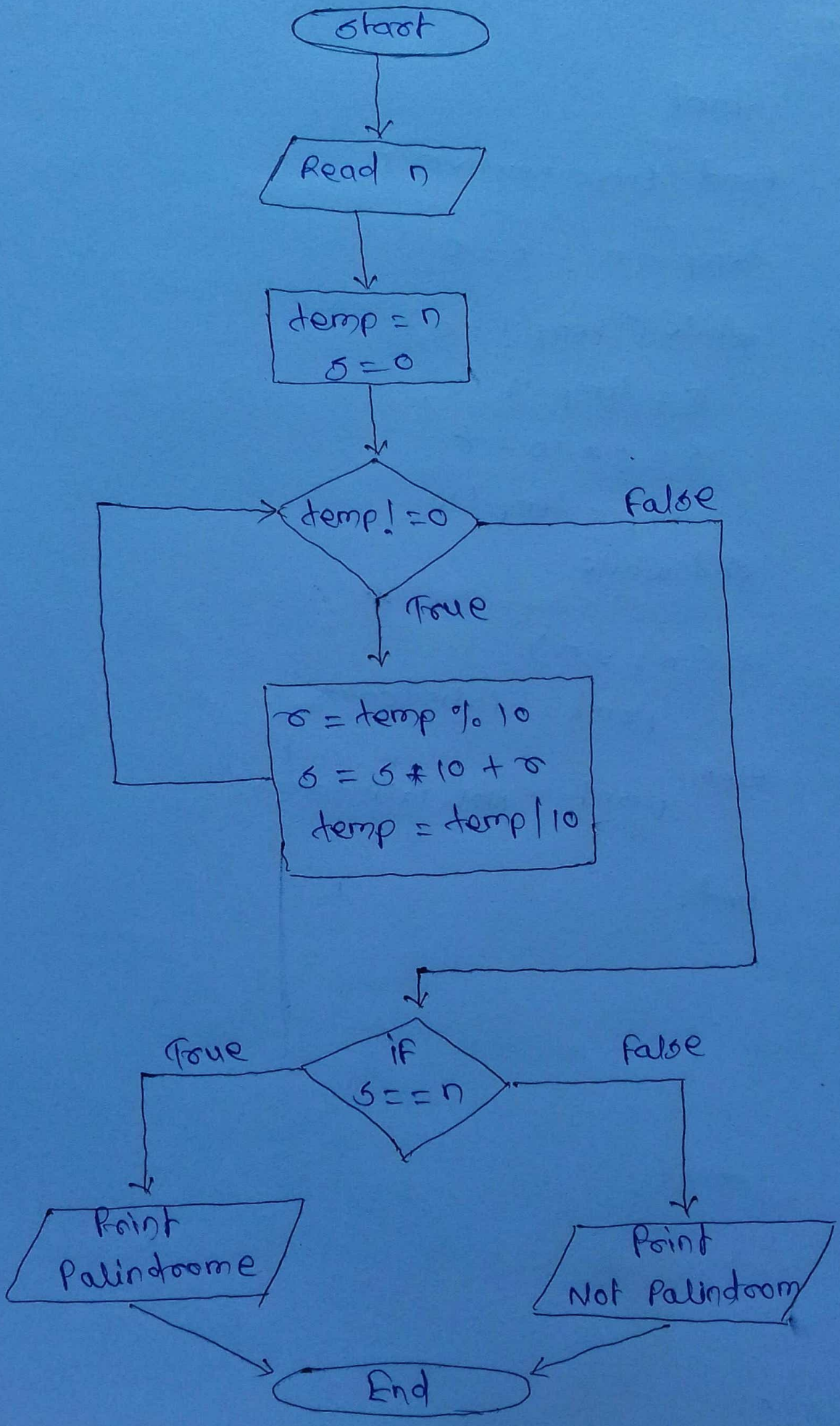
print - "Palindroome"

else

print " Not Palindroome"

Step 6 :- End.

Flowchart


start
Read n
temp = n
s = 0
temp != 0
False
True
r = temp % 10
s = s * 10 + r
temp = temp / 10
if
s == n
True
False
Print
Palindroome
Print
Not Palindroom
End

(18) Program to print all the Prime factors of the given number.

Algorithm :-

Step 1 :- Start

Step 2 :- Read Number n,

Step 3 :- for(i=2; i*i<=n; i++)

create while loop

while (n % i == 0) then divide n

n/=i;

Print value of i

Step 4 :- if(n>1) then print remaining value of i

Step 5 :- End.

(19) To print the EVEN numbers in series within given range.

Algorithm :-

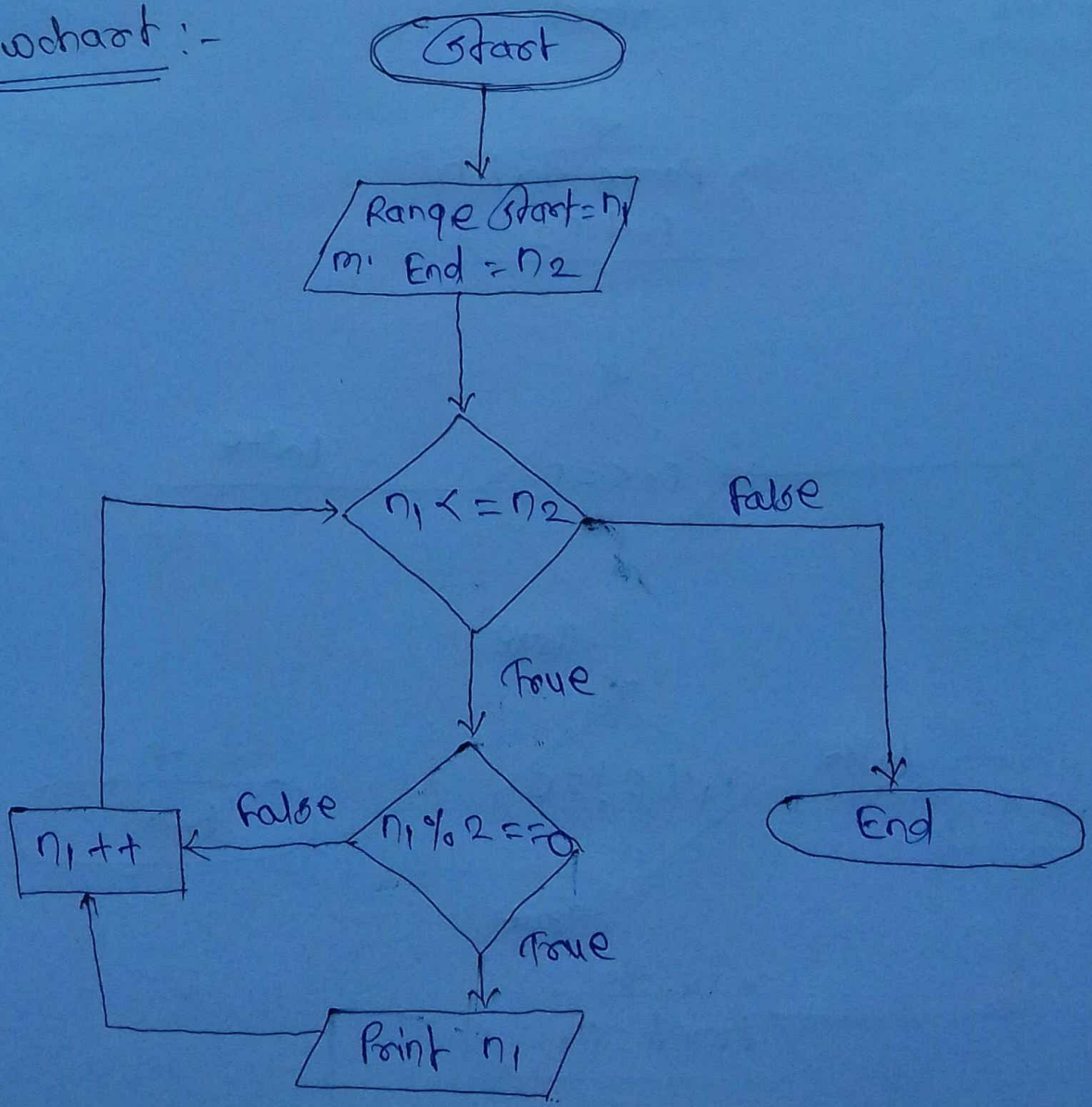
Step 1 :- Start

Step 2 :- Enter Range N_1 & N_2
Start = N_1 End = N_2

Step 3 :- check if $n_1 <= n_2$ true then
$n_1 \% 2 == 0$ → print n_1
else - n_1++

Step 4 :- ~~else~~ Stop.

Flowchart :-



(20) To print the odd numbers in a series with given range.

Algorithm :-

Step 1 :- Start

Step 2 :- Read Range Start = n_1, End = n_2

Step 3 :- check if $n_1 <= n_2$ true

$n_1 \% 2 != 0$ print n_1

else n_1++

Step 4 :- When Step 3 False — End / Stop.

Flowchart :-