

Assignment - 1

(Q1) Check if the given number is Even or Odd.

⇒ Algorithm :-

Step 1 :- Start

Step 2 :- Read input n.

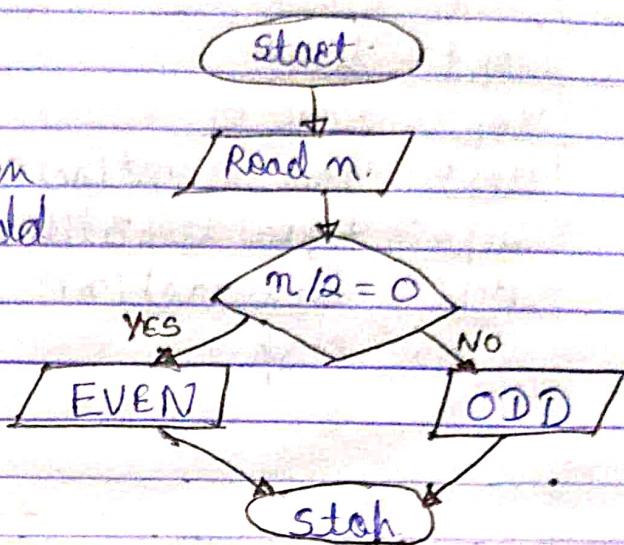
Step 3 :- $n \div 2 = 0$

 Yes. Print 'n' is Even

 No. Print 'n' is odd

Step 4 :- Stop.

Flowchart :-



(Q2) Write a program to find the Factorial of given number.

⇒ Algorithm :-

Flowchart

Step 1 :- Start.

Step 2 :- Read Input n, fact = 1, i = 1

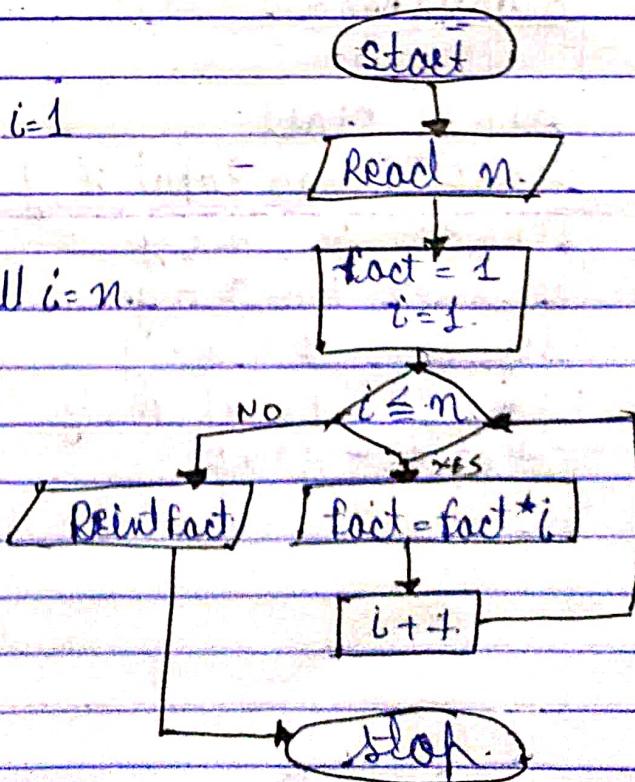
Step 3 :- fact = fact * i;

Step 4 :- i ++

Step 5 :- Repeat step 3 & 4 till i = n.

Step 6 :- Print fact

Step 7 :- Stop.



(Q3) Find the Factorial of a number using Recursion.

Algorithm:-

Step 1:- Start

Step 2:- Read n

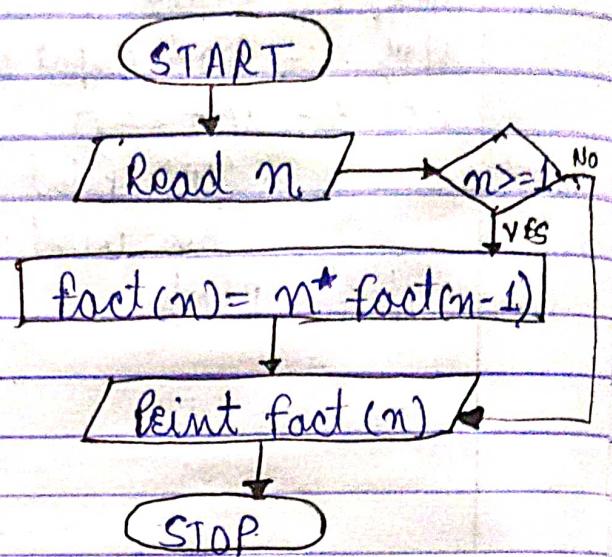
Step 3:- fact(n) = $n * \text{fact}(n-1)$

Step 4:- Repeat Step 3 till n=1

Step 5:- Print fact(n)

Step 6:- Stop

Flowchart:-



(Q4) Swap two numbers without using the third variable approach.

Algorithm:-

Flowchart:-

Step 1:- Start

Step 2:- Read input x, y .

Step 3:- $x = x + y$

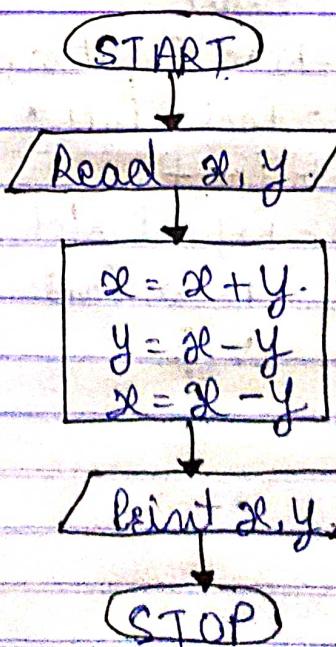
Step 4:- $y = x - y$

Step 5:- $x = x - y$

Step 6:- Print x, y .

Step 7:- Stop.

Flowchart:-



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

Algorithm:-

Step 1:- Start

Step 2:- Read input n

Step 3:- If $n > 0$

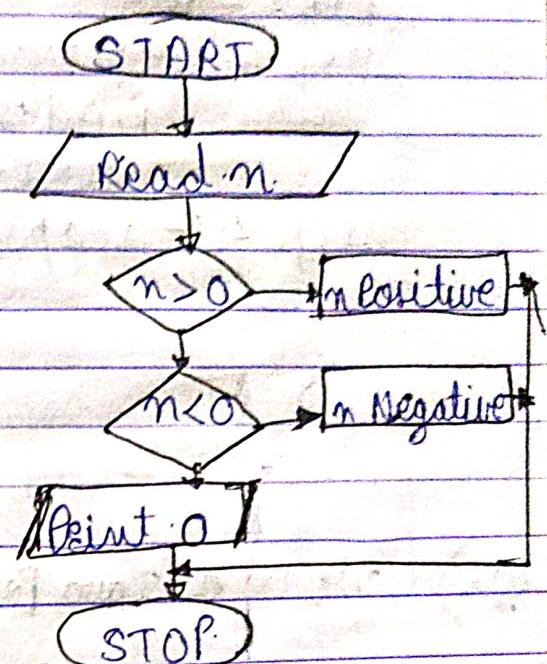
 YES print Positive

 if $n < 0$

 print Negative.

Step 4:- Stop.

Flowchart



(Q6) Write a program to find given number is leap year or not.

Algorithm:-

Step 1:- Start

Step 2:- Read input N.

Step 3:- If $N \% 4 = 0$

 yes then

$N \% 100 = 0$

 No print "Not leap year"

 if $N \% 100 = 0$

 yes then

$N \% 400 = 0$

 Yes print "Leap year"

 No print "Not leap year"

Step 4:- Stop.

Flowchart:-

```
graph TD; START([START]) --> READ[/Read n/]; READ --> DECISION1{ }; DECISION1 -- "N \% 4 = 0" --> DECISION2{ }; DECISION2 -- "No" --> NOTLEAP["Not leap year"]; DECISION2 --> DECISION3{ }; DECISION3 -- "N \% 100 = 0" --> DECISION4{ }; DECISION4 -- "No" --> LEAP["Leap year"]; DECISION4 --> DECISION5{ }; DECISION5 -- "Yes" --> DECISION6{ }; DECISION6 -- "N \% 400 = 0" --> LEAP; DECISION6 --> NOTLEAP; DECISION5 --> NOTLEAP;
```

The flowchart starts with an oval labeled "START". An arrow points down to a parallelogram labeled "Read n". From "Read n", an arrow points to a diamond labeled "N \% 4 = 0". If "No", the flow goes to a rectangle labeled "Not leap year". If "Yes", the flow goes to another diamond labeled "N \% 100 = 0". If "No", the flow goes to a rectangle labeled "Leap year". If "Yes", the flow goes to another diamond labeled "N \% 400 = 0". If "Yes", the flow goes to a rectangle labeled "Leap year". If "No", the flow goes to a rectangle labeled "Not leap year". Both paths from the diamonds converge at a rectangle labeled "Not leap year".

START

Read n

$N \% 4 = 0$ No Not leap year

$N \% 100 = 0$ Yes Leap year

$N \% 400 = 0$ Yes Not leap year

$N \% 400 = 0$ No Not leap year

$N \% 400 = 0$ Yes Leap year

$N \% 400 = 0$ No Not leap year

$N \% 400 = 0$ Yes Not leap year

$N \% 400 = 0$ No Not leap year

$N \% 400 = 0$ Yes Leap year

$N \% 400 = 0$ No Not leap year

STOP

Q7) Write a Java program to print 1 to 10 without using loop.
Algorithm:-

Step 1:- Start

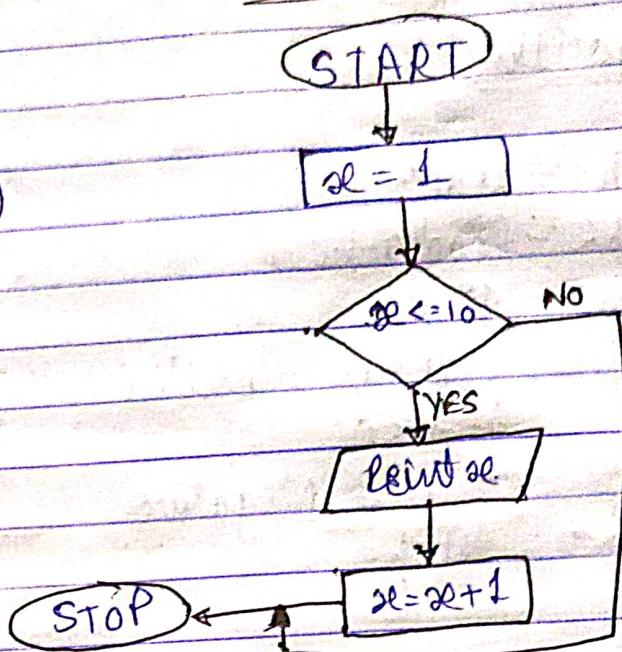
Step 2:- $x = 1$

Step 3:- If ($x \leq 10$)

Print x

$x = x + 1$

Step 4:- Stop



Q8) Write a Java Program . to print the digits of given number.

Algorithm:-

Step 1:- Start

Step 2:- Read input n

Step 3:- Sum = 0, R = 0

Step 4:- $R = n \% 10$

Step 5:- Print R .

Step 6:- $n = n / 10$

Step 7:- Repeat steps 4 to 6 till $n = 0$

Step 8:- Stop

(Q) Write a Java Program to print all Factors of Given Number.

Algorithm:

Flowchart:

Step 1:- Start

Step 2:- Read input $n, i=1$

Step 3:- $n \mod i == 0$ \rightarrow

Print i is factorial.

$i++$

START

Read $n, i=1$

if
 $(n \mod i == 0)$

$n \mod i == 0$

Step 4:- Repeat step 3 till $i=n$

Step 5:- Stop.

(Q) Write a Java program to find the sum of digits of Number.

Algorithm:

Flowchart:

Step 1:- Start

Step 2:- $Sum = 0, R = 0$

Step 3:- Read input n .

Step 4:- $R = n \% 10$

$Sum = Sum + R$

$n = n / 10$

Step 5:- Repeat step 4 till $n=0$

Step 6:- Print Sum

Step 7:- Stop.

START

Input $n ; Sum = 0$

If ($n >= 0$)

YES

$R = n \% 10$

$Sum = Sum + R$

$n = n / 10$

Point Sum

STOP

(Q11) Write a Java program to find the sum of digits of smallest of the 3 numbers (a,b,c)

Algorithm

Step 1:- START

Step 2:- Read input a,b,c

Step 3:- If ($a < b$)
yes $\rightarrow (a < c)$

yes print a is smaller

No print b is smaller

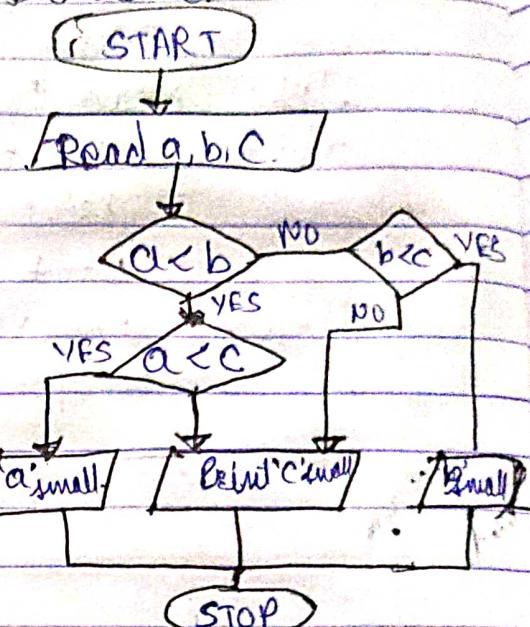
No $\rightarrow (b < c)$

yes print b is smaller

No print c is smaller

Step 4:- Stop.

Flowchart:-



(Q12) How to add two number without using arithmetic operators in Java?

Algorithm-

Step 1:- Start

Step 2:- Read input x, y .

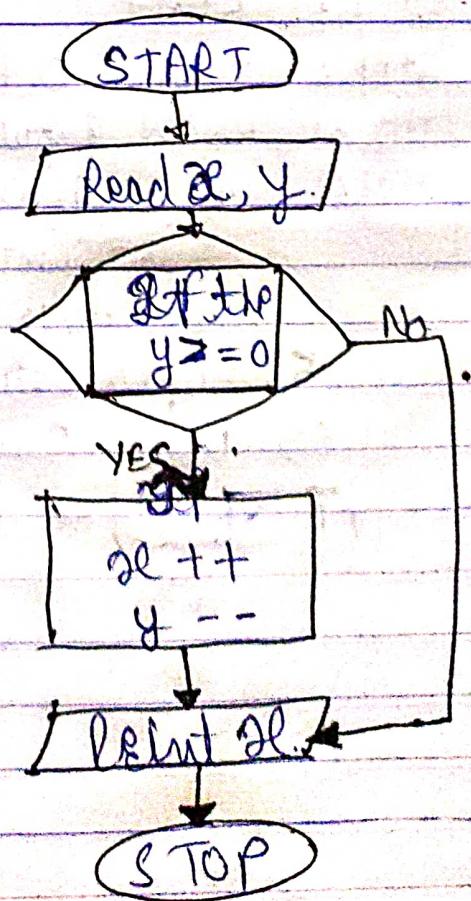
Step 3:- $x++ \& y--$

Step 4:- Repeat above till $y = 0$

Step 5:- Print $sum = x$.

Step 6:- Stop.

Flowchart:



(13) Write a Java program to Reverse a given number.

Algorithm

Step 1:- Start.

Step 2:- Read n, R=0, Reverse=0

Step 3:- $R = n \% 10$

Reverse = Reverse * 10 + R.

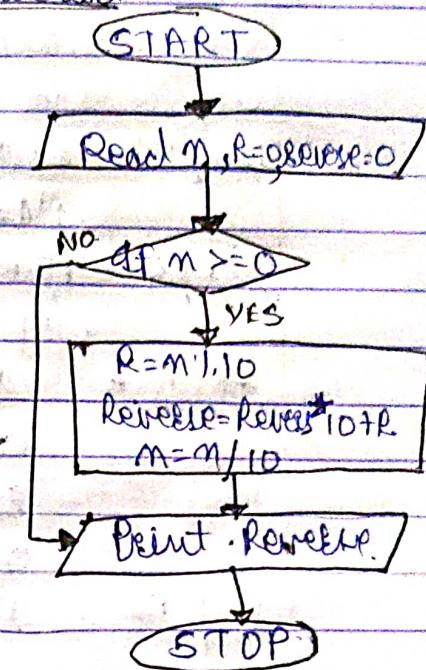
$n = n / 10$

Step 4:- Repeat Step 3 till $n=0$

Step 5:- Print Reverse.

Step 6:- Stop.

Flowchart



(14) Write program to find GCD of two given numbers.

Algorithm

Flowchart

Step 1:- Start

Step 2:- Read $m_1, m_2, G=1, i=1$

Step 3:- Repeat till ($m_1 \geq i \wedge m_2 \geq i$)

Step 4:- If $m_1 \% i = 0 \wedge m_2 \% i = 0$,

$G = i$

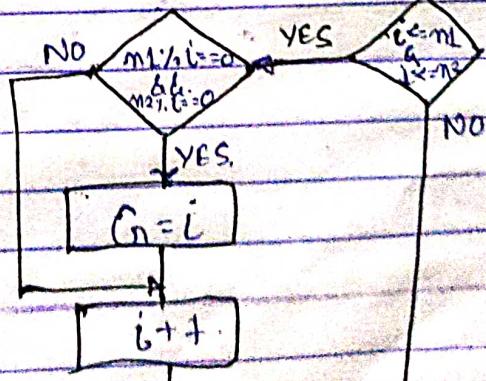
Step 5:- Print G

Step 6:- Stop.

START

/Read m_1, m_2 /

$G=1, i=1$



/Print G/

STOP

(Q15)

Write a program to find the LCM of two given numbers.
Algorithms Flowchart.

Step 1:- Start

Step 2:- Read m_1, m_2 .

Step 3:- Store Common Multiples of A & B into
max variable.

Step 4:- Check whether the max is divisible
by both m_1 & m_2 .

Step 5:- If max is divisible, display max
as LCM of two numbers.

Step 6:- Else, the value of max is increased
of 10 & go to Step 3.

Step 7:- Stop.

Q1) Check whether the Given number is Palindrome or Not
 Algorithm:

Step 1:- Start

Step 2:- Read n

Step 3:- ~~variables~~, R, reverse, &c.

Step 4:- ~~R~~ = $n \mod 10$

reverse = reverse*10 + rem.

$m = n / 10$

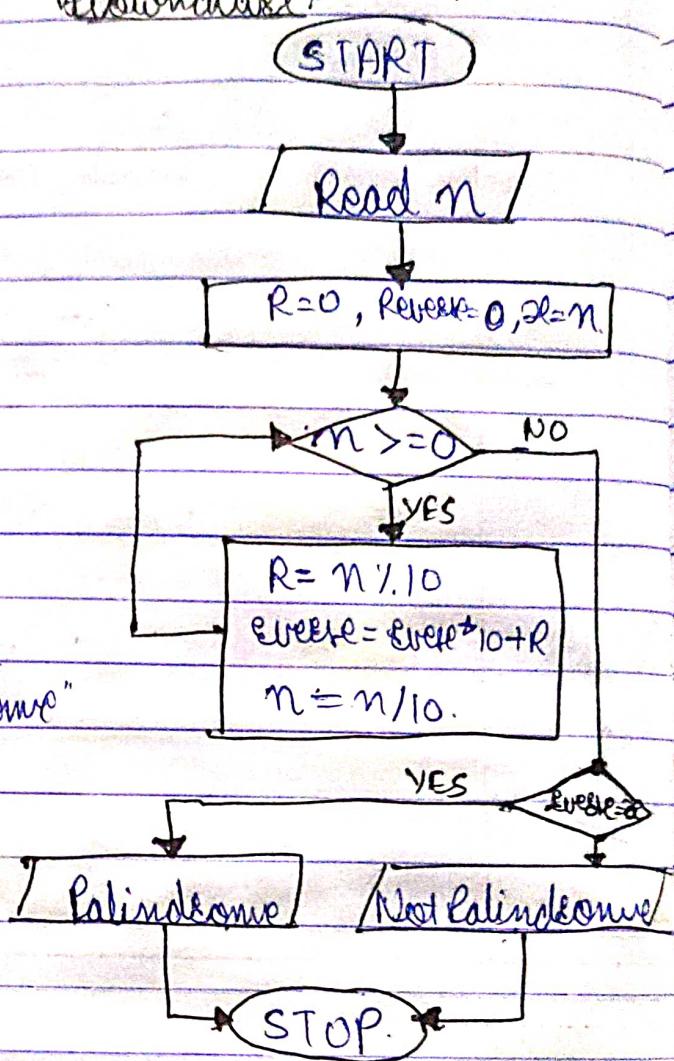
Step 5:- Repeat 4 till $n = 0$

Step 6:- if reverse = num

YES "Palindrome"

NO "Not Palindrome"

Step 7:- Stop.



Q18) Write a program to find all prime factors of a given program number.

→ Algorithm :-

Step 1 :- Start.

Step 2 :- Enter number as n ,

Step 3 :- Initialize local variable $i = 2$.

Step 4 :- Repeat while $i < n/2$:

 if $n \mod i == 0$

 if checkprime(i) == TRUE

 display print i

Step 5 :- increment i by 1.

Step 6 :- Stop.

(Q19) To print EVEN number series 2, 4, 6, 8, 10, 12, 14, 16 ...
Algorithm:

→ Step 1 :- Start
Step 2 :- Initialize $i = 1$, count = 0;
Step 3 :- Read num;
Step 4 :- while (count < num)
 Check $i \% 2 == 0$
Step 5 :- Print i.
Step 6 :- count++;
Step 7 :- i++
Step 8 :- Repeat step 4 to 7 till (count < num).
Step 9 :- Stop.

(Q20) To print ODD number series 1, 3, 5, 7, 9, 11, 13, 15 ...
Algorithm:-

→ Step 1 :- Start
Step 2 :- Initialize $i = 1$, count = 0.
Step 3 :- Read num.
Step 4 :- Check $i \% 2 != 0$
Step 5 :- Print i.
Step 6 :- count++;
Step 7 :- i++
Step 8 :- Repeat step 4 to 10 till
 (count < num)
Step 9 :- Stop.