Assignment - 1

Assignment - 1

Name : MAHENDRA PRATAP ROY

PRN No. : 220940320059

Centre : CDAC Mumbai Khargarh

Date : 07/09/2022

Q1> Check if the given number is Even or odd.

Algorithm:

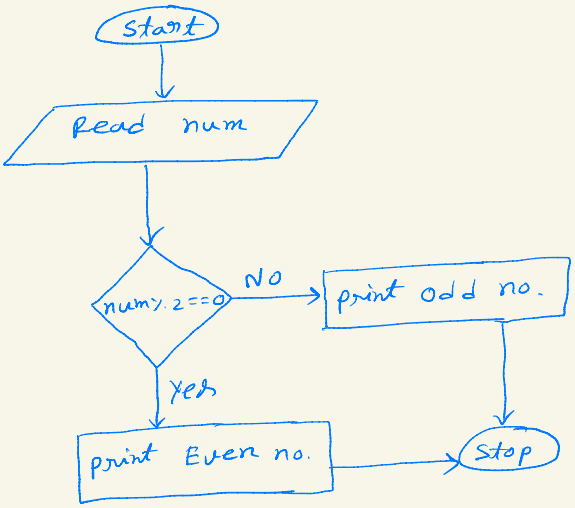
1> Read given number

2> check number %. 2 == 0

3> if divisible then even no.

4> otherwise odd no.

Flowchart:

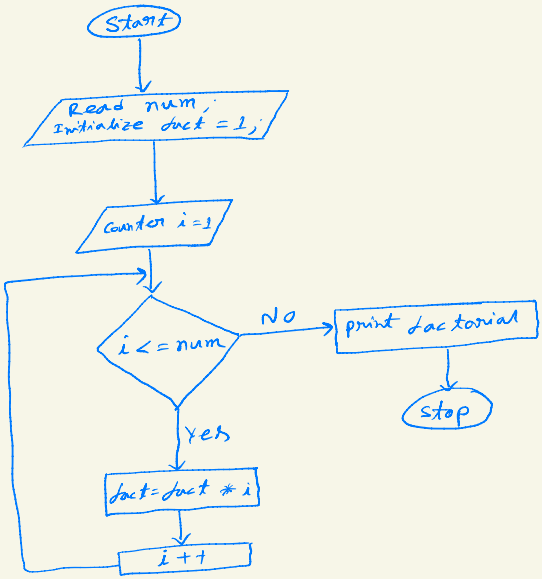


Q2> Write a Java program to find the Factorial of a given number.

Algorithm:

1) Read the num from user
2) Take a factorial variable initialized to 1.
3) Take a for loop where i is counter.
4) Use the process fact = fact * i till i <= num

Flowchart:



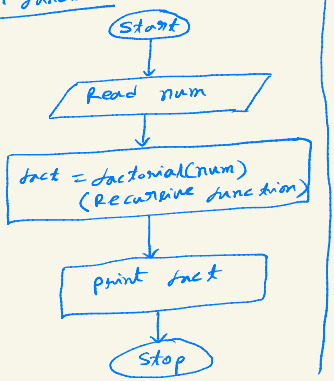
Q3> Find the factorial of a number using Recursion.

Algorithm:

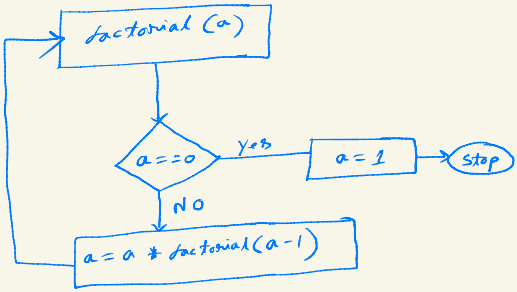
1) Read the num from user
2) Create a factorial function with parameter a where if a = 0 return 1
3) otherwise return a = a * factorial(a-1)
4) Call the factorial (num) from main function and print the output.

Flowchart:

main function



Recursive function

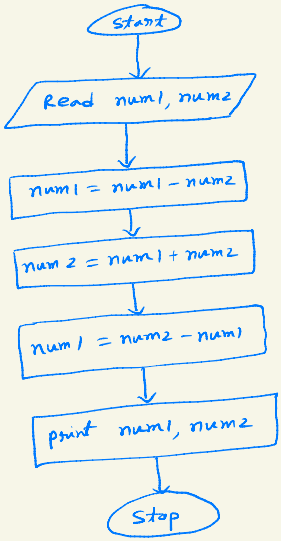


Q4> Swap two numbers without using third variable approach.

Algorithm:

1) Read two numbers num1, num2
2) Then subtract the numbers and store it to num1
3) then add new num1 to num2 and store it to num2
4) then subtract num1 from new num2 and store it to num1.
5) print num1 and num2.

Flowchart:

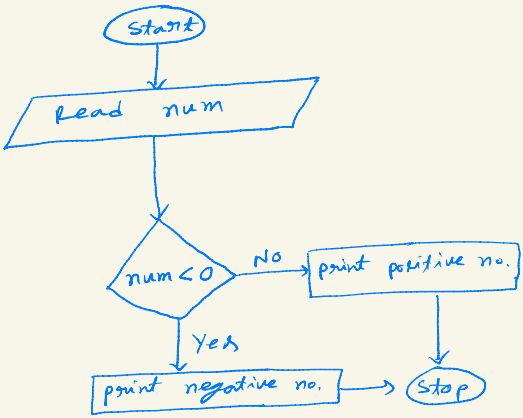


Q5> How to check whether the given number is positive or negative in Java.

Algorithm:

1) Read the num.
2) Check num < 0 then print negative
3) Otherwise print positive

Flowchart:

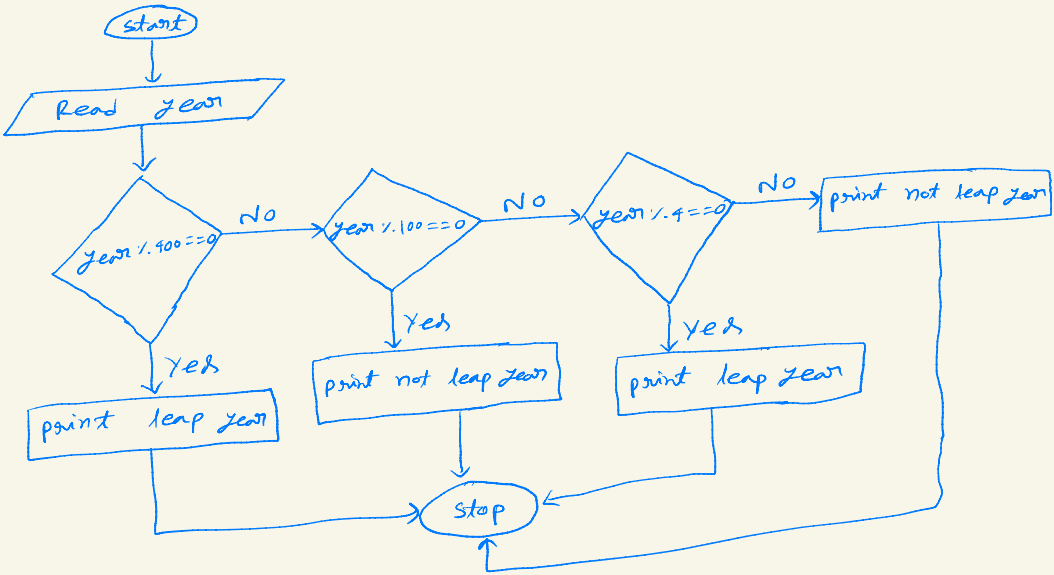


Q6> Write a Java program to find whether a given number is leap year not.

Algorithm:

1) Read the year.
2) Check the year %. 400 == 0
   then print leap year
3) Check the year %. 100 == 0
   then not leap year
4) Check the year %. 4 == 0
   then print leap year.

Flowchart:



Q7> Write a Java program to print 1 to 10 without using loop.

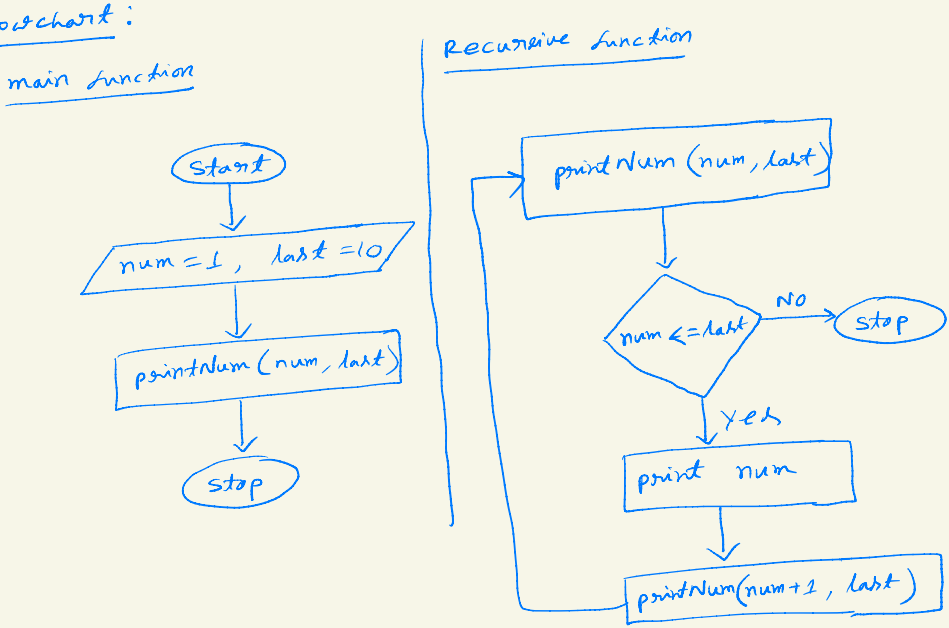
Algorithm:

1> Create a recursive function printNum() with parameter num = 1 and last = 10

2> Check num <= last if yes then print the num otherwise terminate

3> Increase num by 1

Flowchart:

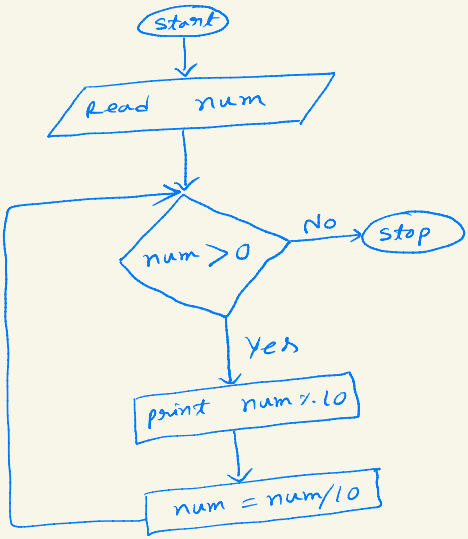


Q8> Write a Java program to print the digits of a given number.

Algorithm:

1) Read the num.
2) Take a loop with condition num > 0
3) Inside loop do num % 10 and print it.
4) Also do num = num/10

Flowchart:

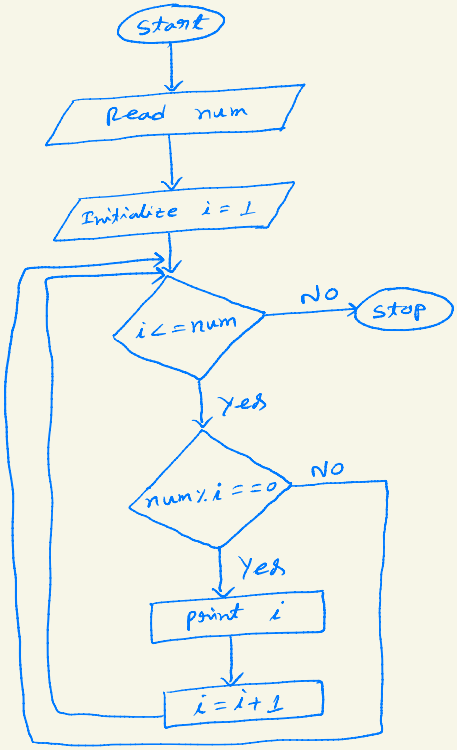


Q9) Write a Java program to print all the factors of the given number.

Algorithm:

1) Read the num.
2) Take a loop with i = 1 and condition i <= num
3) Inside loop check for num % i == 0 if yes print the i

Flowchart:



Q 10 > Write a Java program to find the sum of the digits of a given number.

Algorithm :

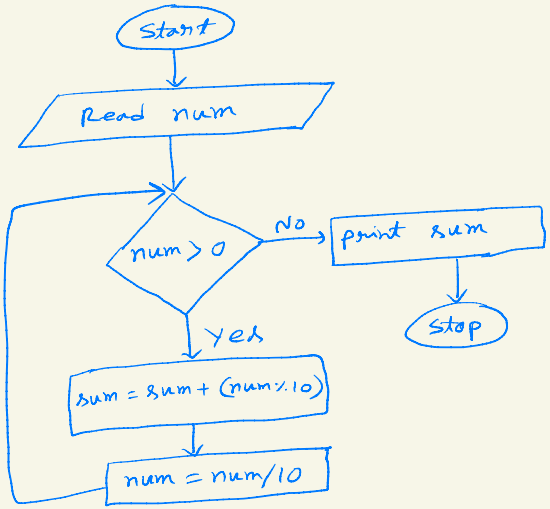
1> Read the num

2> Take a loop and extract the digits one by one

3> In same loop add the digits one after another

4> print the final addition

Flowchart :

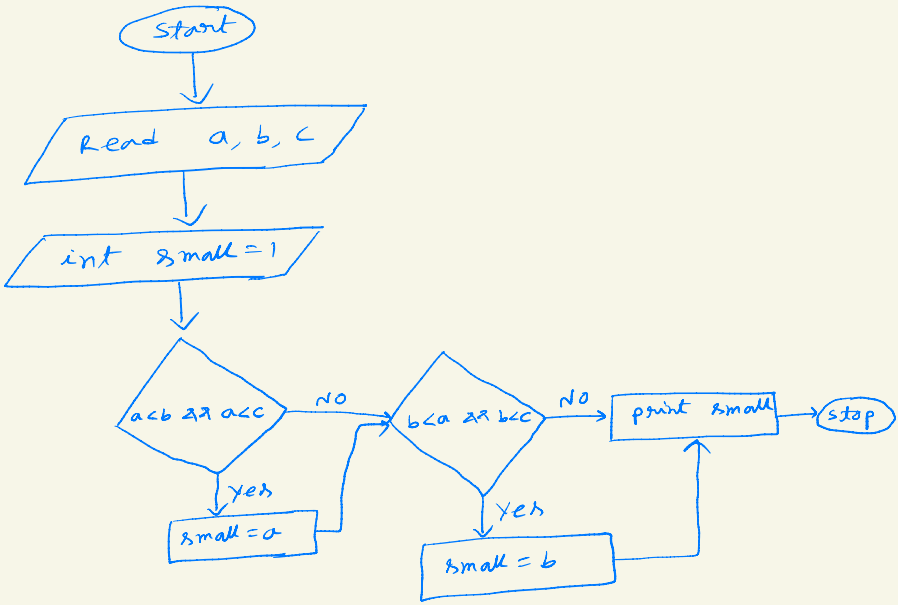


Q11) Write a Java program to find the smallest of 3 nos (a, b, c).

Algorithm:

1) Read 3 numbers a, b, c
2) Check a < b and a < c
   then store a to small
3) Check b < a and b < c
   then store b to small
4) print small

Flowchart:



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

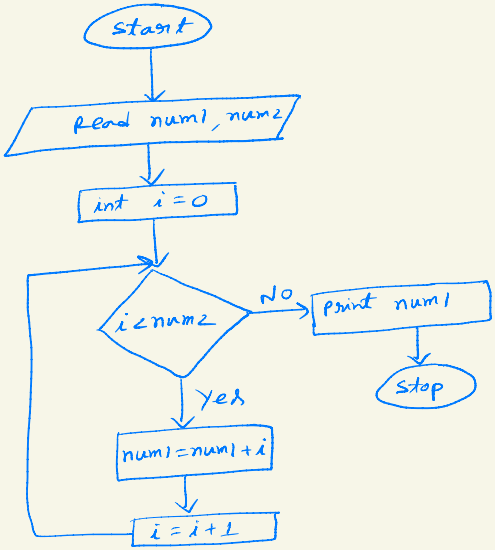
Algorithm:

1> Read num1, num2

2> Use a loop where i start from 0 and condition is i<num2

3> add i to num1 till termination of the loop.

Flowchart:

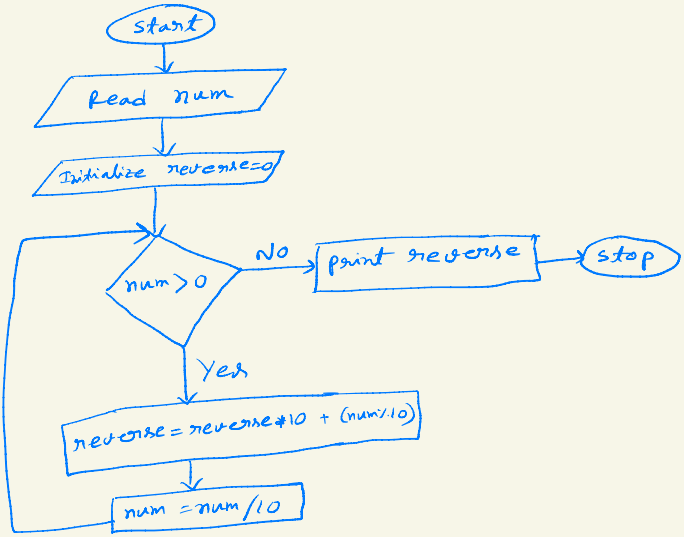


Q13) Write a java program to reverse a given number.

Algorithm:

1) Read the num.
2) Take a loop with condition num > 0
3) Update the reverse with num % 10 result
4) Do num = num/10
5) print the reverse at last.

Flowchart:



Q 19> Write a Java program to find the GCD of two given numbers.

Algorithm:

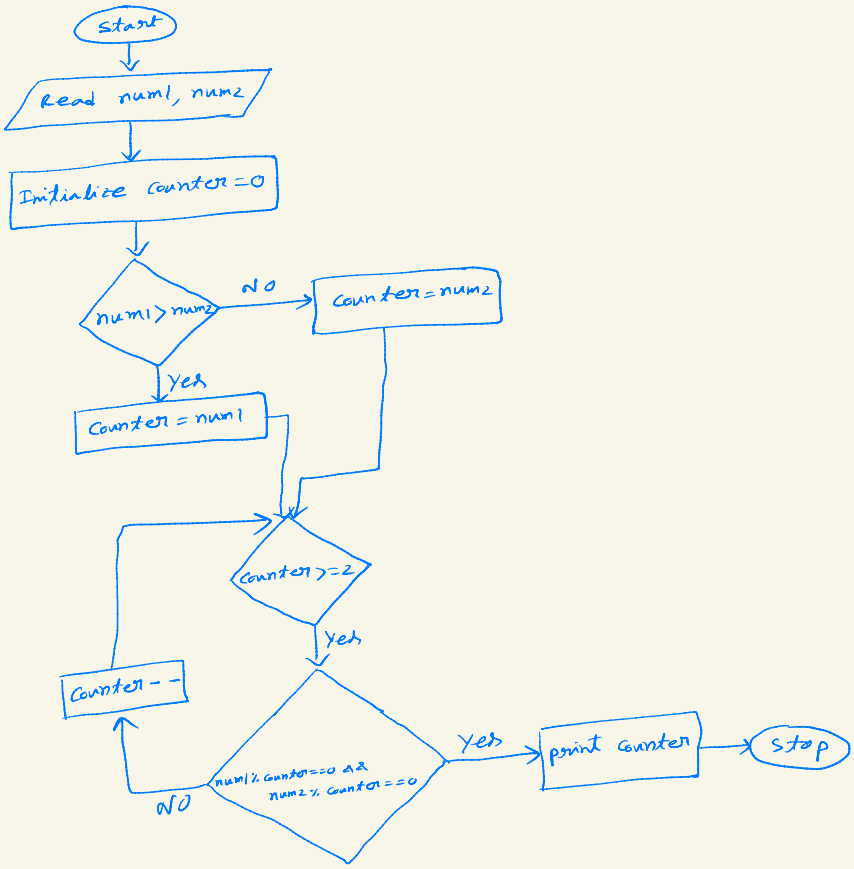
1) Read the two numbers num1, num2
2) Find maximum of num1 & num2 and assign the maximum to a counter
3) Use a loop with condition counter >= 2
4) Inside loop check don

num1 % counter == 0 && num2 % counter == 0

If yes break the loop

5) print counter

Flowchart:

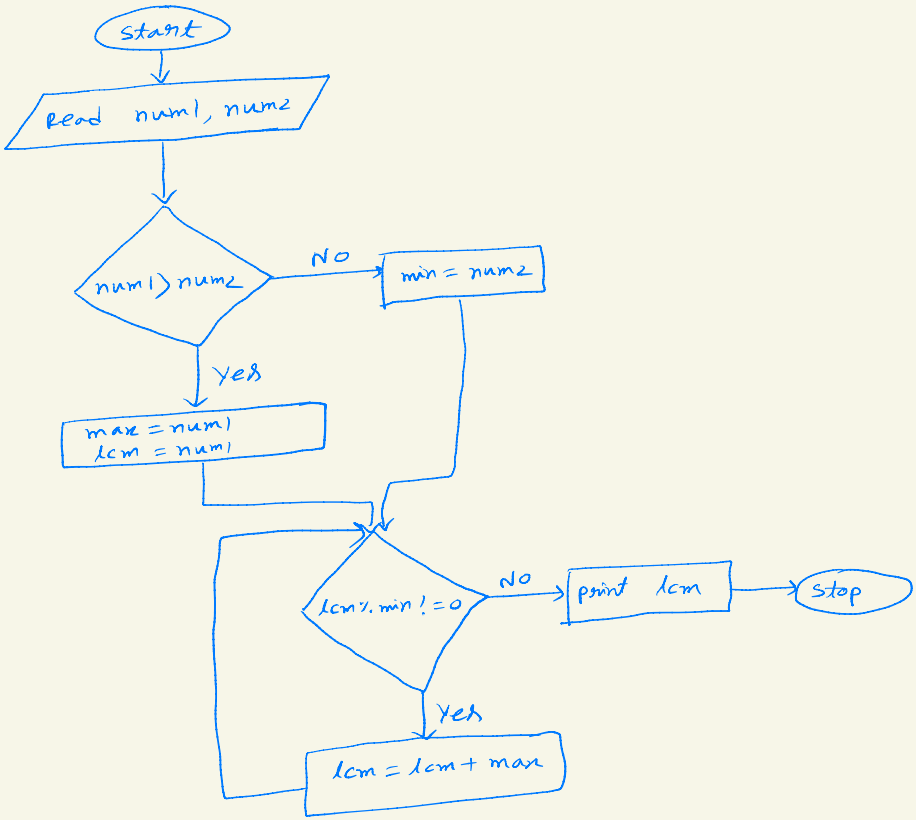


Q 15> Write a Java program to find LCM of two given numbers.

Algorithm :

1) Read two number num1, num2
2) Compare num1 & num2 and store large number to max and lcm, & store small number to min.
3) Use loop with condition lcm % min != 0
4) Inside loop do lcm = lcm + max
5) print lcm at last

Flowchart :

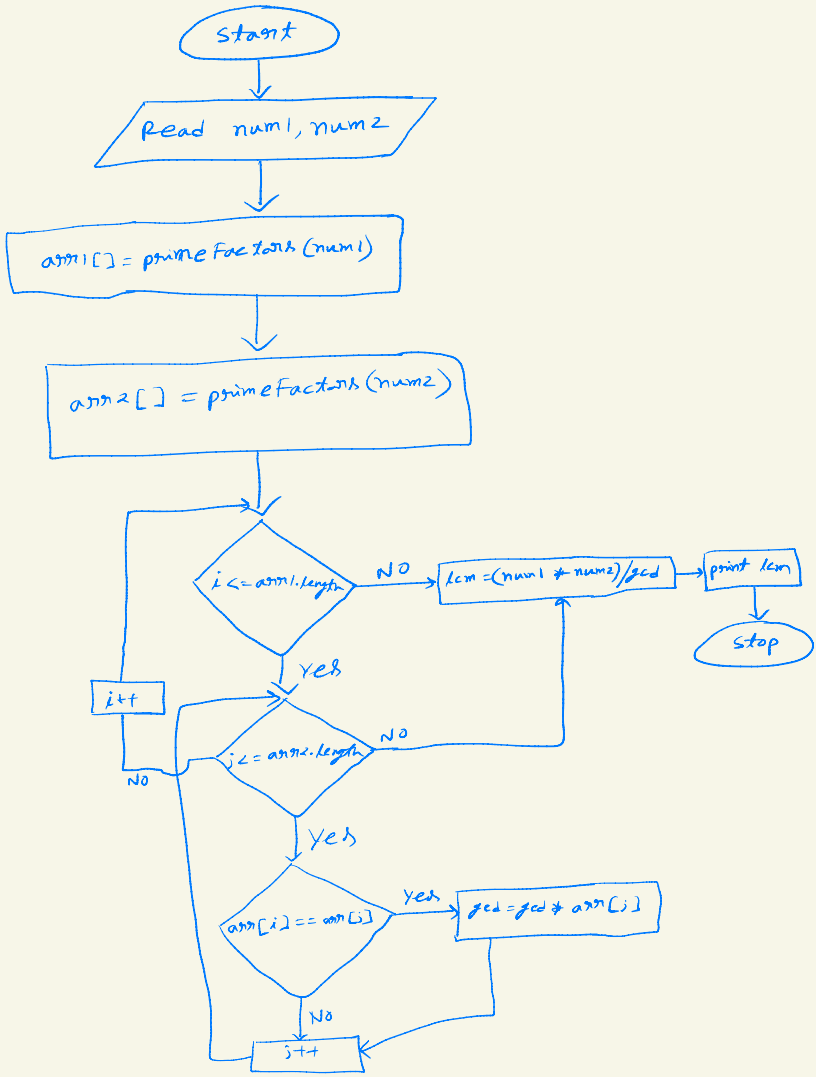


Q16> Write a Java program to find LCM of two numbers using the prime factors method.

Algorithm:

1) Read two numbers
2) Find prime factors of both numbers
3) Compare the primefactors
4) Then multiply the primefactors to find lcm.

Flowchart:

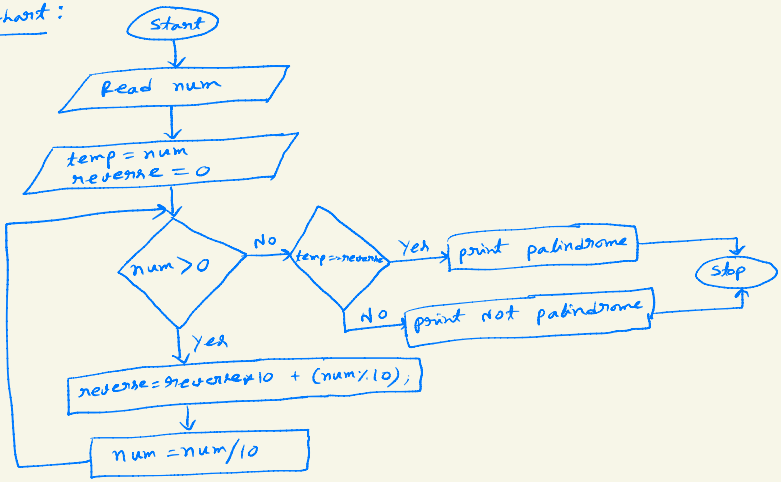


Q17> Check whether the given number is palindrome or not.

Algorithm:

1> Read the num
2> store num to a temp variable
3> Reverse the num
4> Check temp == reverse
   if yes then print palindrome
5> otherwise print not a palindrome

Flowchart:

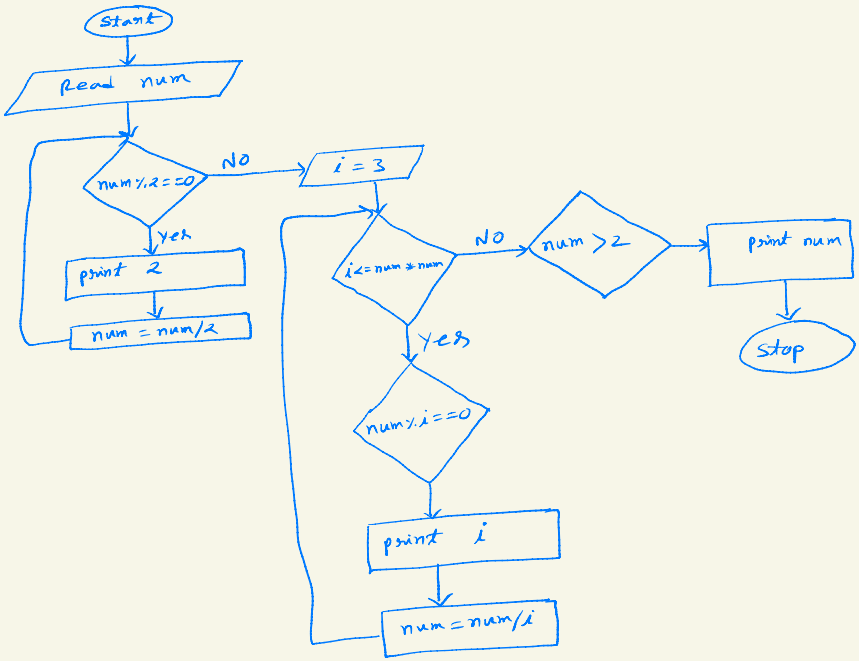


Q18> Write a Java program to print all the prime factors of the given number.

Algorithm:

1) Read the num
2) Check for num%.2==0
3) Then take a loop of i=3 to i<=num*num
4) Again Inside loop take a loop of Condition num%i==0
5) Do num = num/i
6) Again out of the loop check for prime number greater than 2

Flowchart:

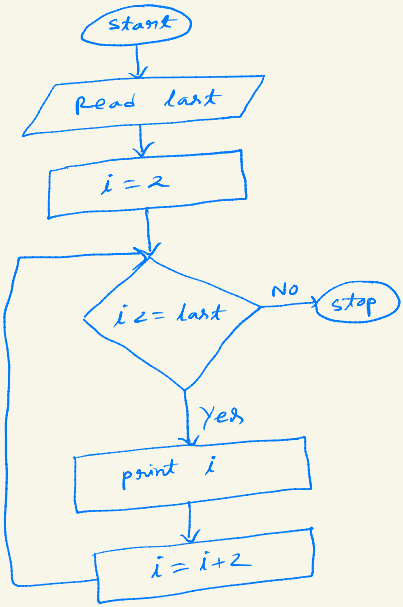


Q19> To print the following series Even number series 2 4 6 8 10 12 14 - - - -

Algorithm:

1) Read the last number
2) Use a loop where $i=2$ and $i <= last$
3) print i
4) DO $i = i+2$

Flowchart:



Q 20> To print the following series ODD number series

1 3 5 7 9 11 13 . . .

Algorithm :

1> Read the last number

2> Use a loop from $i=1$ to $i <= last$

3> print i

4> Do $i = i+2$

Flowchart :