ASSIGNMENT NO.1

Write Algorithm and Flowchart for the following programs.

Q1] Check if the given number is EVEN or ODD.

Algorithm:

Step1 – START.

Step2 – PRINT “Enter a Number:”

Step3 – Read number.

Step4 – IF number % 2 == 0

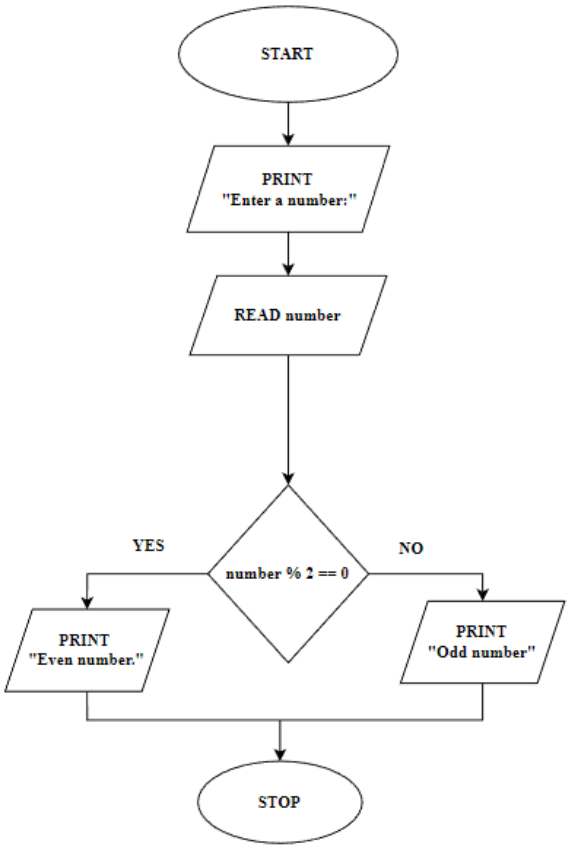
PRINT “Even Number.”

ELSE

PRINT “Odd Number.”

Step 5 – STOP

Flowchart:



Q2] Write a Java Program to find the Factorial of a given number.

Algorithm:

Step1 – START.

Step2 – DECLARE result = 1, number.

Step3 – READ number.

Step4 – DECLARE i = number

Step5 – REPEAT until i>=1

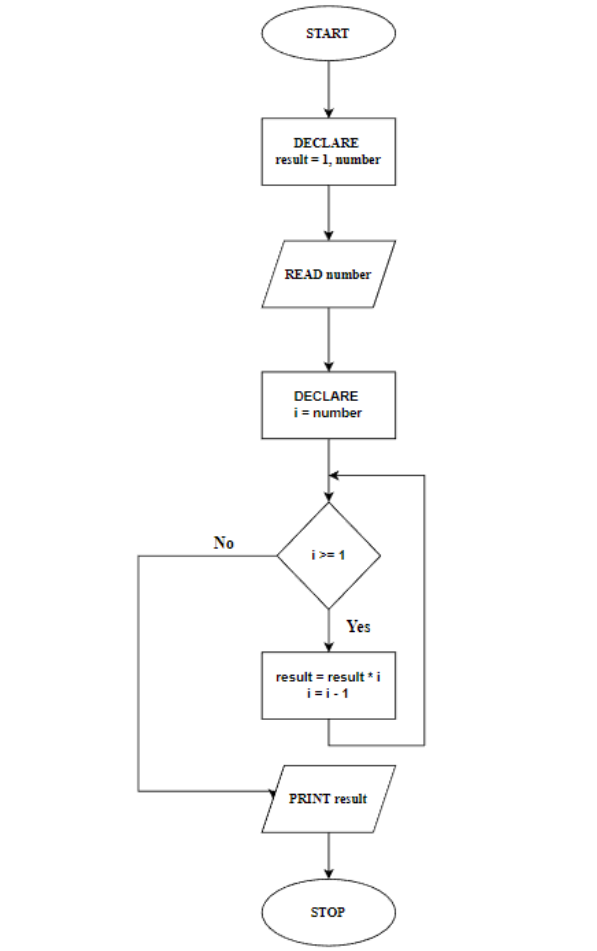
result = result * i

i = i – 1

Step6 – PRINT result.

Step7 - STOP

Flowchart:



Q3] Write a Java Program to find the Factorial of a given number.

Algorithm:

Step1 – START.

Step2 – PRINT “Enter a number:”

Step2 – READ N.

Step3 – DECLARE fact

Step4 – fact = CALL factorial(N)

Step5 – PRINT fact.

Step7 – STOP.

factorial(int N)

Step1 – DECLARE f = 1

Step2 – if (N == 1)

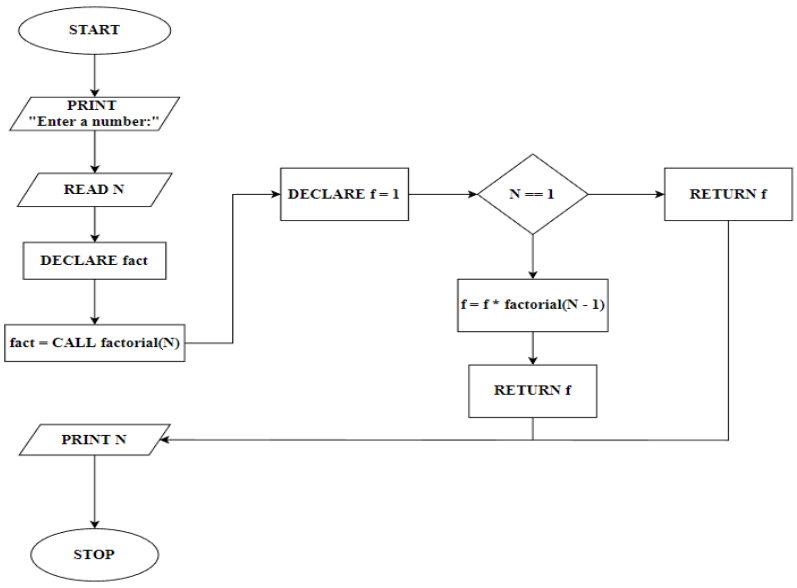
return f

else

f = N * factorial(N -1)

return f

Flowchart:



Q4] Swap two numbers without using the third variable approach.

Algorithm:

Step1 – START.

Step2 – DECLARE a = 2, b = 5

Step2 – PRINT “Before Swapping”

PRINT a, b

Step3 – a = a + b

Step5 – b = a – b

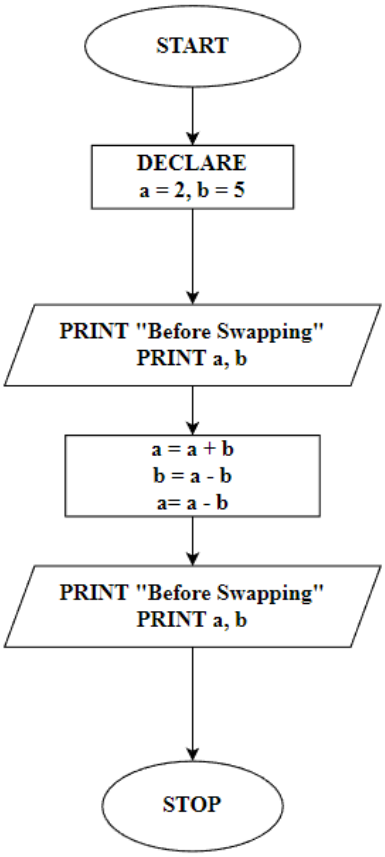
Step6 – a = a – b

Step7 – PRINT “After Swapping”.

PRINT a, b

Step8 – STOP.

Flowchart:



Q5] How to check whether the given number is Positive or Negative in Java?

Algorithm:

Step1 – START.

Step2 – DECLARE number

Step3 – READ number

Step4 – IF number > 0

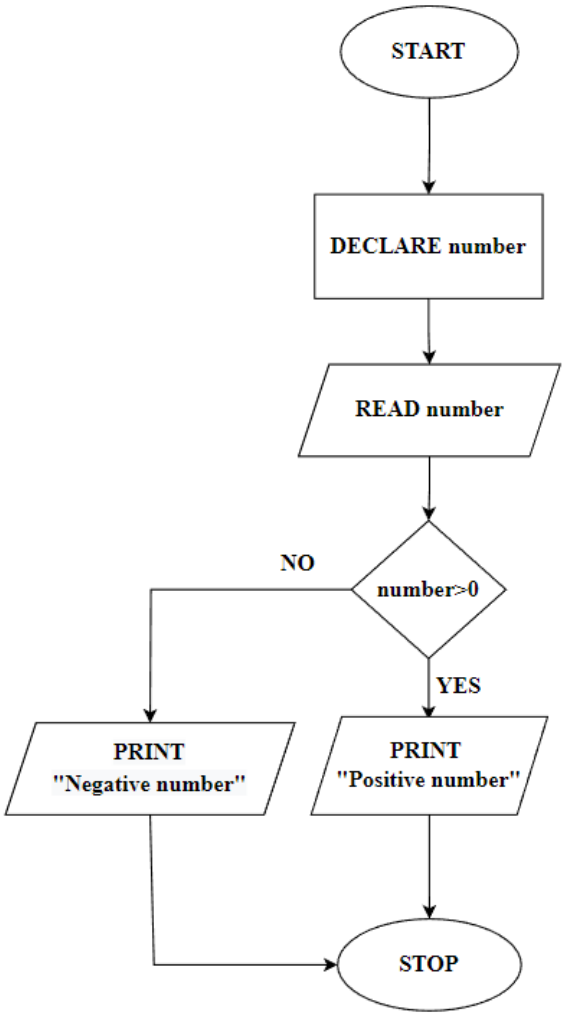
THEN PRINT “Positive Number.”

ELSE

PRINT “Negative Number.”

Step5 – STOP.

Flowchart:



Q6] Write a JAVA Program to find whether a given number is Leap year or not.

Algorithm:

Step1 – START.

Step2 – READ Year

Step3 – IF (Year % 4 == 0 && Year % 100 != 0) || (Year % 400 == 0)

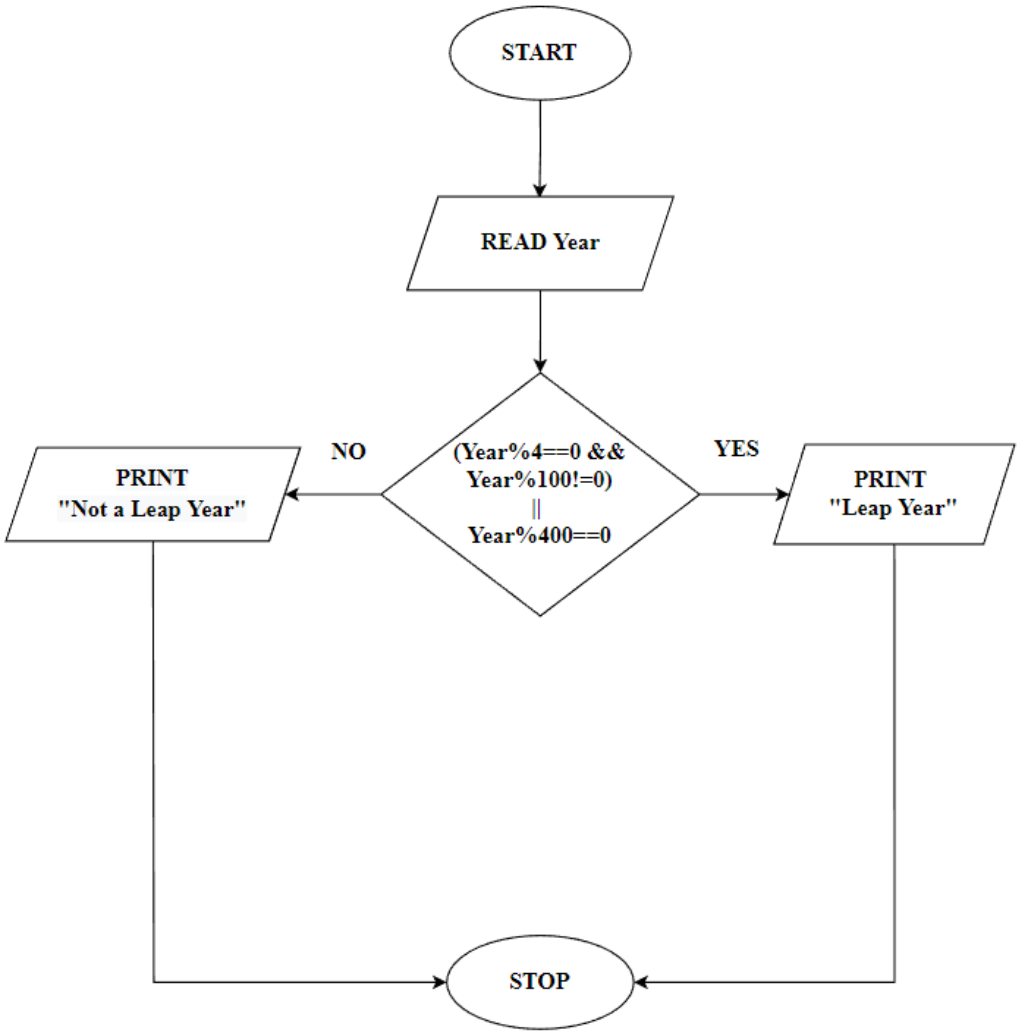
THEN PRINT “Leap Year.”

ELSE

PRINT “Not a Leap Year.”

Step4 – STOP.

Flowchart:



Q7] Write a Java Program to Print 1 to 10 Without Using Loop.

Algorithm:

Step1 – START.

Step2 – DECLARE N =1

Step3 – CALL printNumber(N)

Step4 – STOP.

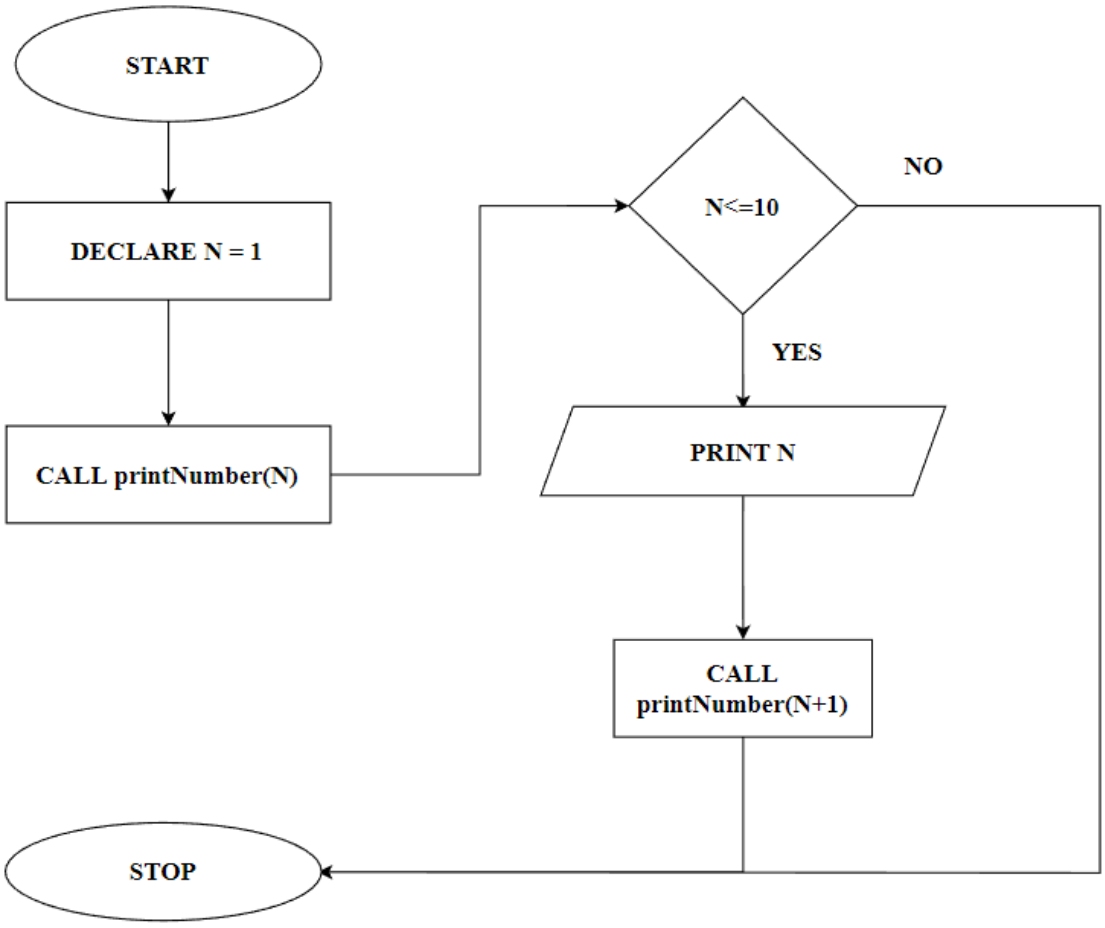
printNumber(int N)

Step1 – IF N< = 10

THEN PRINT N

CALL printNumber(N+1)

Flowchart:



Q8] Write a Java Program to print the digits of a Given number.

Algorithm:

Step1 – START.

Step2 – DECLARE number, rem

Step3 – READ number

Step4 – REPEAT until number != 0

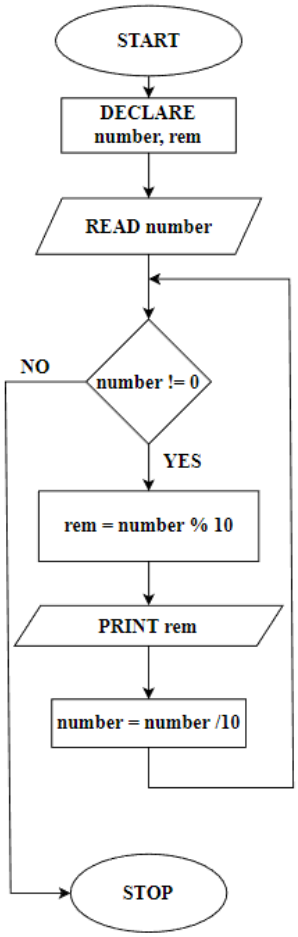
rem = rem % 10

PRINT rem

number = number / 10

Step5 – STOP.

Flowchart:



Q9] Write a Java Program to print all the factors of the Given number.

Algorithm:

Step1 – START.

Step2 – DECLARE number, i = 1

Step3 – READ number

Step4 – REPEAT until i < number

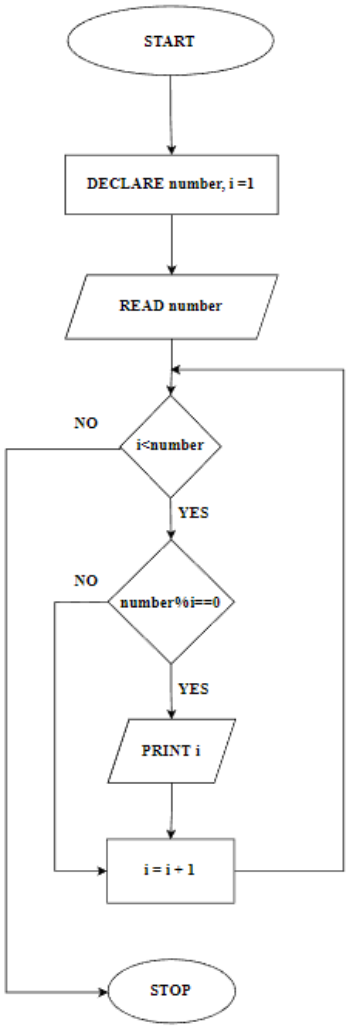
IF number % i == 0

THEN PRINT i

i = i + 1

Step5 – STOP.

Flowchart:



Q10] Write a Java Program to find the sum of the digits of a given number.

Algorithm:

Step1 – START.

Step2 – DECLARE number, rem, sum = 0

Step3 – READ number

Step4 – REPEAT until number ! = 0

rem = rem % 10

sum = sum + rem

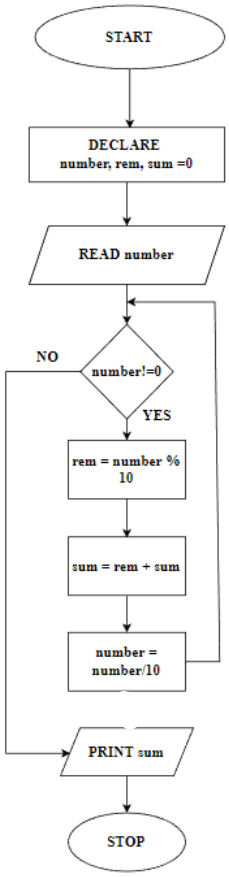
number = number / 10

THEN PRINT i

Step5 – Print sum

Step6 – STOP.

Flowchart:



Q11] Write a Java Program to find the smallest of the digits of a given number.

Algorithm:

Step1 – START.

Step2 – DECLARE n1, n2, n3

Step3 – READ n1, n2, n3

Step4 – IF n1 < n2 && n1 < n3

THEN PRINT Smallest n1

ELSE IF n2 < n3

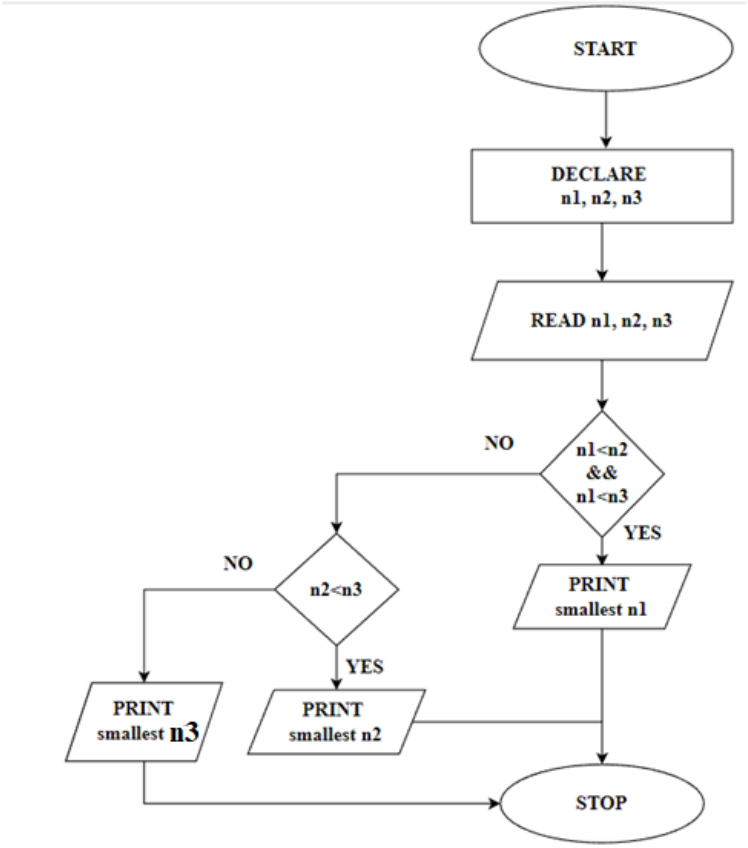
THEN PRINT Smallest n2

ELSE

PRINT Smallest n3

Step5 – STOP.

Flowchart:



Q12] How to add two numbers without using the arithmetic operators in Java?

Algorithm:

Step1 – START.

Step2 – DECLARE a = 10, b = 23, sum = 0

Step3 – sum = CALL add(a, b)

Step4 – PRINT sum

Step5 – STOP.

add(int a, int b)

Step1 – DECLARE i = 1

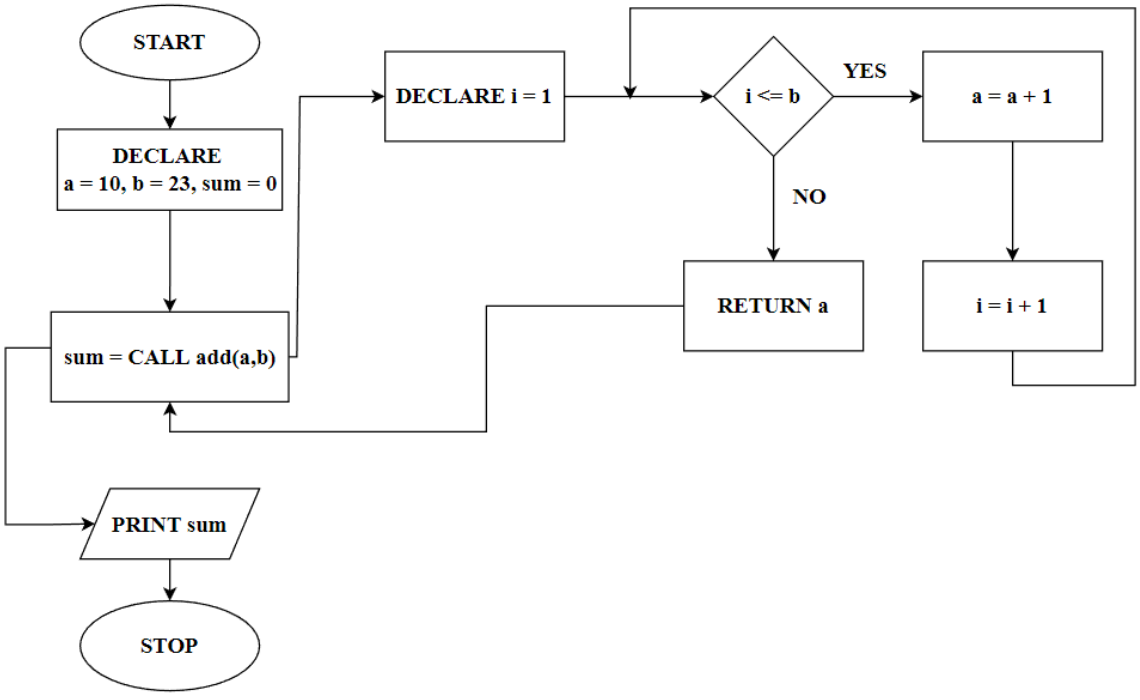
Step2 – REPEAT until i <= b

a = a + 1

i = i + 1

Step3 – RETURN a

Flowchart:



Q13] Write a java program to Reverse a given number.

Algorithm:

Step1 – START.

Step2 – DECLARE number, rev = 0, rem

Step3 – READ number

Step4 – REPEAT until number != 0

rem = rem % 10

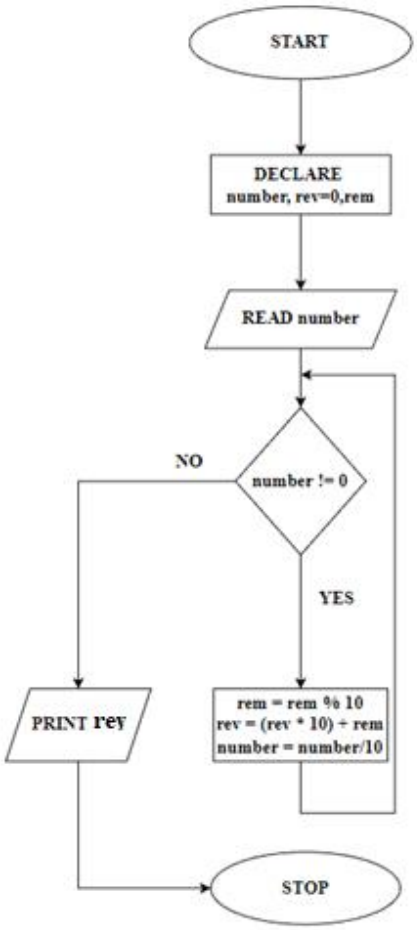
rev = (rev * 10) + rem

number / 10

Step5 – PRINT rev

Step6 - STOP

Flowchart:



Q14] Write a java program to find the GCD of two given numbers.

Algorithm:

Step1 – START.

Step2 – DECLARE n1, n2, i = 1, gcd = 1

Step3 – READ n1, n2

Step4 – REPEAT until i <= n1 && i <= n2

IF n1 % i == 0 && n2 % i == 0

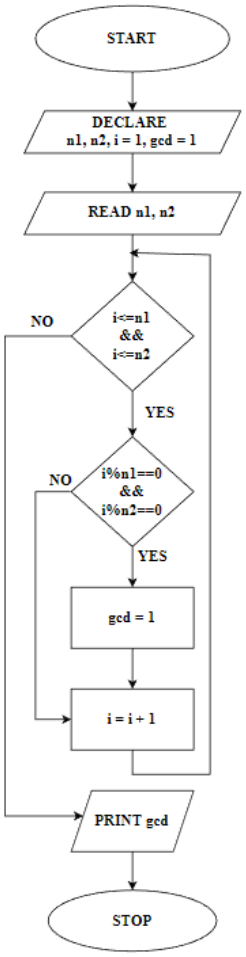
THEN gcd = i

i = i + 1

Step5 – PRINT gcd

Step6 - STOP

Flowchart:



Q15] Write a java program to find the LCM of two given numbers.

Algorithm:

Step1 – START.

Step2 – DECLARE n1, n2, i = 1, gcd = 1, lcm

Step3 – READ n1, n2

Step4 – REPEAT until i <= n1 && i <= n2

IF n1 % i == 0 && n2 % i == 0

THEN gcd = i

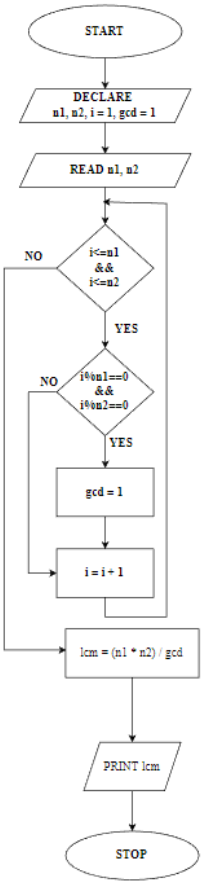
i = i + 1

Step5 – lcm = (n1 * n2) / gcd

Step6 – PRINT lcm

Step7 – STOP

Flowchart:



Q16] Write a java program to find the LCM of two given numbers using Prime Factors method.

Algorithm:

Step1 – START.

Step2 – DECLARE n1, n2, i =2, lcm =1

Step3 – READ n1, n2

Step4 – REPEAT until i<= n1 && i <= n2

IF n1 % i == 0 && n2 % i == 0

n1 = n1 / i

n2 = n2 / i

lcm = lcm * i

ELSE

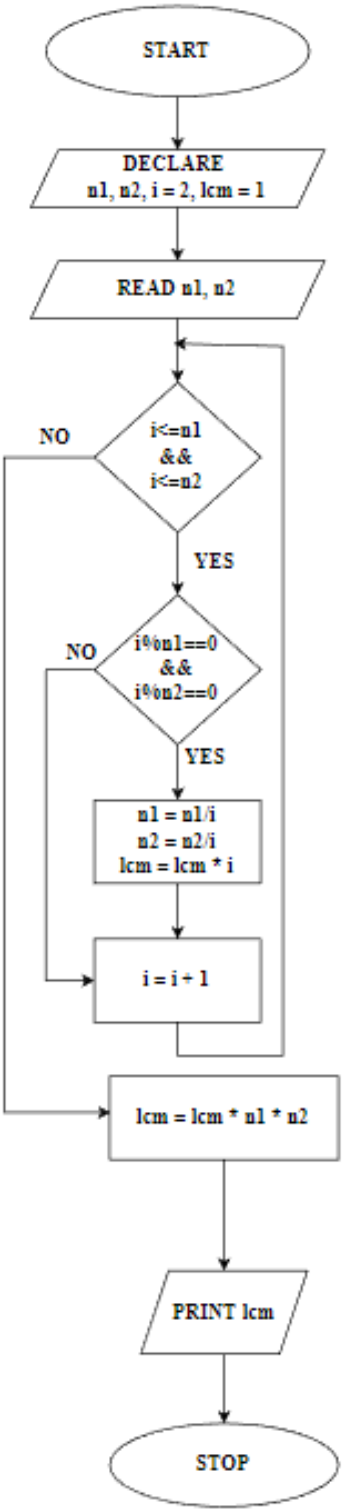
i = i + 1

Step5 – lcm = lcm * n1 * n2

Step6 – PRINT lcm

Step7 – STOP

Flowchart:


START
DECLARE
n1, n2, i = 2, lcm = 1
READ n1, n2
i<=n1
&&
i<=n2
NO
YES
i%n1==0
&&
i%n2==0
NO
YES
n1 = n1/i
n2 = n2/i
lcm = lcm * i
i = i + 1
lcm = lcm * n1 * n2
PRINT lcm
STOP

Q17] Check whether the Given Number is a Palindrome or not.

Algorithm:

Step1 – START.

Step2 - PRINT “Enter a number:”

Step3 – READ number

Step4 – DECLARE rem, rev, temp = number

Step5 – REPEAT until number != 0

rem = number % 10

rev = (rev * 10) + rem

number = number /10

Step6 - IF temp == rev

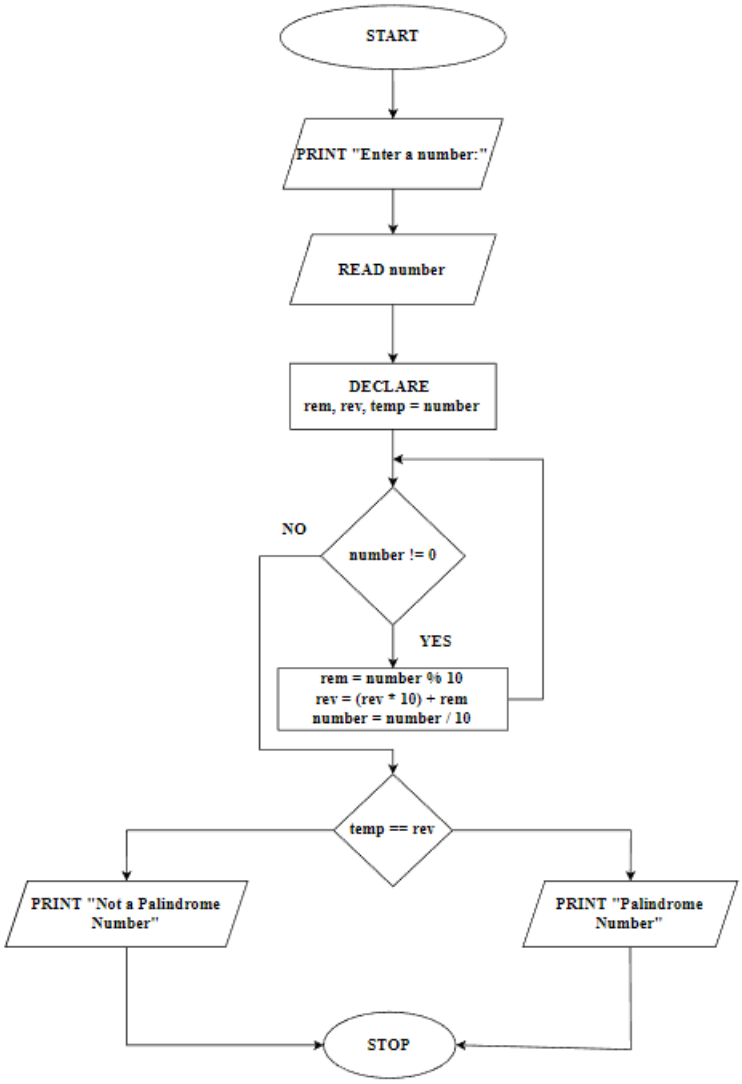
THEN PRINT “Palindrome Number”

ELSE

PRINT “Not Palindrome Number”

Step7 – STOP

Flowchart:


START
PRINT "Enter a number:"
READ number
DECLARE
rem, rev, temp = number
NO
number != 0
YES
rem = number % 10
rev = (rev * 10) + rem
number = number / 10
temp == rev
PRINT "Not a Palindrome Number"
PRINT "Palindrome Number"
STOP

Q18] Write a Java Program to print all the Prime Factors of the Given Number.

Algorithm:

Step1 – START

Step2 – PRINT “Enter a number:”

Step3 – READ n

Step4 – REPEAT until i < n

REPEAT until n % i == 0

PRINT i

n = n / i

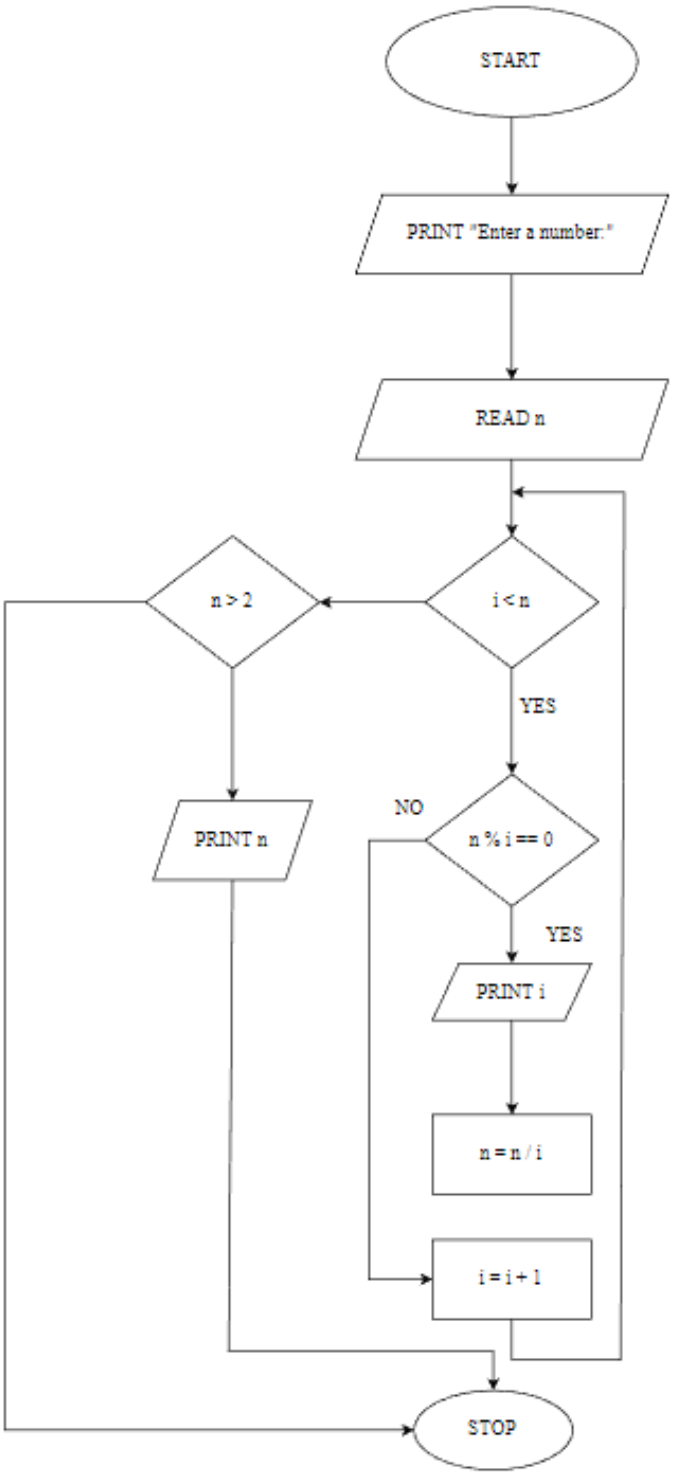
i = i + 1

Step5 – IF number > 2

THEN PRINT n

Step6 – STOP

Flowchart:


START
PRINT "Enter a number:"
READ n
i < n
n > 2
YES
NO
PRINT n
n % i == 0
YES
PRINT i
n = n / i
i = i + 1
STOP

Q19] To print the following series EVEN number series 2 4 6 8 10 12 14 16…..

Algorithm:

Step1 – START.

Step2 - DECLARE number, i = 1

Step3 – READ number

Step4 – REPEAT until i <= number

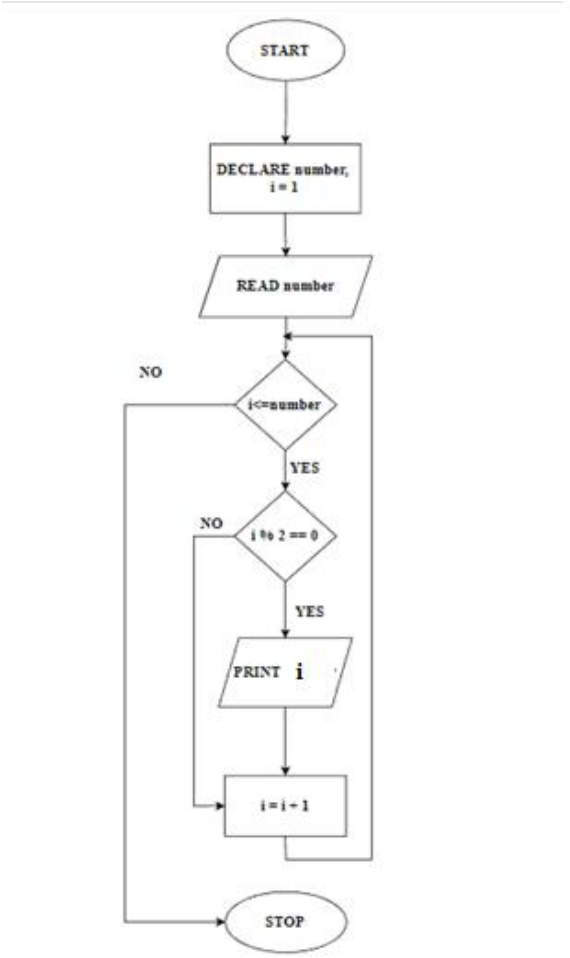
IF i % 2 == 0

THEN PRINT i

i = i + 1

Step5 – STOP

Flowchart



Q20] To print the following series ODD number series 1 3 5 7 9 11 13…..

Algorithm:

Step1 – START.

Step2 - DECLARE number, i = 1

Step3 – READ number

Step4 – REPEAT until i <= number

IF i % 2 != 0

THEN PRINT i

i = i + 1

Step5 – STOP

Flowchart