

Assignment No 1

1. Check if the given number is EVEN or ODD.

Algorithm:

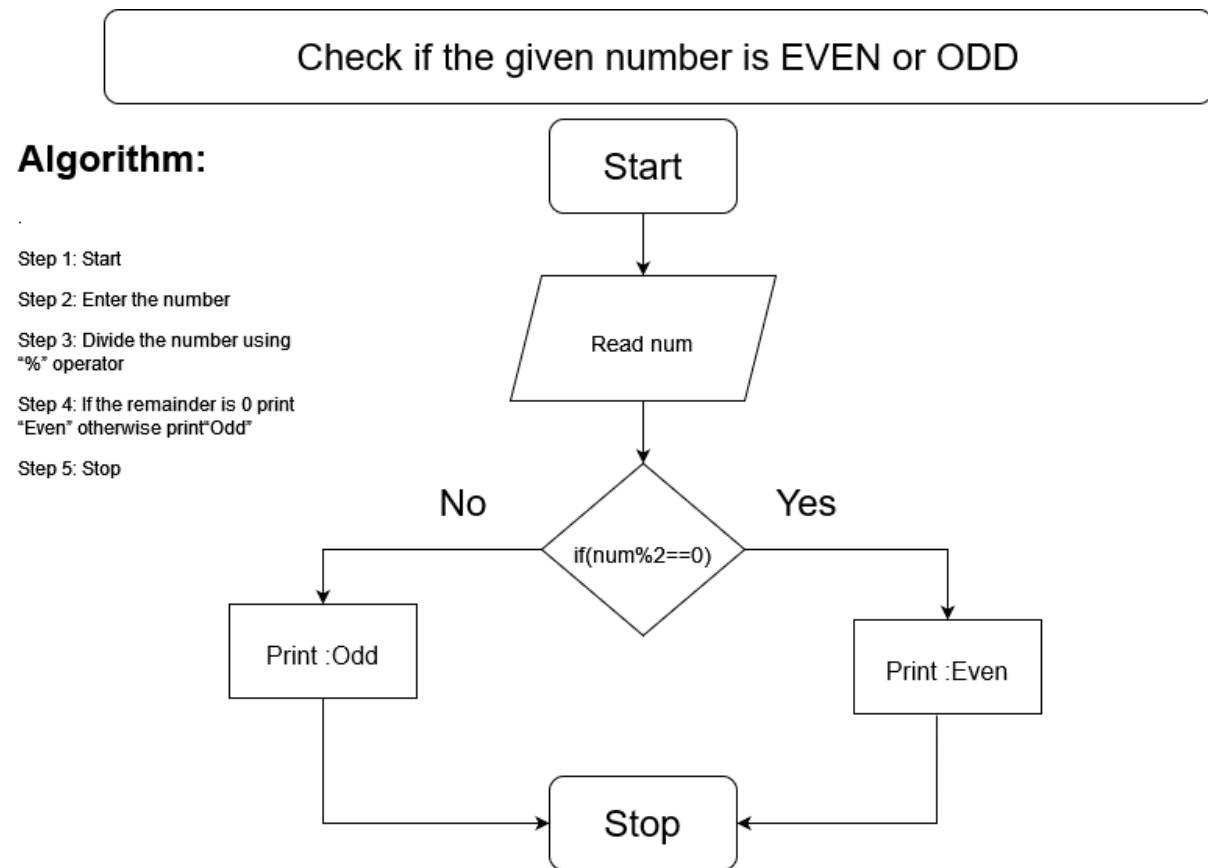
Step 1: Start

Step 2: Enter the number

Step 3: Divide the number using “%” operator

Step 4: If the remainder is 0 print “Even” otherwise print “Odd”

Step 5: Stop



2. Write a Java Program to find the Factorial of a given number.

Algorithm:

Step 1: Start

Step 2: read number

Step 3: Set fact=1, i=1

Step 4: Check condition $i \leq \text{number}$ if false go to Step 7

Step 5: $\text{fact} = \text{fact} * i$

Step 6: update $i = i + 1$ go to step 4

Step 7: Display fact

Step 8: Stop

Write a Java Program to find the Factorial of a given number.

Algorithm

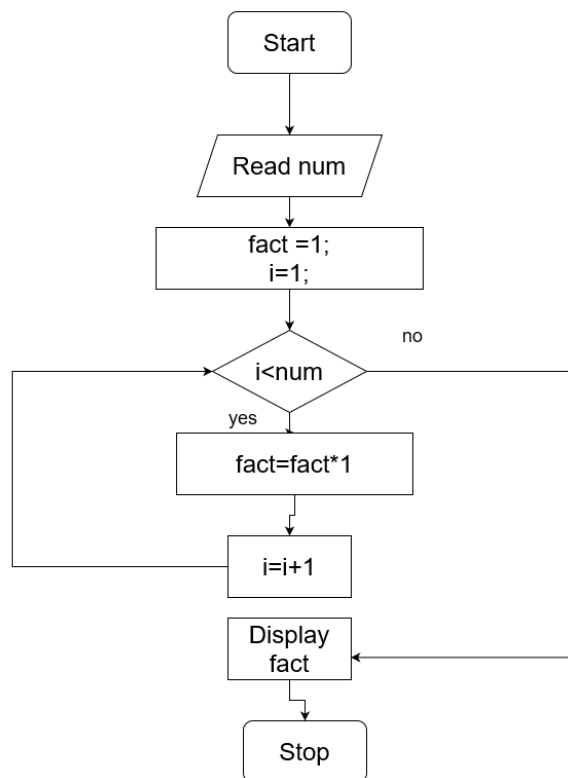
Step 1: Start

Step 2: read number

Step 3: Set $\text{fact} = 1$, $i = 1$

Step 4: Check condition $i \leq \text{number}$ if false go to Step 7

Step 5: $\text{fact} = \text{fact} * i$



3. Find the Factorial of a number using Recursion.

Step 1: Start

Step 2: Read number num

Step 3: Call factorial(num)

Step 4: If $\text{no} < 0$ then return -1

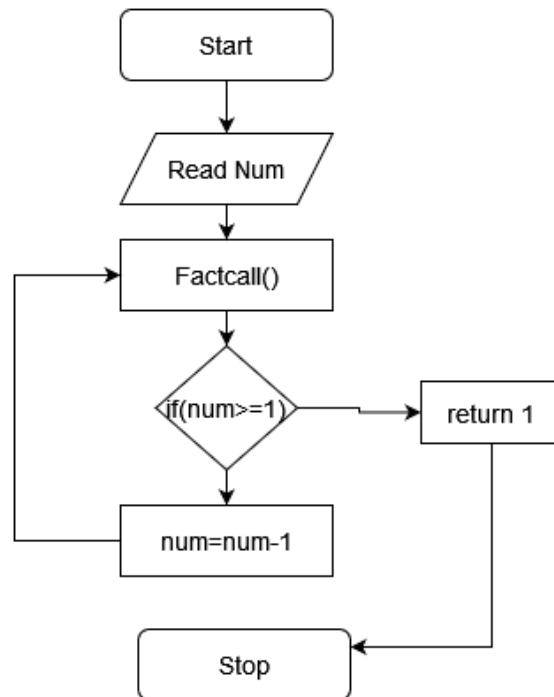
: Else if $\text{no} = 0$ then return 1

: Else

`return(no*factorial(no-1))`

Step 5: Print factorial fact

Find the Factorial of a number using Recursion.



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

Algorithm:

Step 1: START

Step 2: ENTER x, y

Step 3: PRINT x, y

Step 4: $x = x + y$

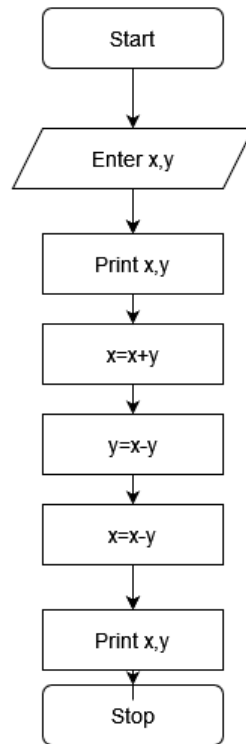
Step 5: $y = x - y$

Step 6: $x = x - y$

Step 7: PRINT x, y

Step 8: END

Swap two numbers without using the third variable approach.



5. How to check whether the given number is Positive or Negative in Java?

Step 1: Start

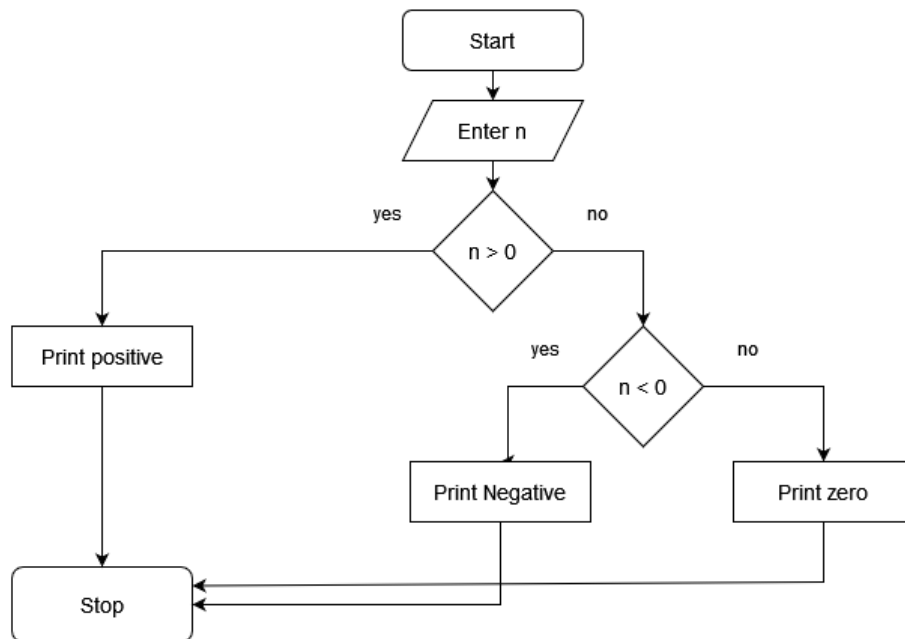
Step 2: Ask the user to Enter the number

Step 3: Input the number, Assign the number as "num"

Step 4: If the ($\text{num} > 0$) print Positive, ($\text{num} < 0$) print Negative ,Or ($\text{num} = 0$) print num is Zero

Step 5: Stop

. How to check whether the given number is Positive or Negative in Java?



6. Write a Java Program to find whether a given number is Leap year or NOT.

Algorithm:

Step 1: Start

Step 2: Ask the user to enter the year number, Assign the year as "year"

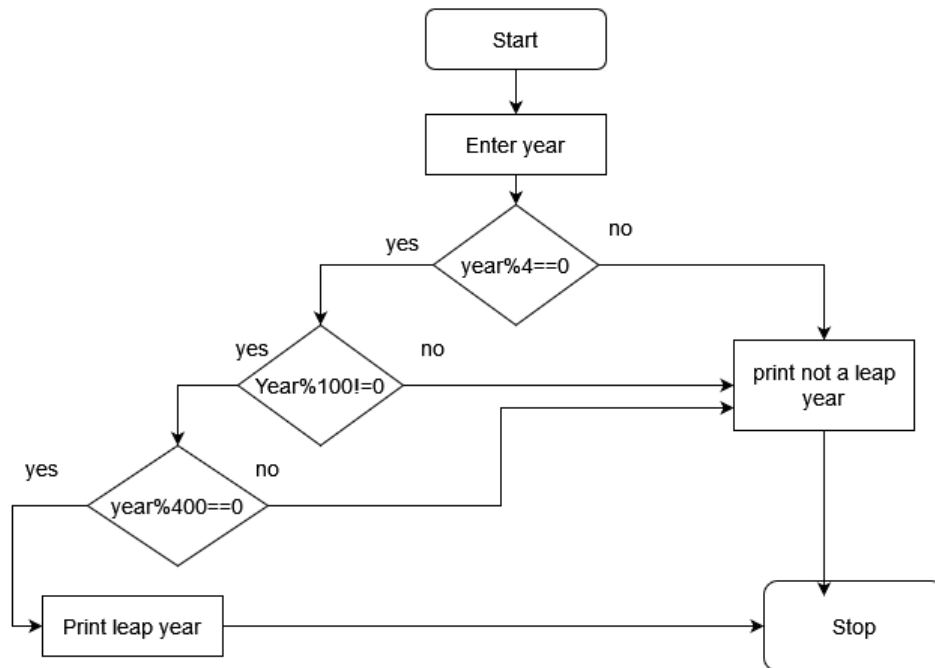
Step 3: Check if the year is divisible by 4 but not 100, DISPLAY "leap year"

Step 4: Check if the year is divisible by 400, DISPLAY "leap year"

Step 5: Otherwise, DISPLAY "not leap year"

Step: Stop

Write a Java Program to find whether a given number is Leap year or NOT.



7. Write a Java Program to Print 1 To 10 Without Using Loop.

Algorithm:

Step 1: Start

Step 2: we initialize the the starting value as 1 and assign it to num

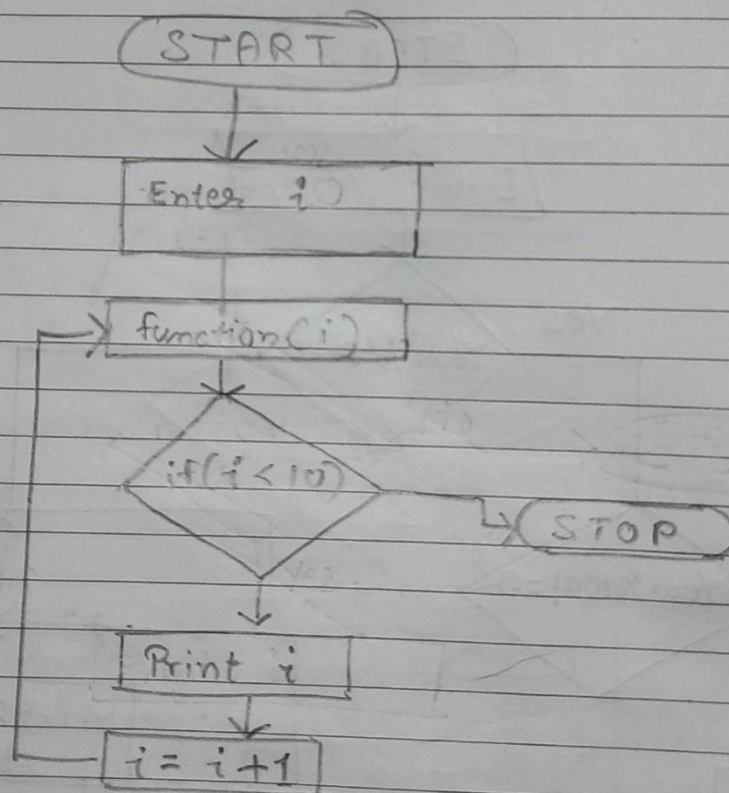
Step 3: we use an if condition stating if(num< 10) ,print the value of num

Step 4: then we increment the value by 1

Step 5 : we do this till the value num reaches 10

Step: Stop

7) Print 1 to 10 without loop



8. Write a Java Program to print the digits of a Given Number.

Step 1: Start

Step 2: Enter the number

Step 3: Set the counter to 0

Step 4: Divide the num using the % operator and increment the counter

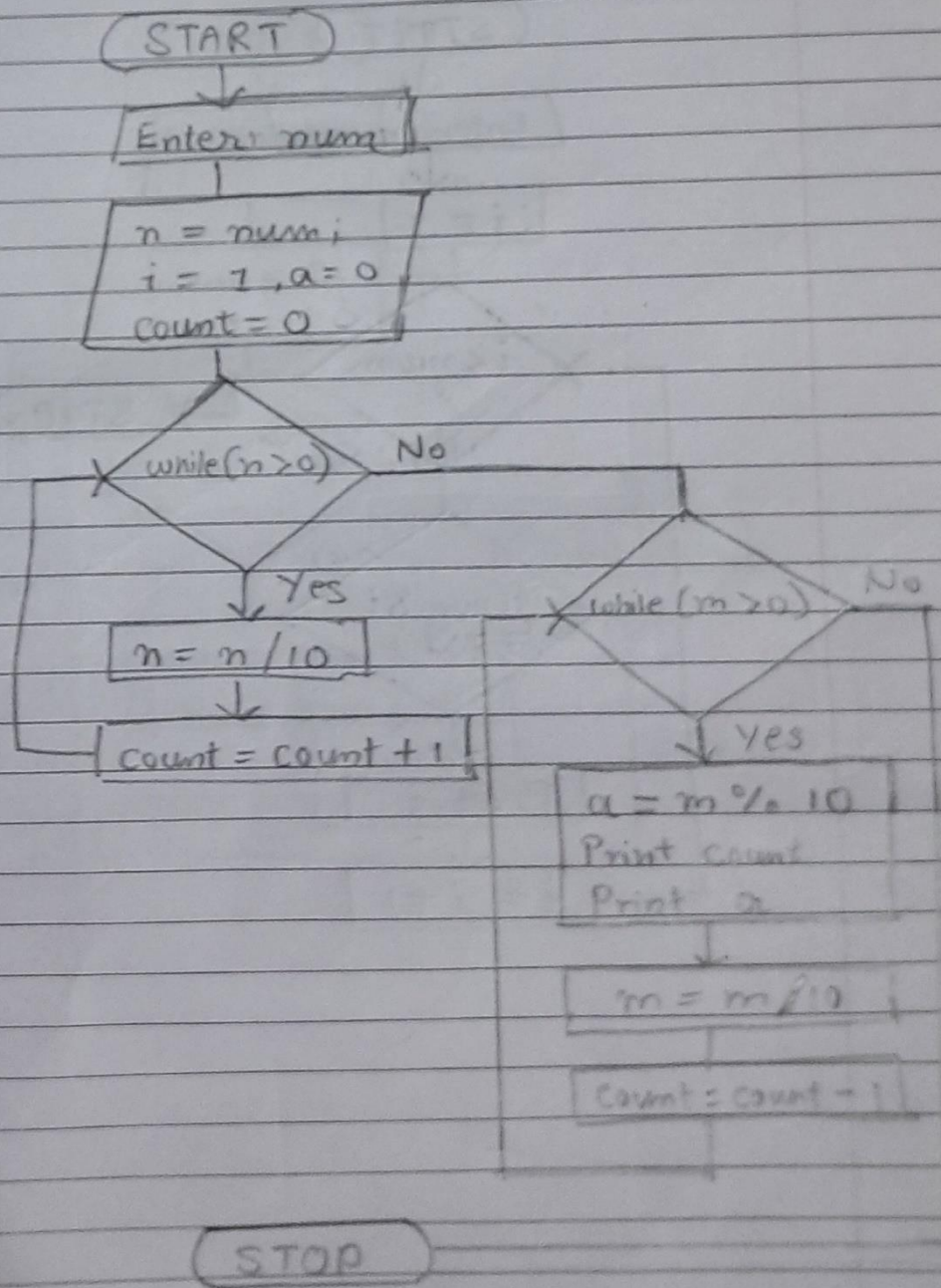
Then divide the number by 10

Print the remainder value along with counter value (estimating the position value of number)

Step 5: Repeat this process till the last number is done

Step 6: Stop

8) Print digits of given Number.



9. Write a Java Program to print all the Factors of the Given number.

Algorithm:

Step 1: Start

Step 2: Enter the number and set it as num

Step 3: set a value "a"=1

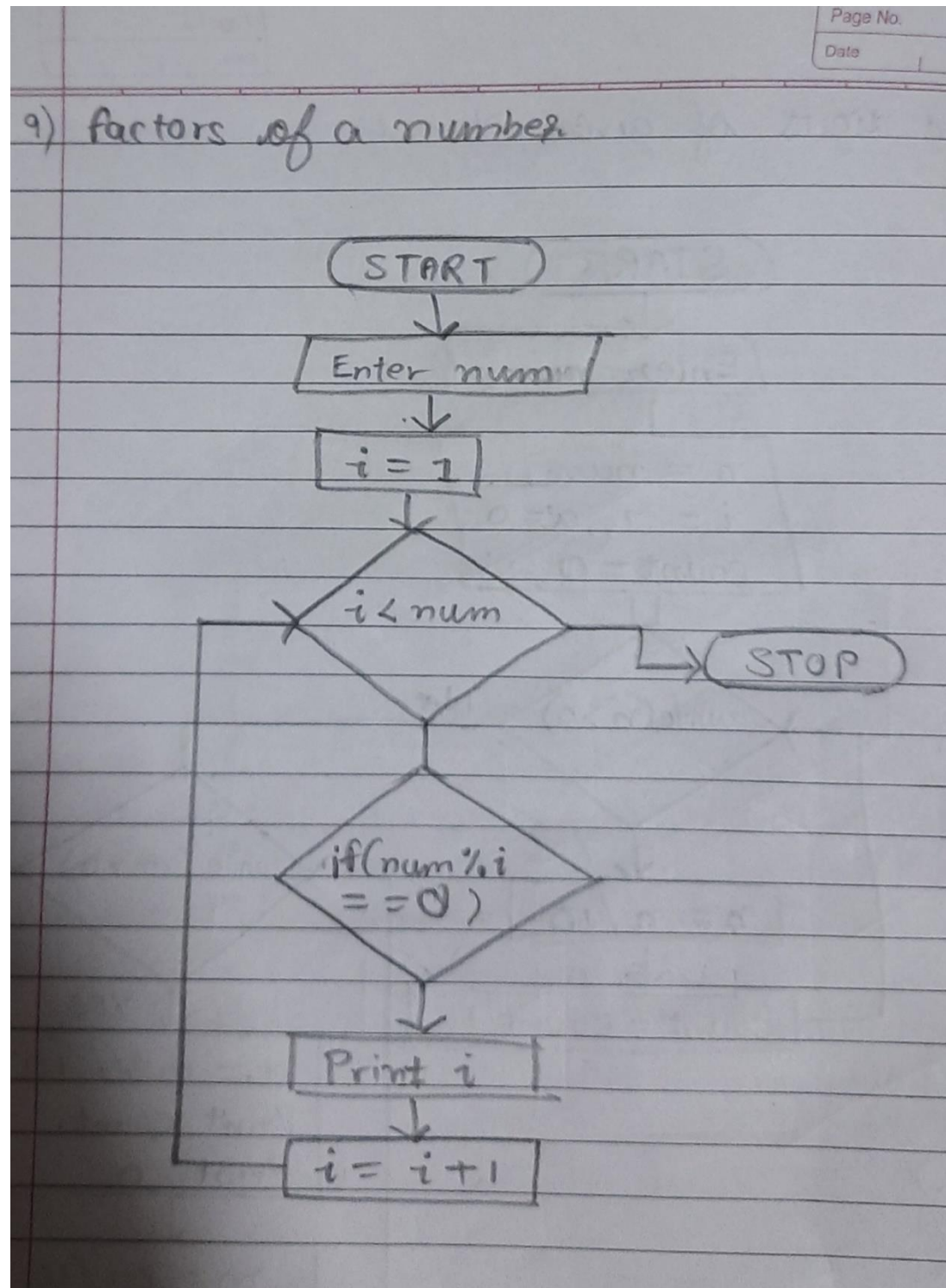
Step 4: Divide num by a using % operator

Step 5: if the remainder is 0 then print the value of a

Step 5: Increment the value of a by 1

Step 6: Repeat this process till value of a is equal to num

Step 7: Stop



10. Write a Java Program to find the sum of the digits of a given number.

Algorithm:

Step 1: Read or initialize an integer N.

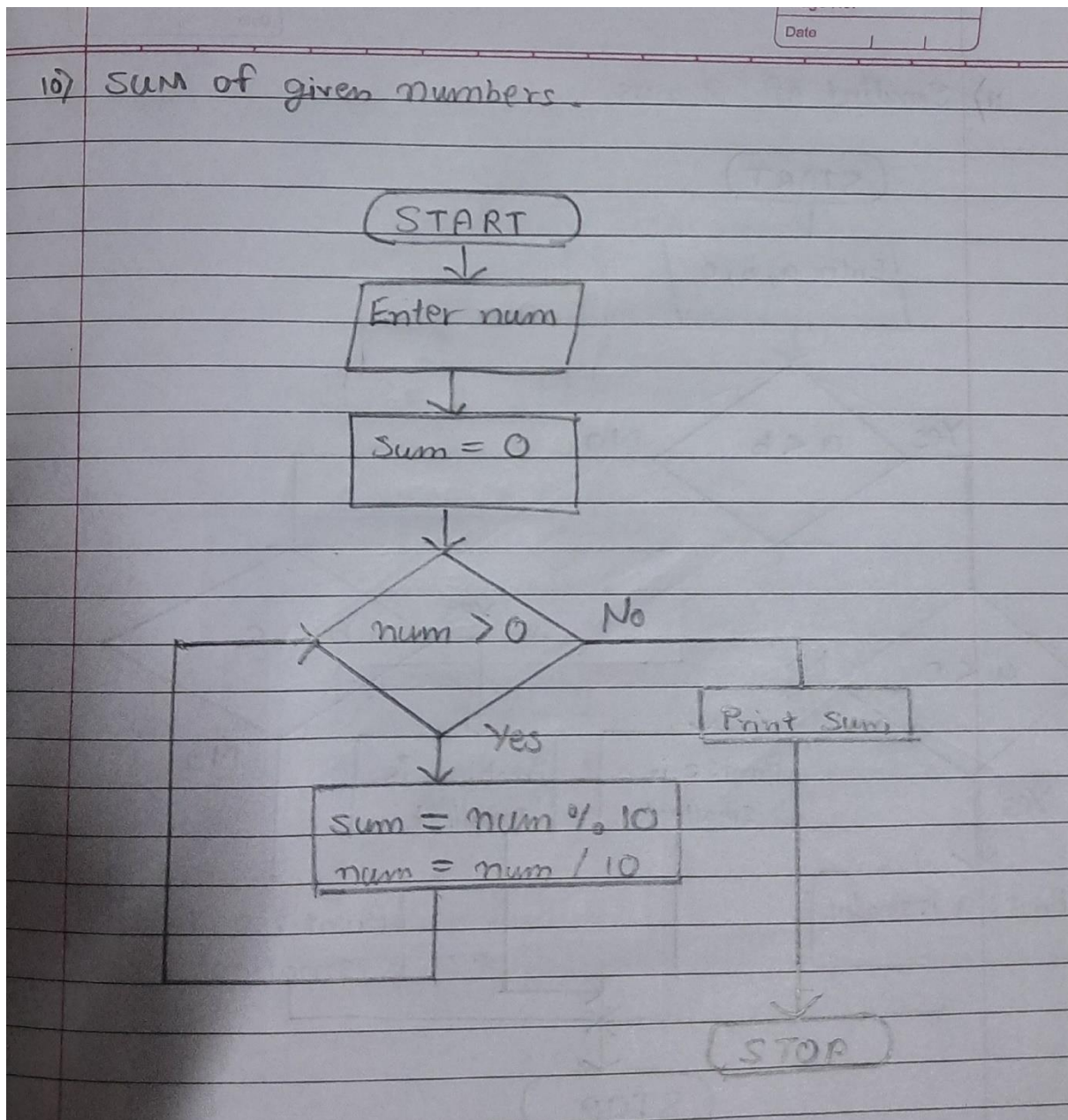
Step 2: Declare a variable (sum) to store the sum of numbers and initialize it to 0.

Step 3: Find the remainder by using the modulo (%) operator. It gives the last digit of the number (N)

Step 4: Add the last digit to the variable sum.

Step 5: Divide the number (N) by 10. It removes the last digit of the number.

Step 6: Repeat the above steps (3 to 5) until the number (N) becomes 0



11. Write a Java Program to find the smallest of 3 numbers (a,b,c)

Algorithm:

Step 1: Start

Step 2: Enter the three numbers

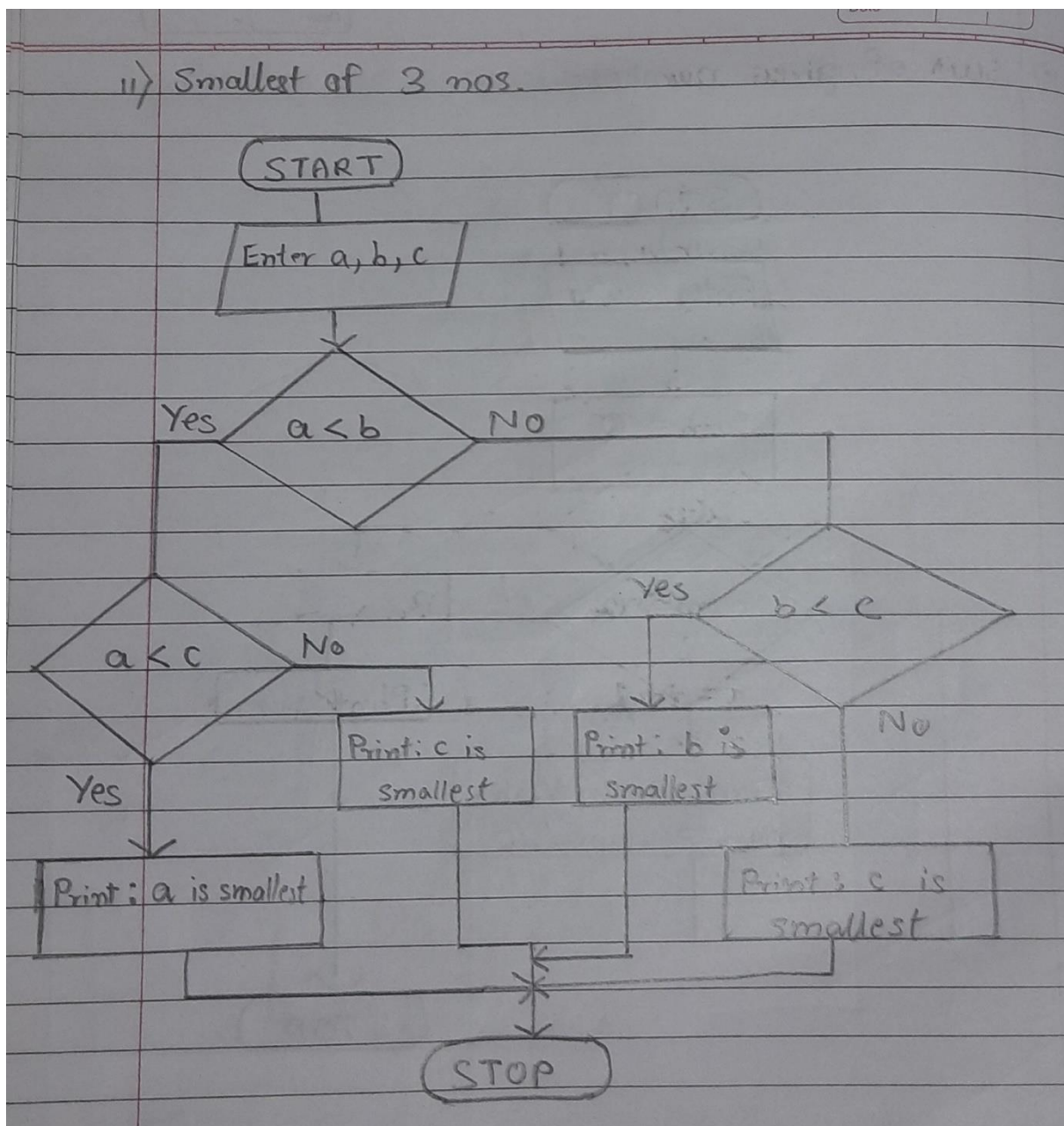
Step 3: Assign the numbers a,b,c

Step 4: If $(a < b) \ \&\& \ (a < c)$, Print "a is Smallest"

 If $(b < a) \ \&\& \ (b < c)$, Print "b is Smallest"

 If $(c < a) \ \&\& \ (c < b)$, Print "c is Smallest"

Step 5: Stop



12. How to add two numbers without using the arithmetic operators in Java?

Algorithm:

Step 1: Start

Step 2: Enter the two numbers "a" and "b"

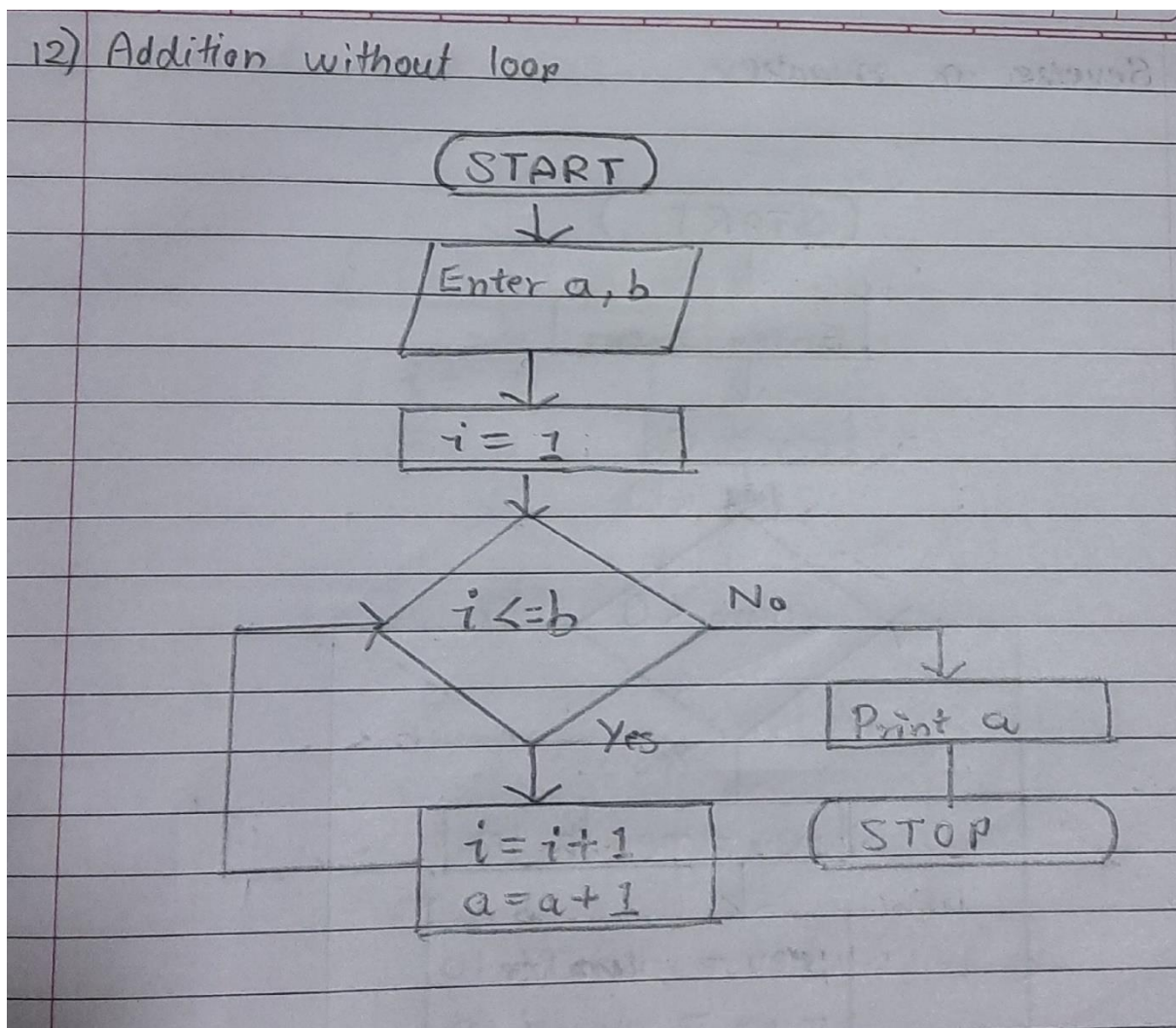
Step 3: set a variable value (i) to 1 and increment by one

Increment the value of a by one accordingly

Step 4: Repeat this till the value of (i) is equal to b

Step 5: Print the new final incremented value of a

Step 6: Stop



13. Write a java program to Reverse a given number.

Algorithm:

Step 1: Start

Step 2: Enter the number

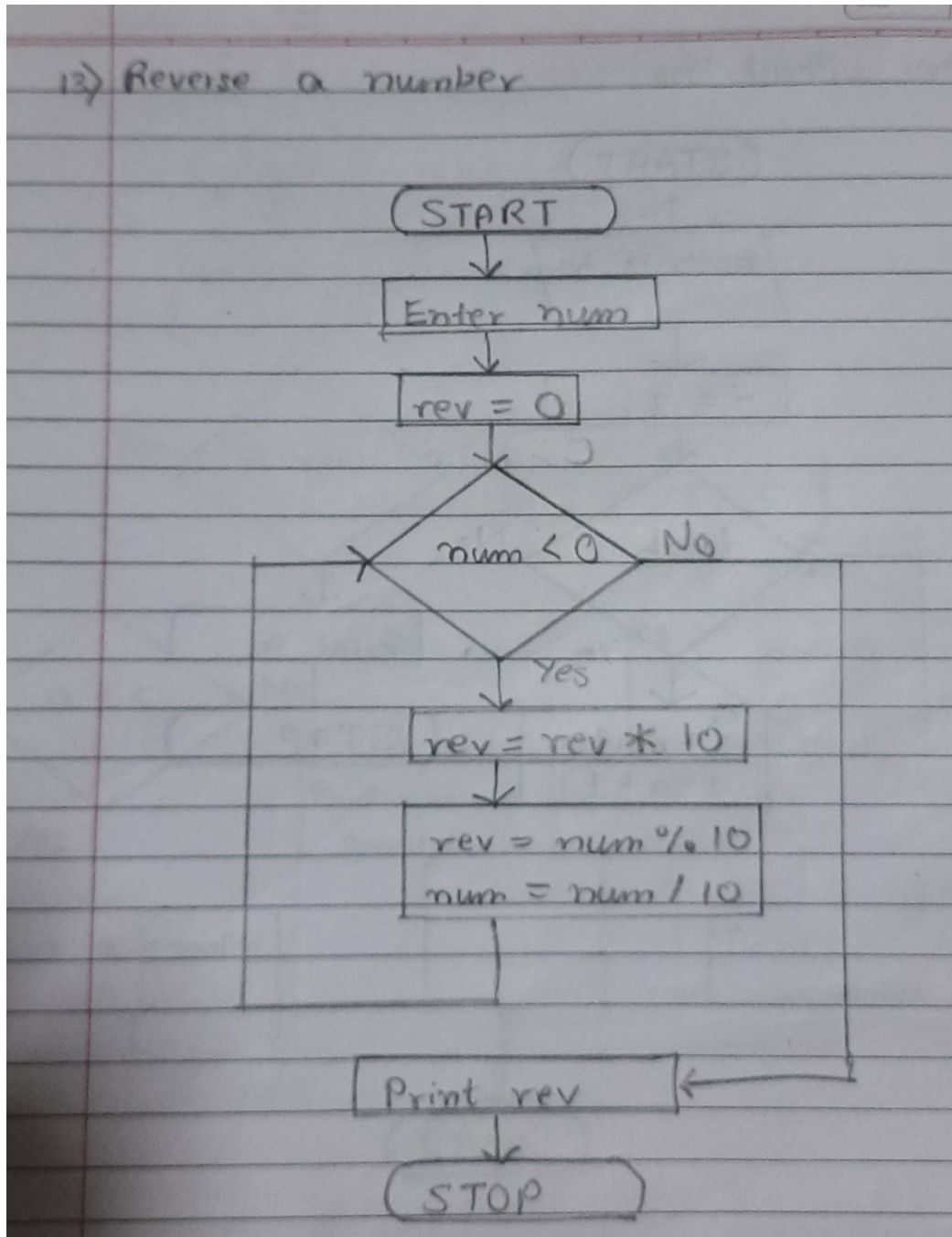
Step 3: Find the remainder of the given number by using the modulo (%) operator.

Step 4: Multiply the variable reverse by 10 and add the remainder into it.

Step 5: Divide the number by 10.

Step 6: Repeat the above Steps until the number becomes 0.

Step 7: Stop



14. Write a Java Program to find the GCD of two given numbers.

Algorithm:

Step 1: Start

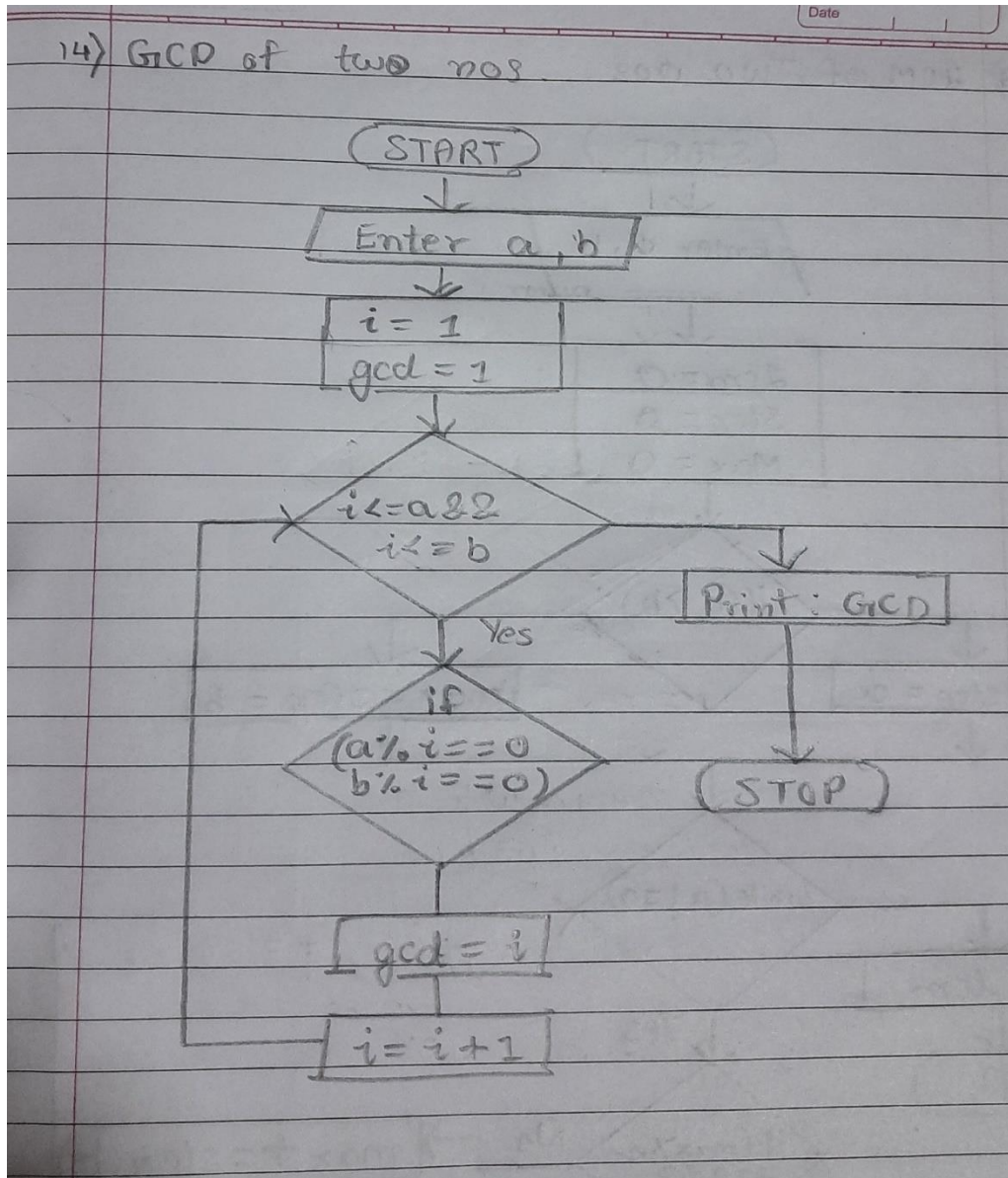
Step 2: Declare two variables, say x and y.

Step 3: Run a loop for x and y from 1 to max of x and y.

Step 4: Check that the number divides both (x and y) numbers completely or not. If divides completely store it in a variable.

Step 5: Divide the stored number

Step 6: Stop



15. Write a java program to LCM of TWO given numbers.

Algorithm:

Step 1: Start

Step 2: Initialize A and B with positive integers.

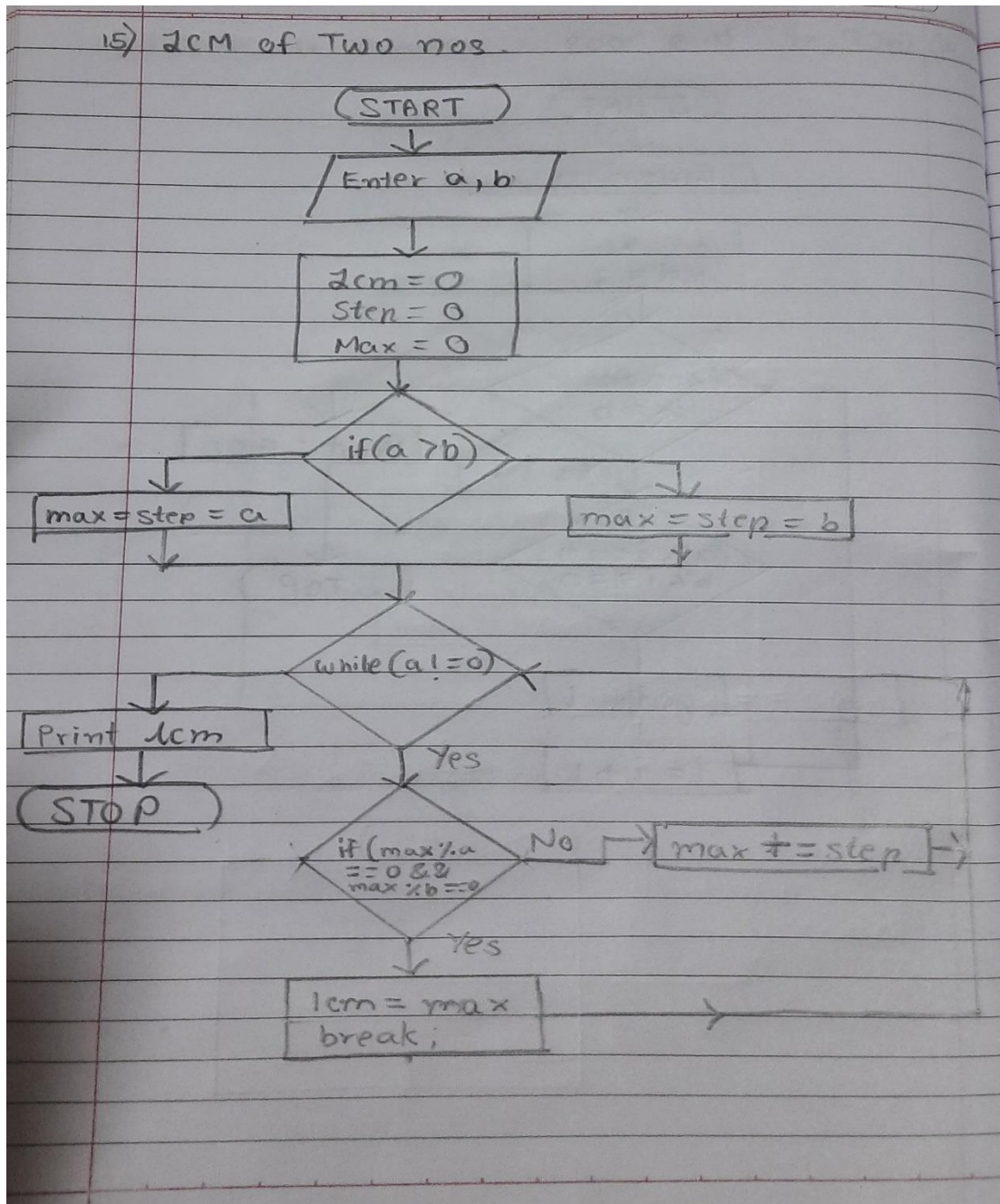
Step 3: Store maximum of A & B to the max.

Step 4: Check if max is divisible by A and B.

Step 5: If divisible, Display max as LCM.

Step 6: If not divisible then step increase max, go to step 3.

Step 7: Stop



16. Write a java program to LCM of TWO given numbers using the Prime Factors method.

Algorithm:

Step 1: Start

Step 2: Enter n1, n2

Step 3: div=2, lcm=1

Step 4: while (div<=n1 && div<=n2) goto step 5

Step 5: if(n1%2==0 && n2%2==0) goto step6 else got step 7

Step 6: n1=n1/div, n2=n2/div,lcm=lcm*div then goto step4

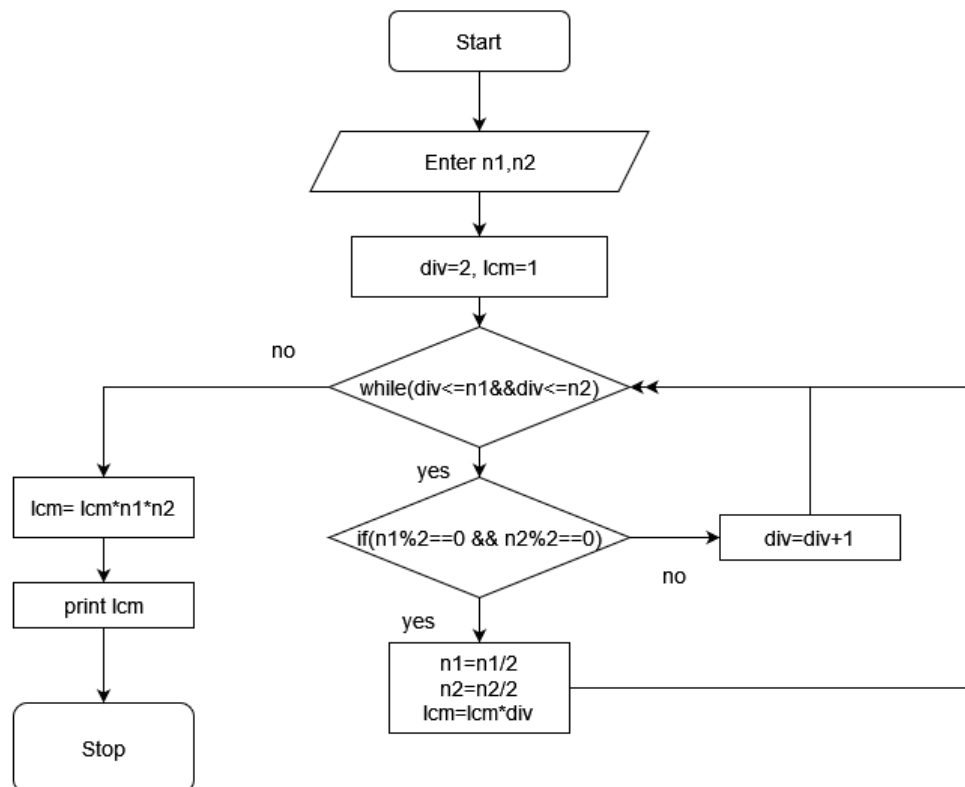
Step 7: div=div+1 goto step 4

Step 8: lcm=lcm*n1*n2

Step 9: print lcm

Step 10: Stop

Write a java program to LCM of TWO given numbers using the Prime Factors method.



17. Check whether the Given Number is a Palindrome or NOT.

Algorithm:

Step 1: Start

Step 2: Enter the number, Assign the value to (num) variable

Step 3: using for loop, find the remainder of the given number by using the modulo (%) operator.

Step 4: Multiply the variable reverse by 10 and add the remainder into it. Assign that value to variable "rev"

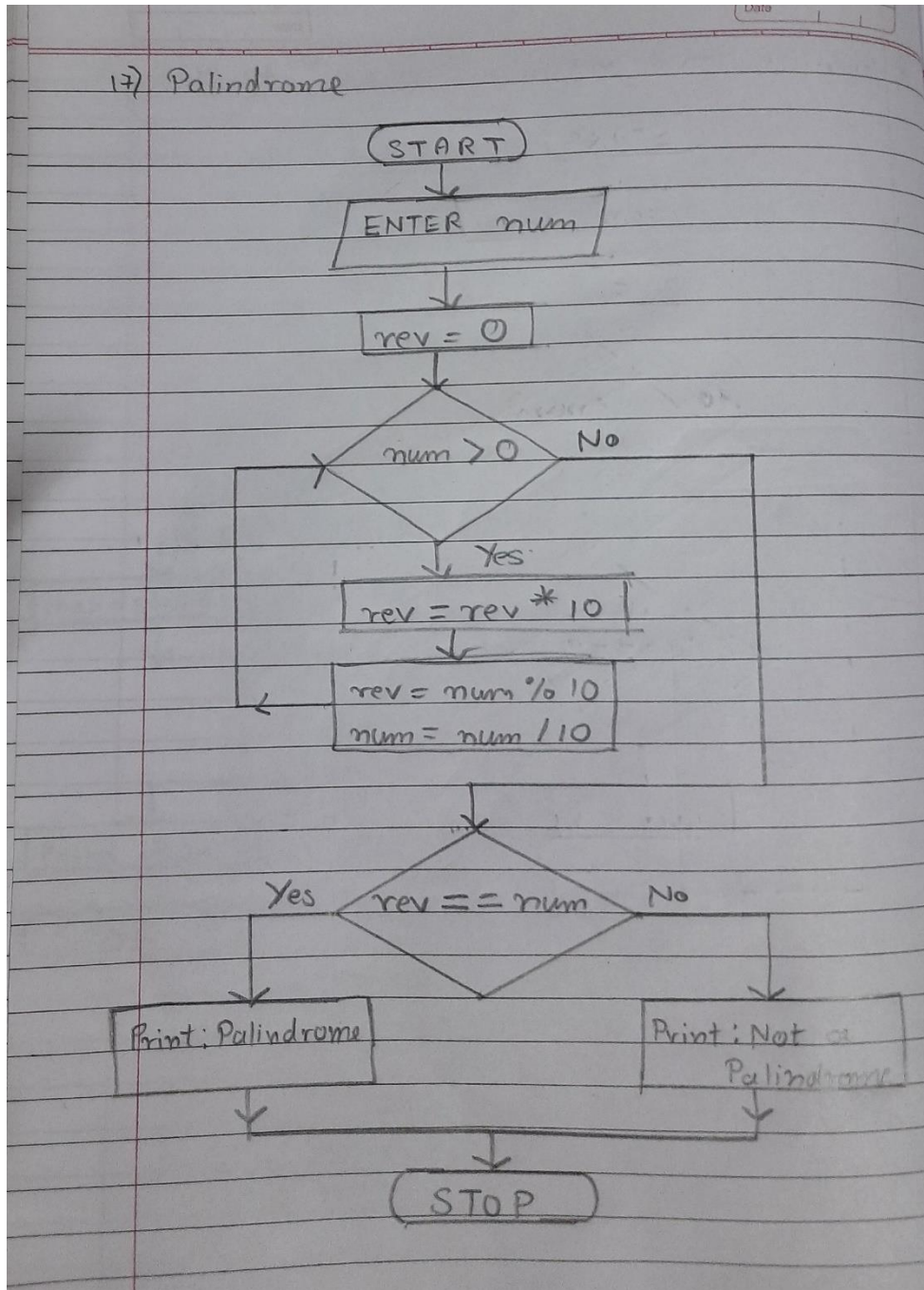
Step 5: Divide the number by 10.

Step 6: Repeat the above Steps until the number becomes 0.

Step 7: if the (rev) value is equal to the (num) value, print "Palindrome"

Step 8: Otherwise print "Not a Palindrome"

Step 9: Stop



18. Write a Java Program to print all the Prime Factors of the Given Number

Algorithm:

Step 1: Start

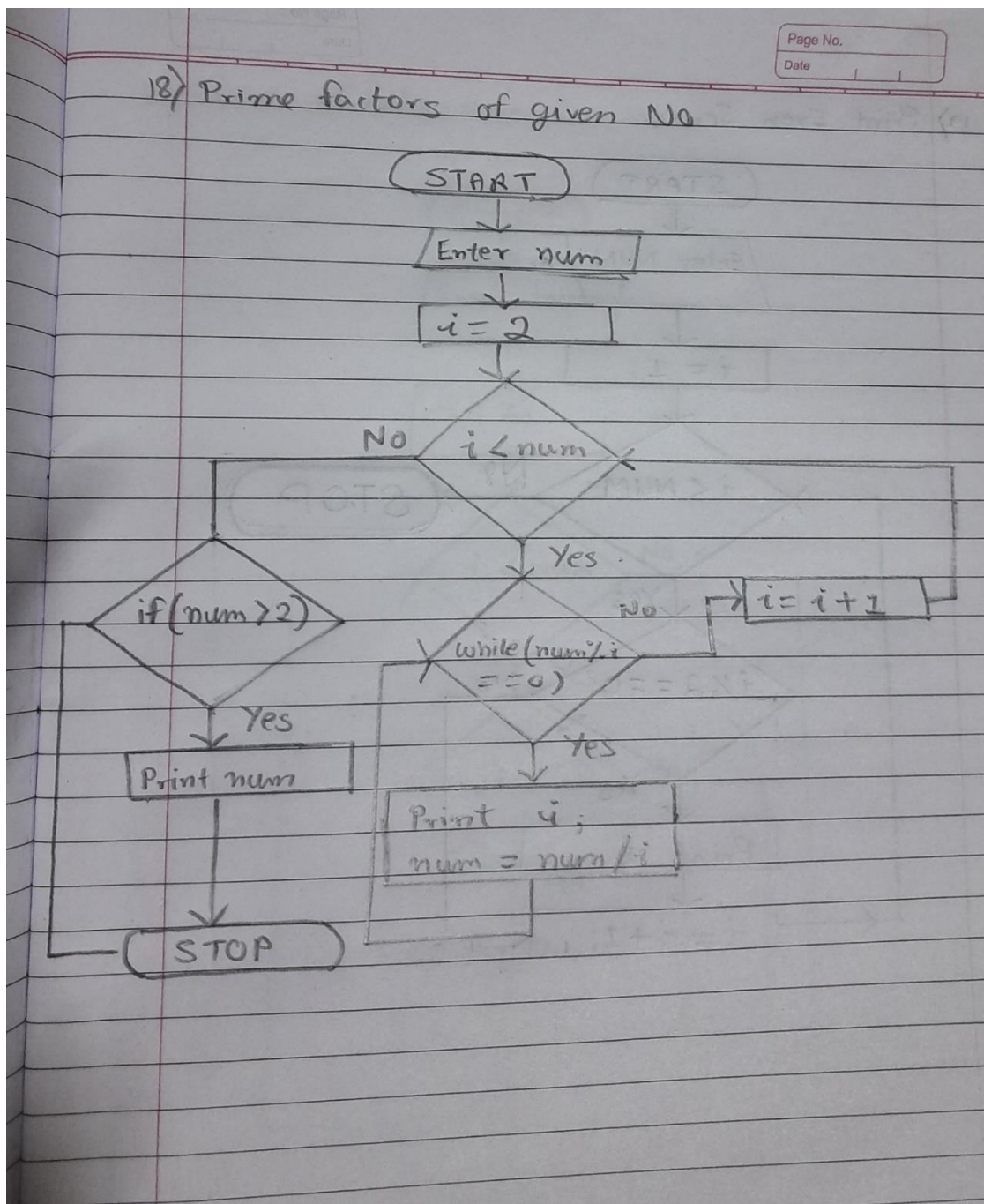
Step 2: Enter the number

Step 3: Create a for loop with the initial value (i) set to 2 and increment it till the value of (i)<number

Step 4: Inside the for loop, Divide the number by value of (i) and print value of (i), Repeat this till $\text{number} \% (i) == 0$ using while loop

Step 5: If the remainder number is greater than 2, print that value as well

Step 6: Stop



19. To print the following series EVEN number Series 2 4 6 8 10 12 14 16

Algorithm:

Step 1: Start

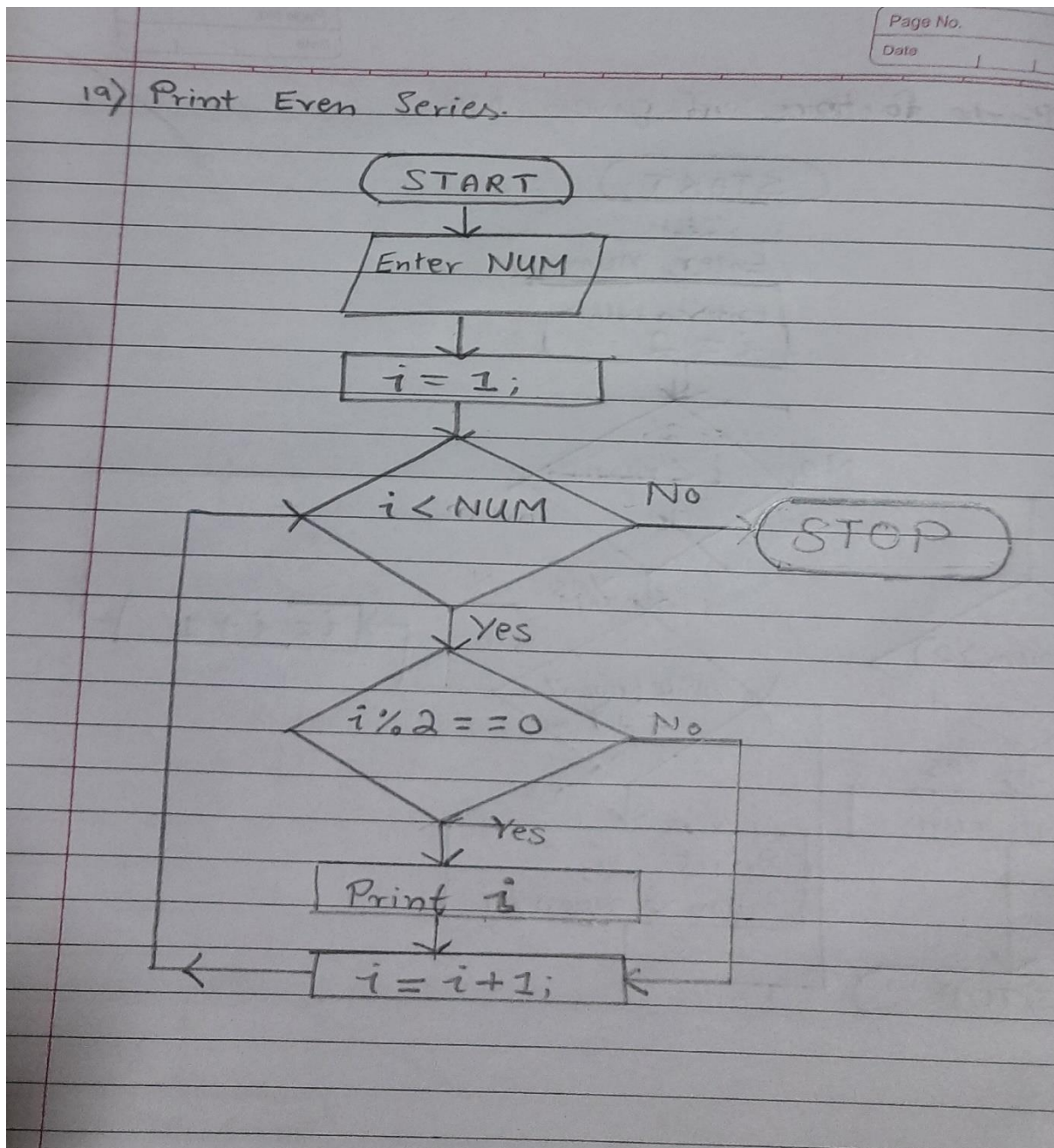
Step 2: Ask the user to Enter the last value, Assign the value to "n"

Step 3: Create a for loop and keep the starting value (i) as 1 and let it increment till the value "n"

Step 4: Divide each value of (i) by 2 using (%) Operator

Step 5: if the remainder is 0 then Print the value of (i)

Step 6: Stop



20. To print the following series ODD number Series 1 3 5 7 9 11 13...

Algorithm:

Step 1: Start

Step 2: Ask the user to Enter the last value, Assign the value to "n"

Step 3: Create a for loop and keep the starting value (i) as 1 and let it increment till the value "n"

Step 4: Divide each value of (i) by 2 using (%) Operator

Step 5: if the remainder is not 0 then Print the value of (i)

Step 6: Stop

