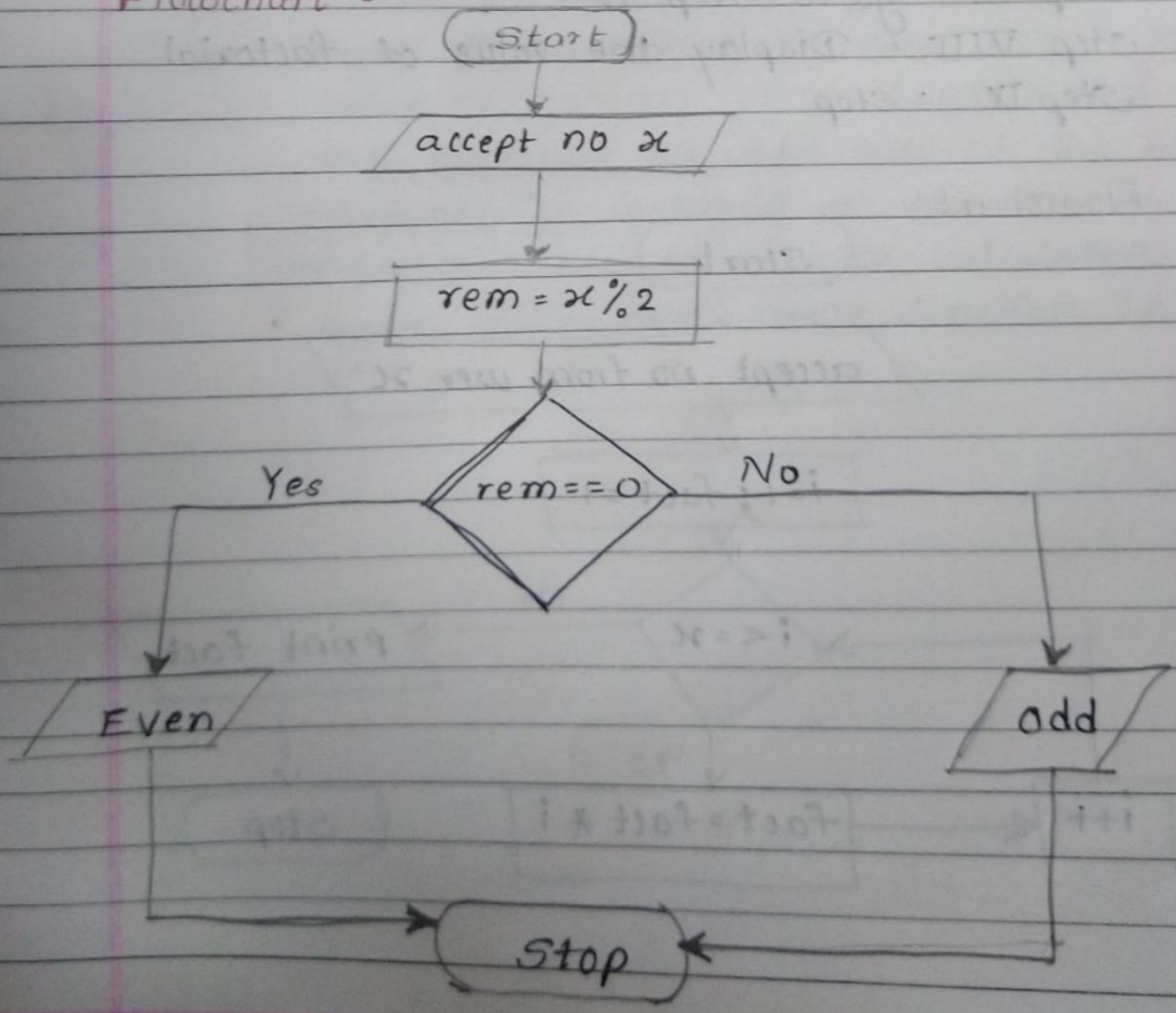
Q] Check Given no is even or odd

Algorithm

Step I - Declare the required Variables
Step II - Enter no. from user or accepting no. from user
Step III - Using modulus operator find remainder
If remainder zero then even no else no. is odd
Step VI - Display the result
Step V - Stop

Flowchart :

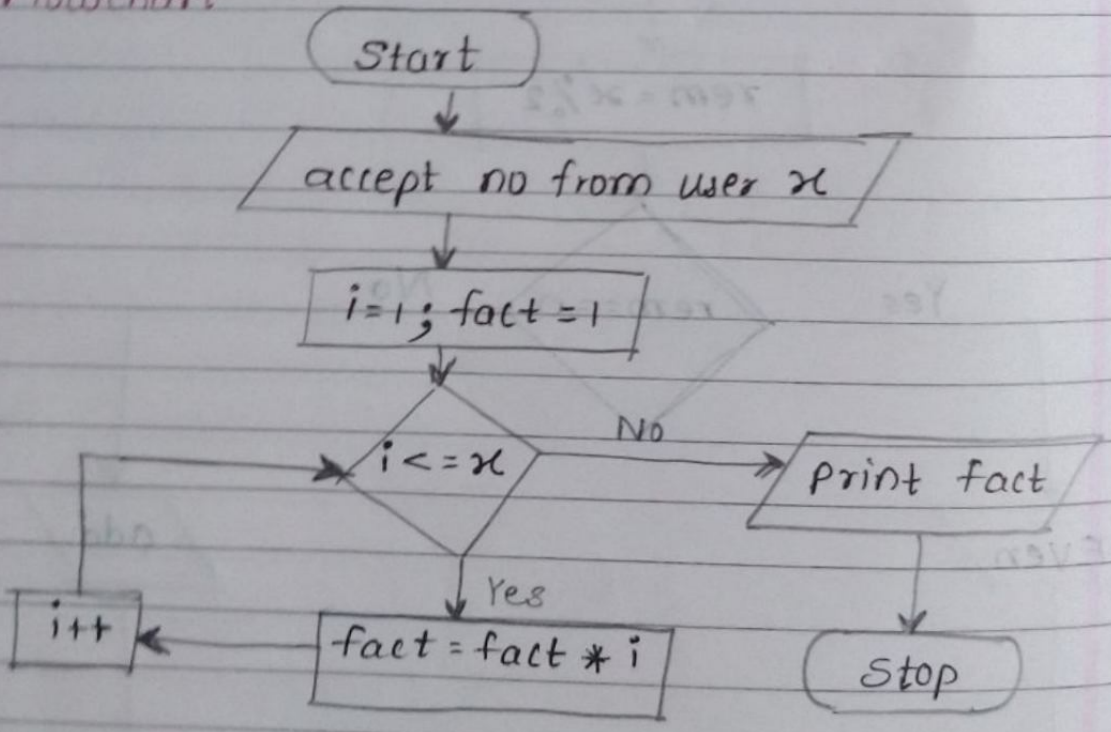


2] Find the Factorial of given no.

Algorithm

Step I - Declare the required Variable
Step II - accept no from user
Step III - Initialise fact Variable as 1 & counter as 1
Step IV - Check if counter greater than the user entered Variable, it yes then go to Step IX, if no then go step V
Step V - Multiply the fact Variable with counter Value
Step VI - Increment the counter Variable
Step VII - go to step IV
Step VIII - Display the Value of factorial
Step IX - Stop

Flowchart



Q] Find the factorial no. using recursion

Algorithm :

main () function

Step I : Declare the required Variable

step II : accept no from user

Step III : Call the factorial function, passing required parameter to it & accept factorial return by the function

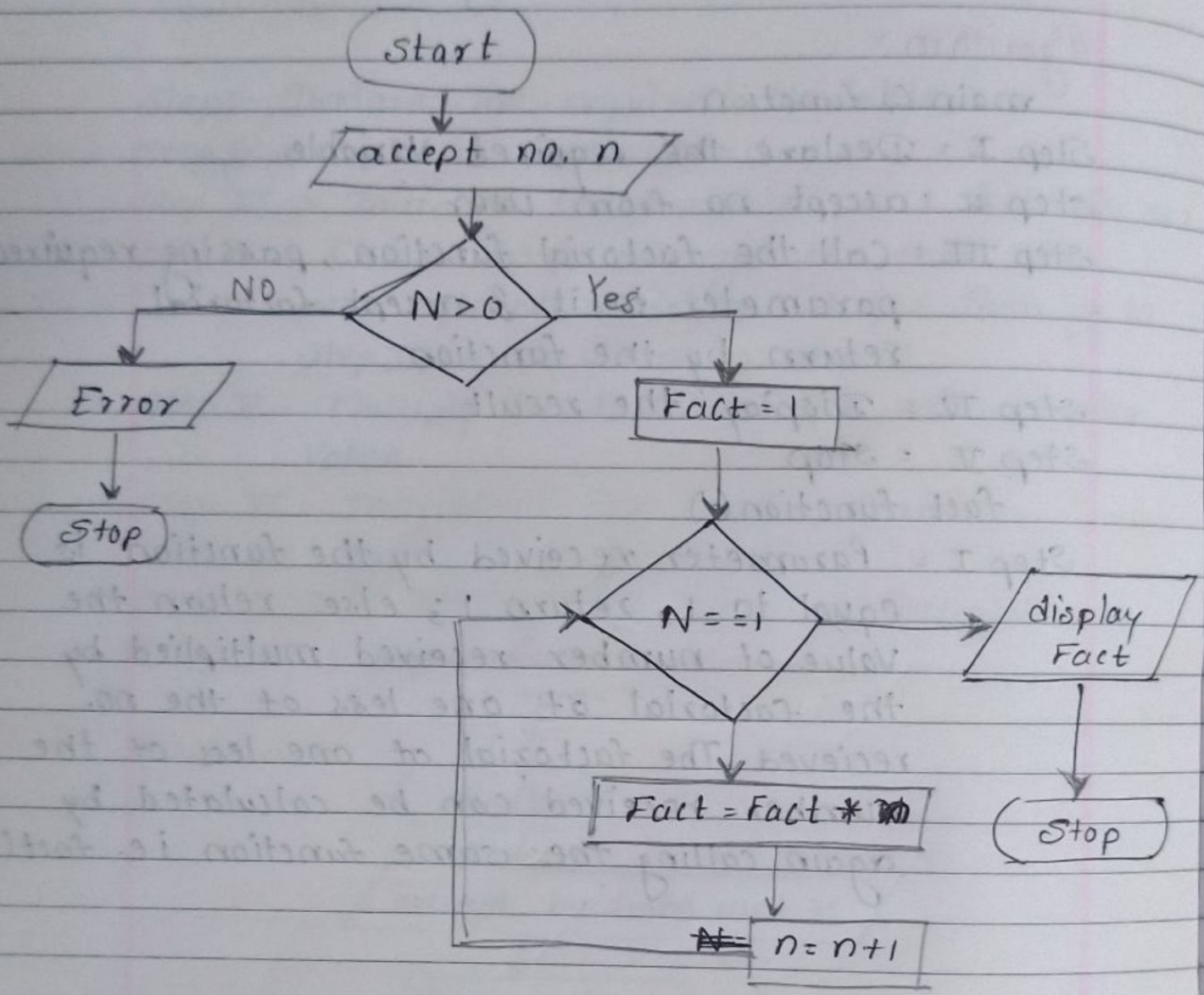
Step IV : Display the result

Step V : Stop

fact function ()

Step I : Parameter received by the function is equal to 1, return 1; else return the Value of number received multiplied by the factorial of one less of the no. recieved. The factorial of one less of the number received can be calculated by again calling the same function i.e. fact()

Flowchart :-

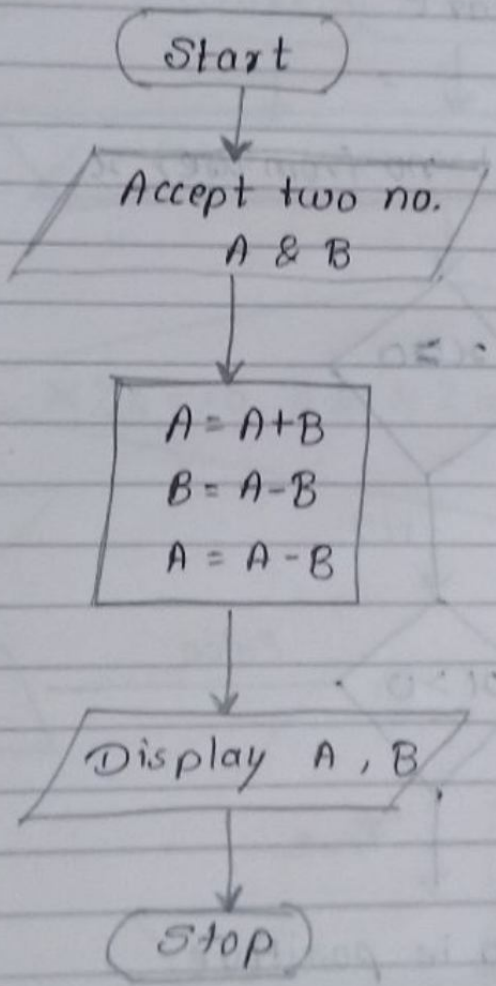

Start
accept no. n
N>0
NO
Yes
Error
Stop
Fact = 1
N==1
display Fact
Stop
Fact = Fact * n
n = n+1

Q] Swap two no. without using third variable

Algorithm

Step I - Delared required Variable
Step II - accept two no A & B from user
Step III A = A+B
B = A-B
A = A-B
Step IV : Display Value of A & B
Step V - Stop

Flowchart :

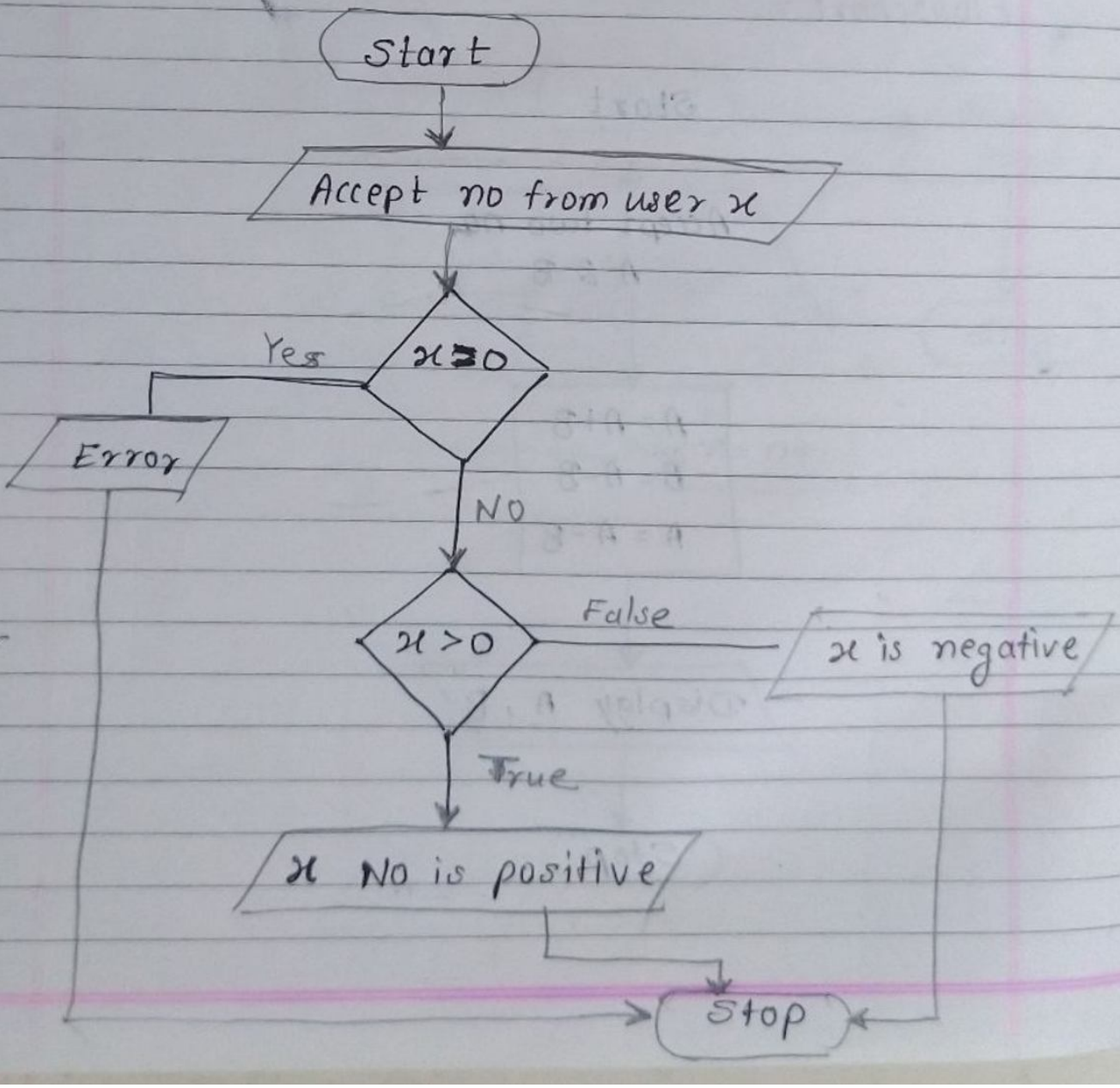


Q5] Check no is positive or negative

Algorithm

Step I = Declare required Variable
Step II = accept no from user x
Step III = if x > 0 then no is positive
else no is negtive
Step IV = Display result
step V = Stop

Flowchart -

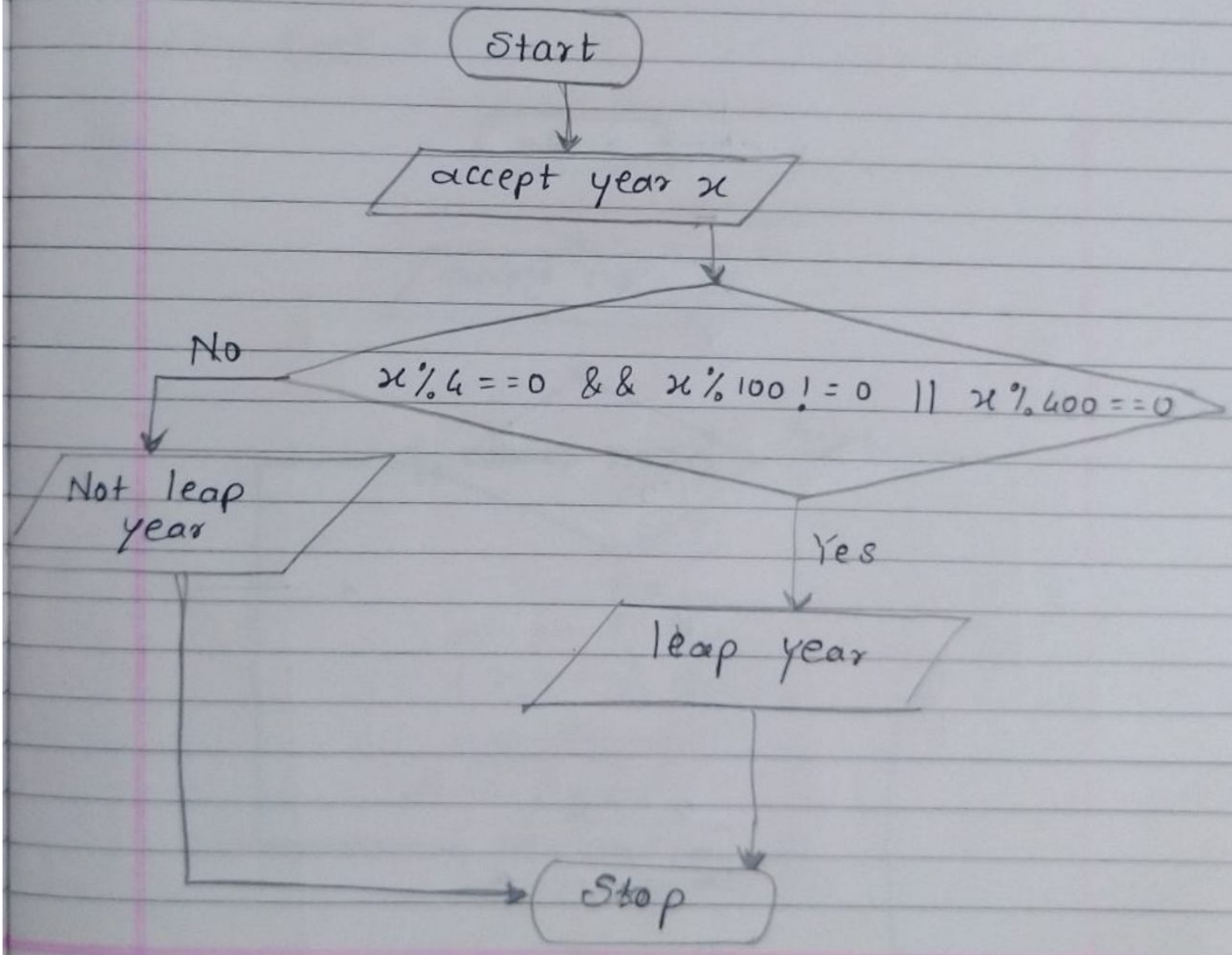


Q6] Find the no is leap year or not

Algorithm :

Step I : Declare the required Variable
Step II : accept the year from user
Step III : year % 4 == 0 && year % 100 != 0 || year % 400 == 0
step IV : Display the result
Step V : stop

Flowchart :

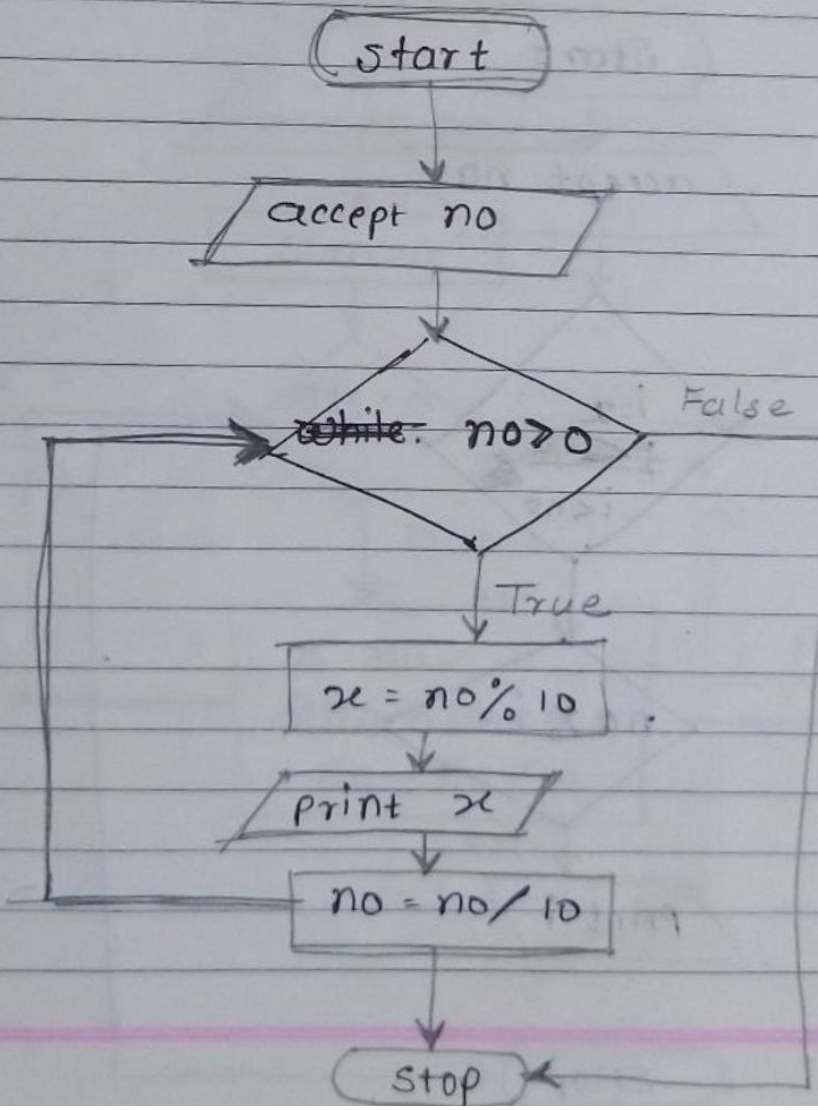


Q8] To print digits of given no.

Algorithm =

Step I = Declare the Variable required
Step II = accept no from user
Step III = while (no>0)
x = no%10
~~if yes then go~~ Display x
no = no/10
Step IV = stop

Flowchart =

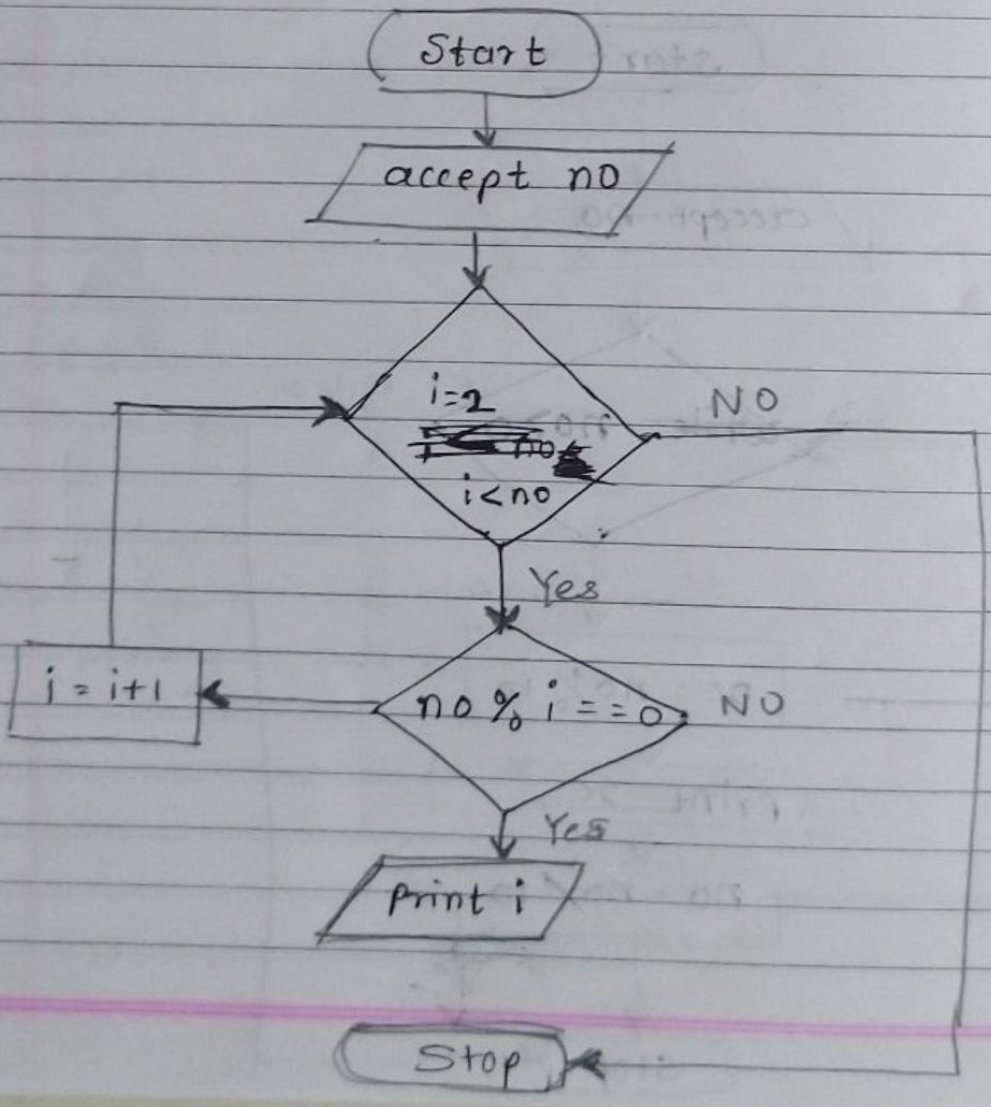


Q9] Print all the factors of given no.

Algorithm :

Step I = Declare required Variables
Step II = accept no from user
Step III = initialise counter as 2, increment counter upto no
Step IV = if no % i == 0 then print i or counter ~~else~~ increment & repeat
Step V = Stop

Flowchart



Q] (b) Sum of the digit of given no.

Algorithm :

Step I : Declare required Variable

Step II : accept no from user

Step III : if no > 0 then

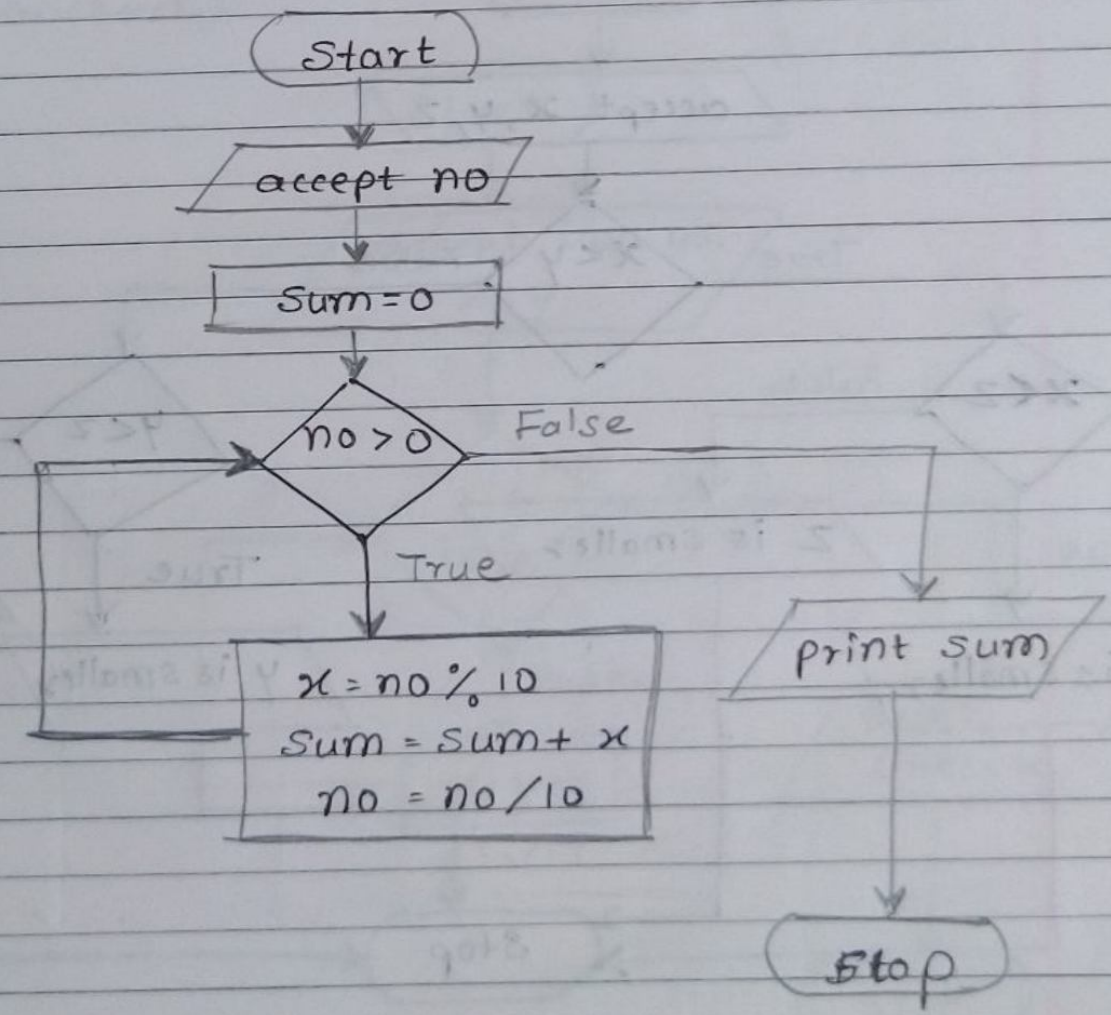
x = no % 10

sum = sum + x

no = no / 10

Step IV ~~else~~ stop

Flowchart

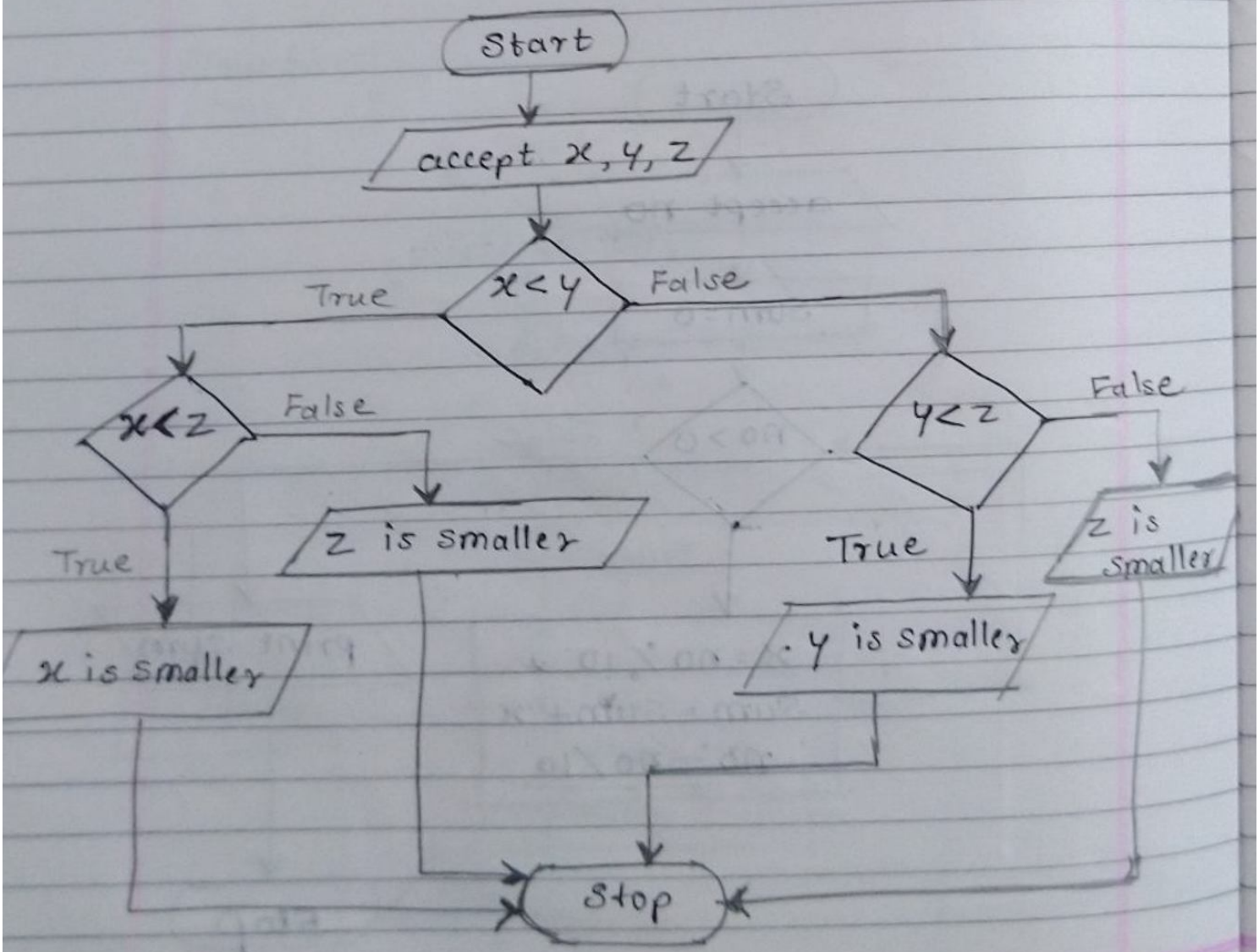


Q ii] Find the smallest of three no.

Algorithm

Step I = Declare required Variable
Step II = accept three no from user x, y, z
Step III = (x<y) ? ((x<z) ? x : z) : ((y<z) ? y : z)
Step IV = Display result
Step V = Stop

Flowchart =

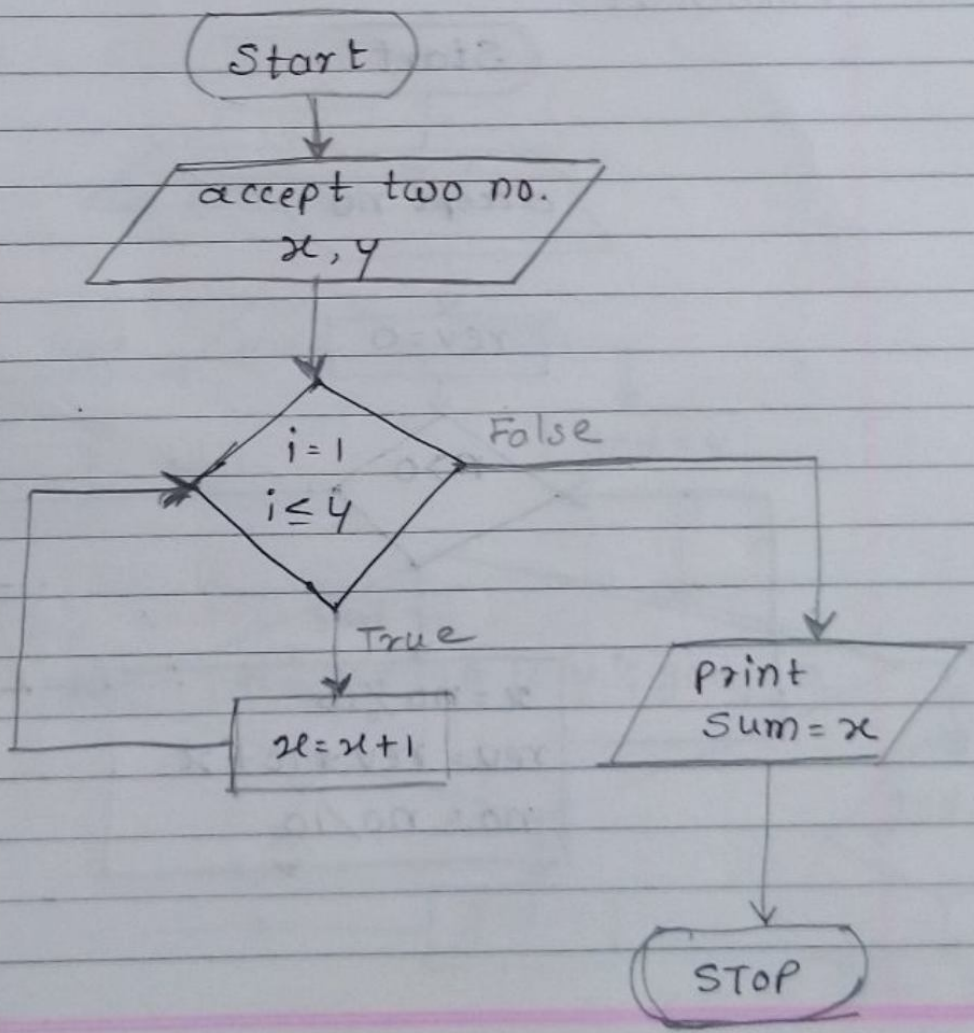


Q 12] Add two no without arithmatic operator

Algorithm

Step I = Declare required Variable
Step II = Accept two no x & y from user
step III = ~~while (num > 0)~~ initialise counter as 1
~~x~~ & increment counter upto y
~~I~~ncrement x ~~inside~~ upto condition not failed
Step IV = Display result
step V = Stop

Flowchart =

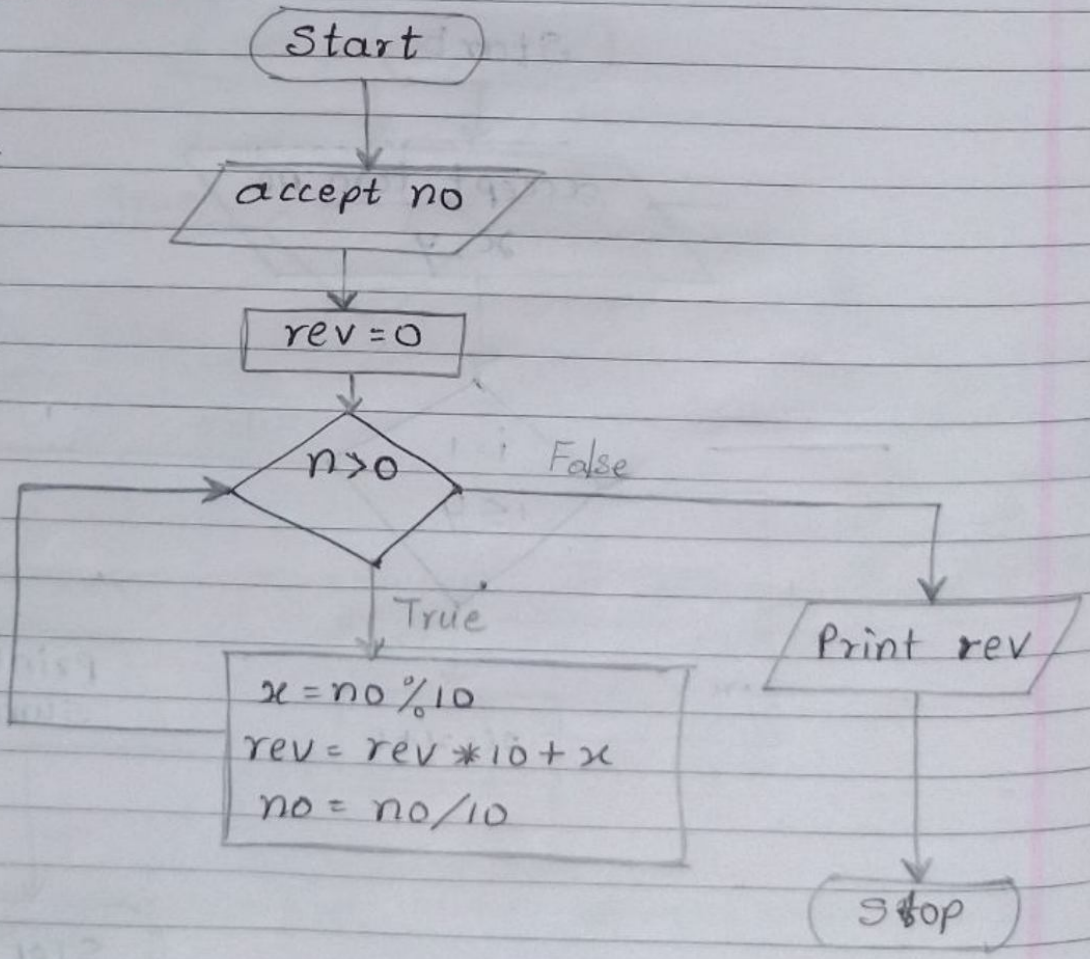


Q] 13 Reverse given no

Algorithm

Step I = Declare the required Variable
Step II = accept no from user
Step III = ~~if n>0 then~~ initialise rev as 0
Step IV = ~~re~~ if n>0 then x = no % 10
rev = rev * 10 + x
no = no / 10
Step V = Display rev
Step VI = Stop

Flowchart :

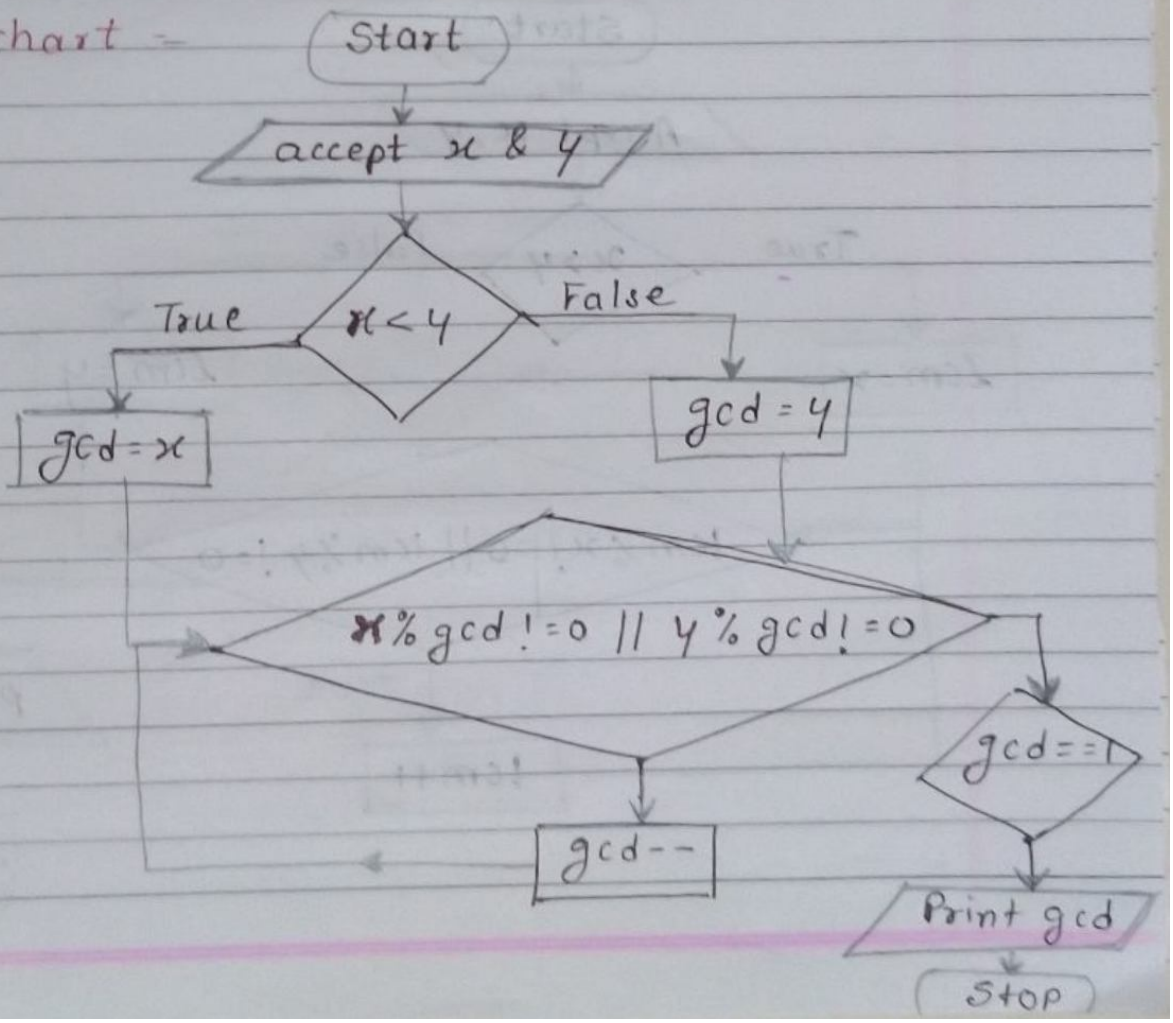


Q14] Find GCD of two no.

Algorithm

Step I = Declare the required Variables
Step II = accept to no. x & y from user
Step III = if x<y
gcd = x else gcd = y
Step IV = x % gcd != 0 || y % gcd != 0
gcd--
Step V = If gcd == 1 then gcd not exit print
else print gcd
Step VI = Stop

Flowchart =

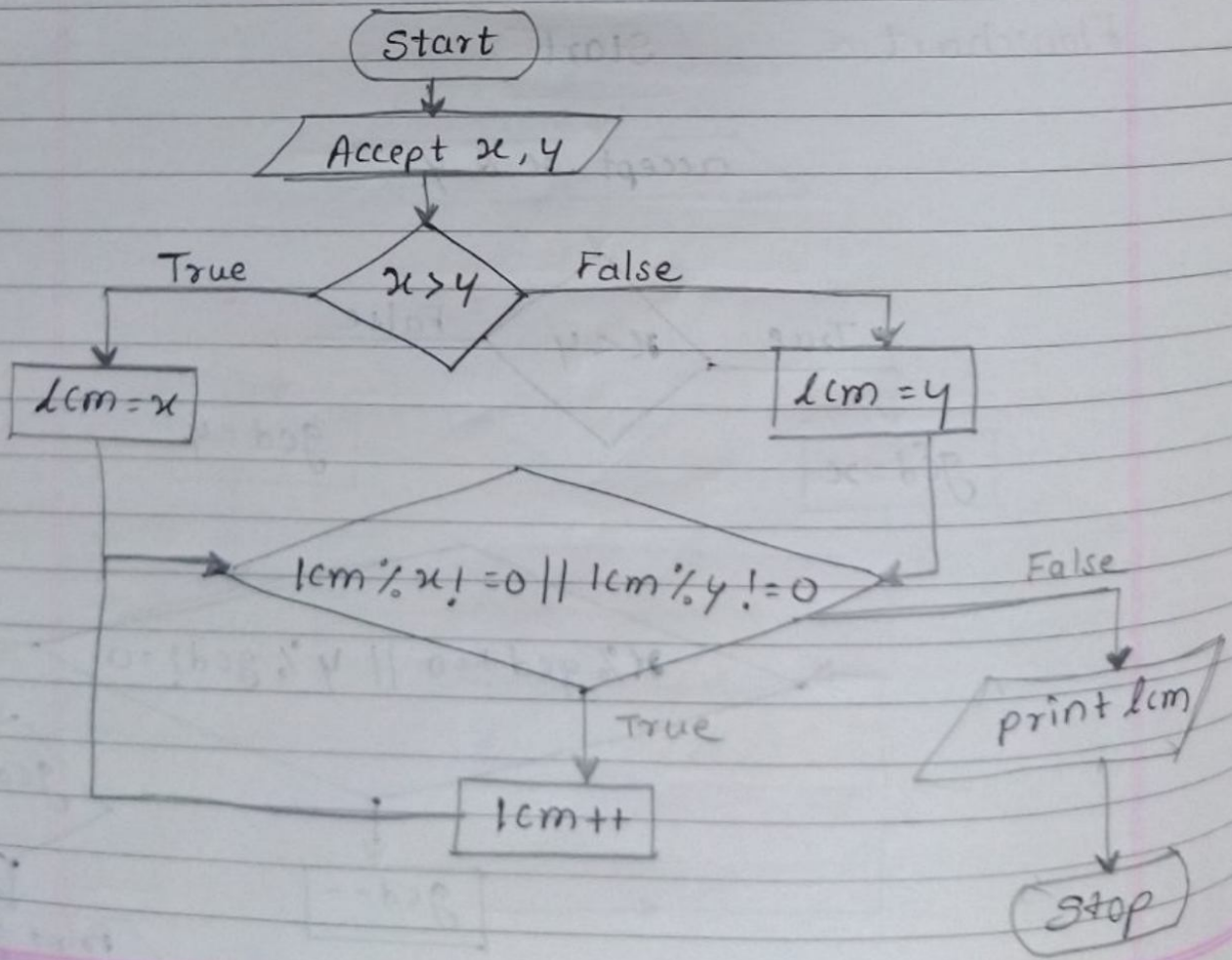


Q 15] LCM of Two No.

Algorithm

Step I = Declare required Variables
Step II = accept two no x & y from user
Step III = if x > y then
lcm = x else lcm = y
Step IV = using while loop lcm % x != 0 || lcm % y != 0
if yes then lcm increment
Step V = print lcm
Step VI = stop

Flowchart =



Q 17] given no is palindrome or not

Algorithm

Step I = Declared required variables

Step II = accept no. from user

Step III = Assign rev = 0 , temp = no.

Step IV = using while loop no > 0 then

~~no = num/~~ x = no % 10,

rev = rev * 10 + x

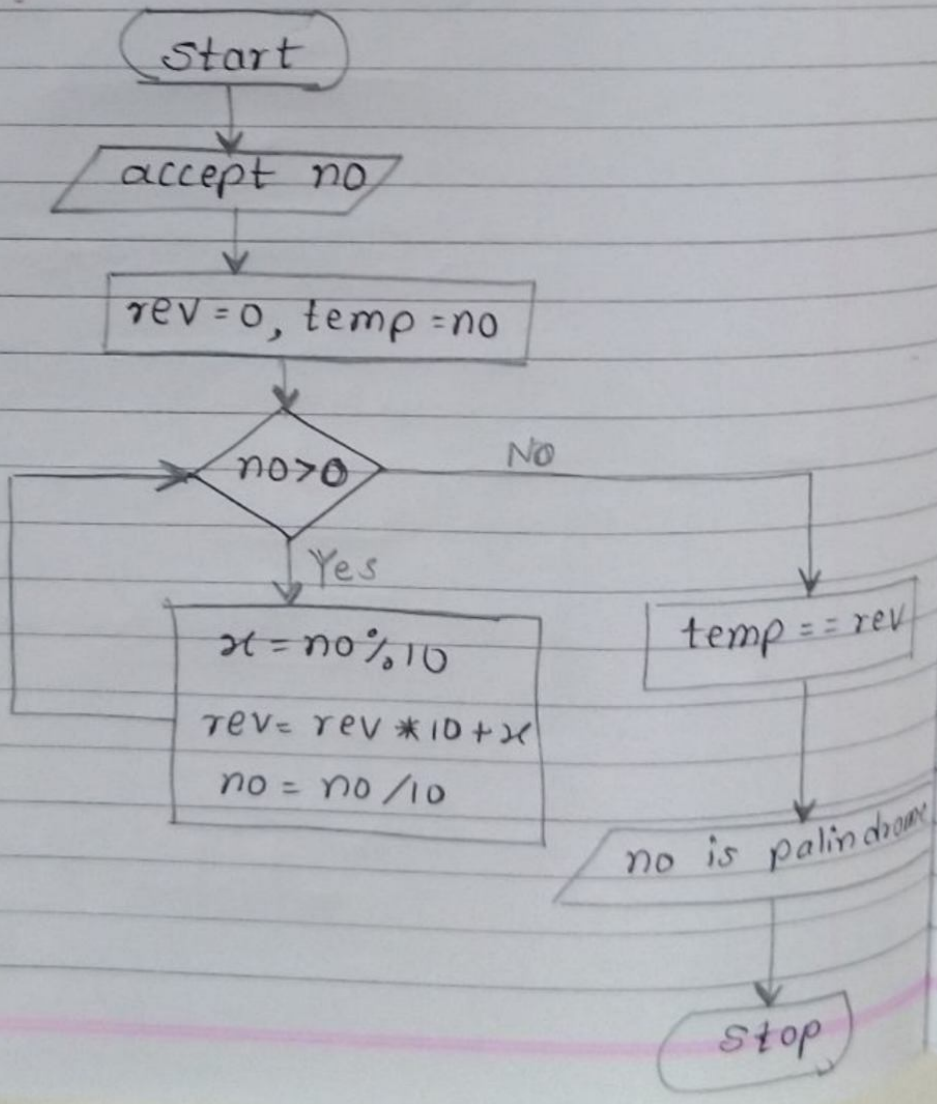
no = no / 10

Step V = temp == rev then no is palindrome

else no. is not palindrome

Step VI = Stop

Flowchart =



19] Print Even no. series

Algorithm

Step I = Declare require Variables

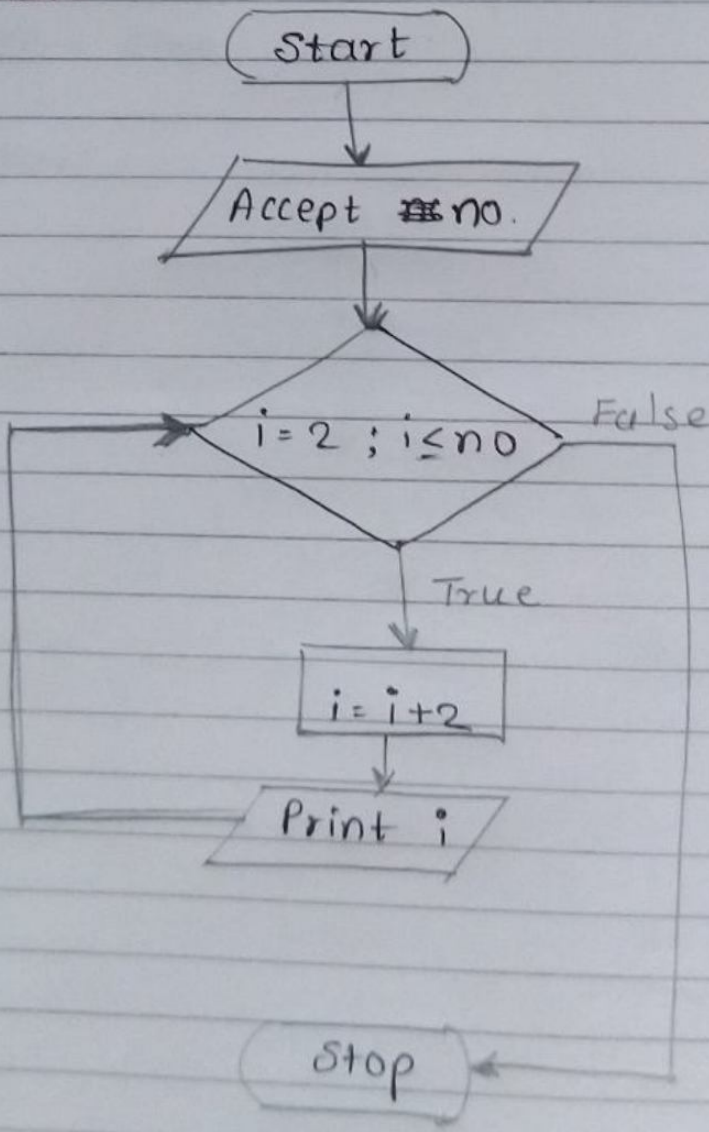
Step II = Accept no ~~from user~~ upto that no series will print ~~from user~~

Step III = initialise i = 2 ; i ≤ no. & increment i by 2

Step IV = print i

Step V = stop

Flowchart



~~Print i~~

~~St~~

20] Print odd no. series

Algorithm

Step I = Declare required Variables

Step II = Accept no upto which odd no. series will print

Step III = initialise i=1 ; i≤no & increment i by 2

Step IV = print i

Step V = Stop

Flowchart =