

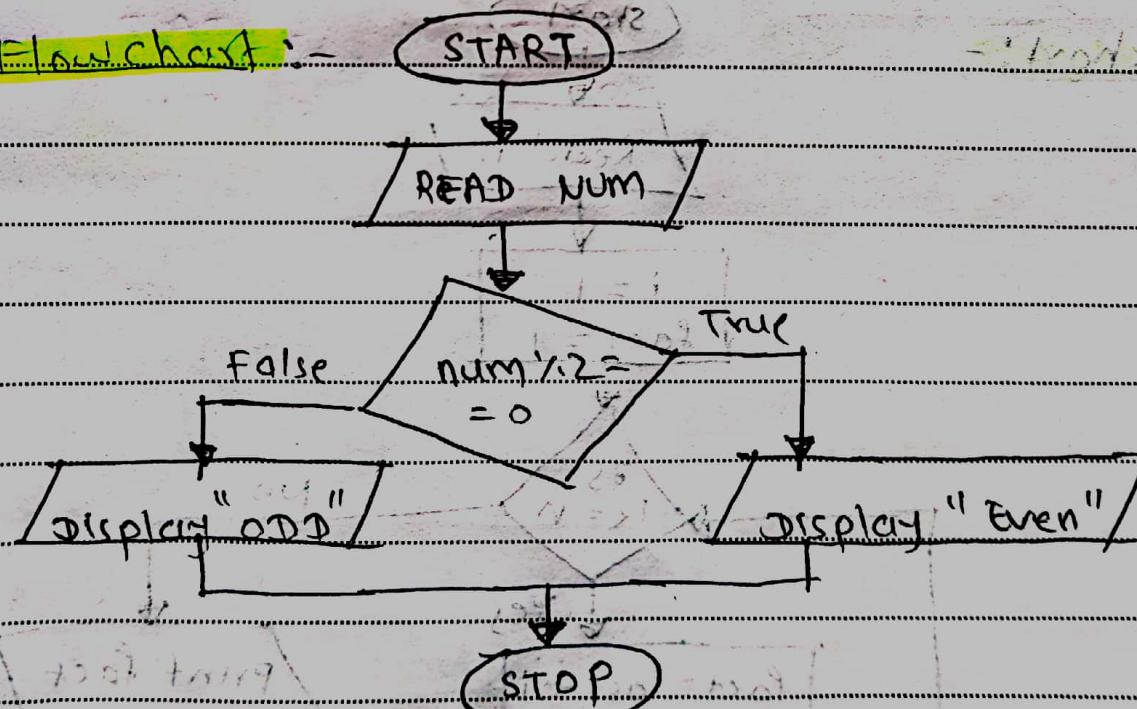
Assignment :- 1

1. check the given number is Even or ODD

Algorithm Steps :-

- ① Start.
- ② Read Num.
- ③ If Num divisible by 2, go to step - 4, else step - 5.
- ④ Display "Even" and STOP.
- ⑤ Display "ODD" and STOP.

Flowchart :-

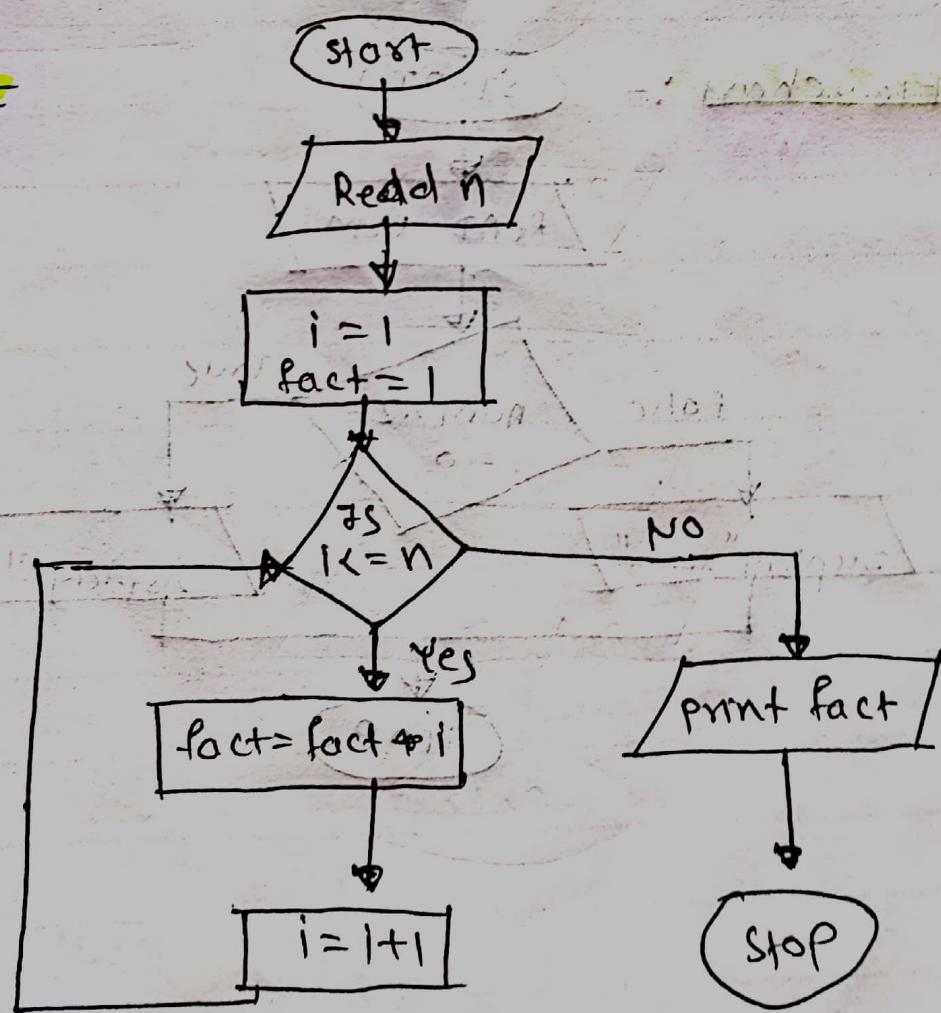


2. Write a Java P. to. find the factorial of given No..

*Algorithm:-

- ① start.
- ② Read a number n.
- ③ Initialize variables. ($i=1$, $fact=1$)
- ④ If $i \leq n$ go to step ⑤ otherwise go to step ⑧.
- ⑤ calculate. ($fact = fact * i$)
- ⑥ Increment the i by 1 ($i=i+1$) and go to step ④.
- ⑦ print fact.
- ⑧ stop.

*Flowchart:-



3. Find the factorial of a number using Recursion

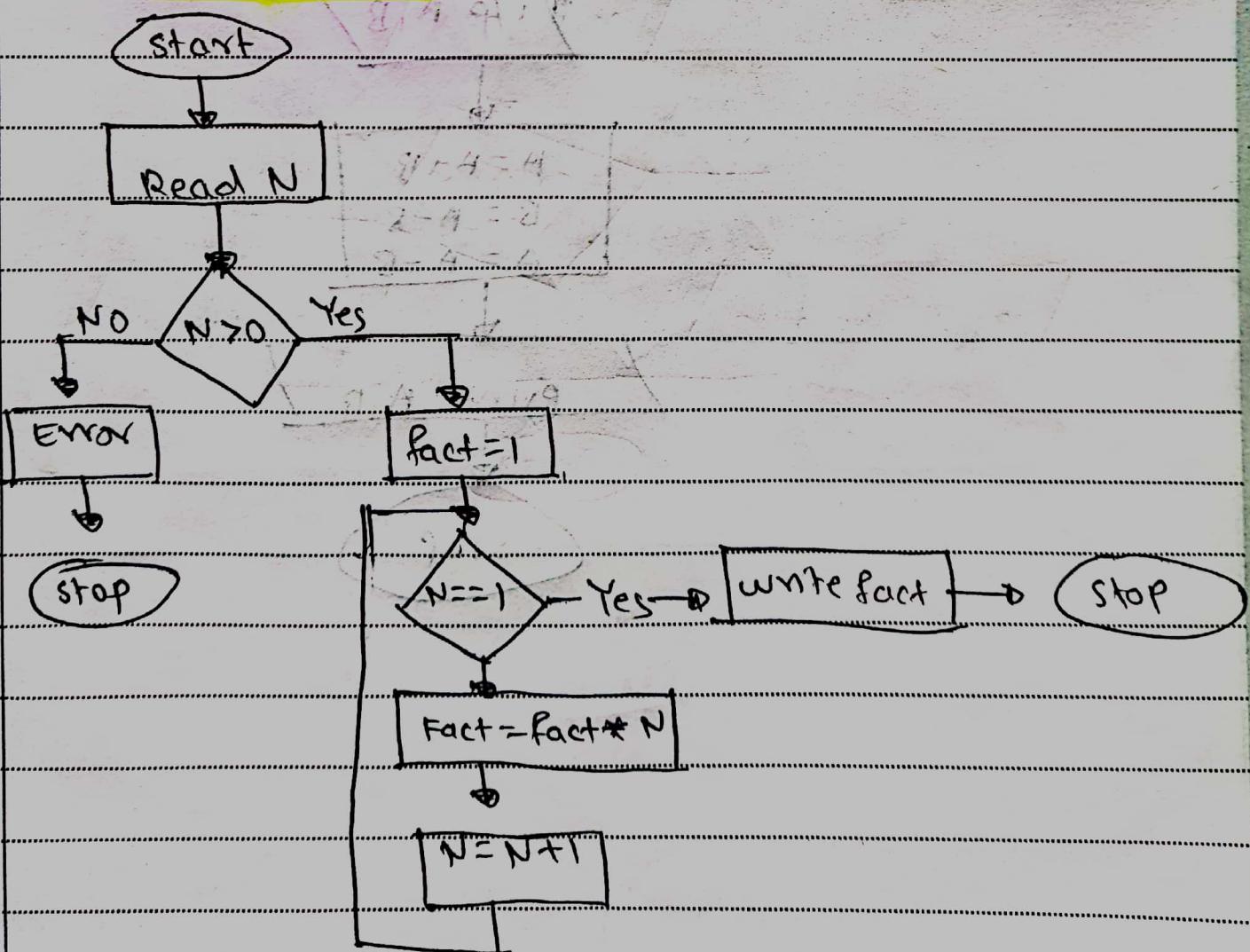
* Algorithm :-

- ① Start
- ② Read number n
- ③ call factorial (n)
- ④ Print factorial f.
- ⑤ Stop.

factorial (n)

- ① If $n=1$ then return 1
- ② Else $f = n * \text{factorial}(n-1)$
- ③ Return f.

* Flowchart :-

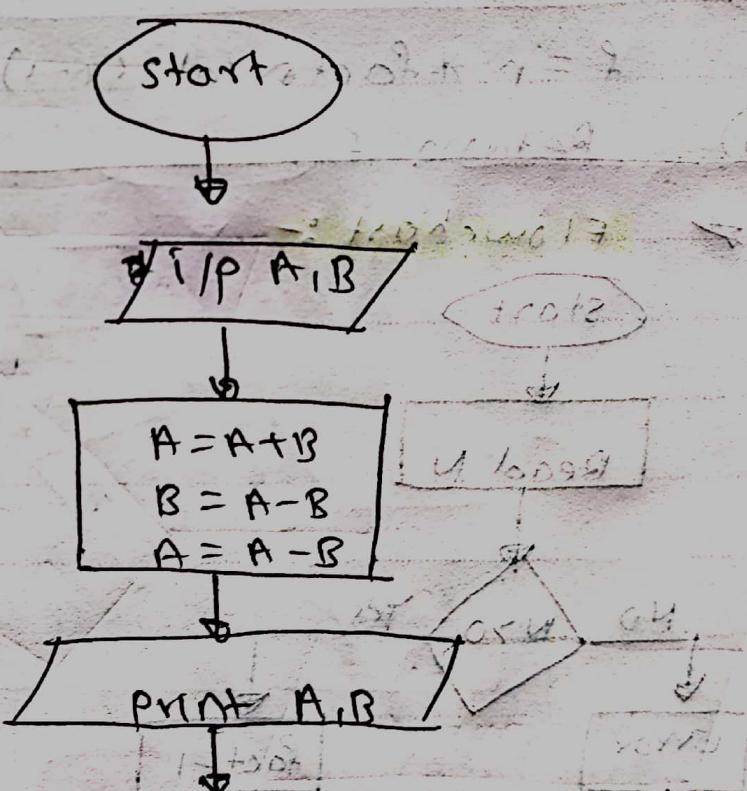


Q. Swap two numbers of a arr without using the third variable approach.

* Algorithm :-

- ① Start.
- ② Enter A,B.
- ③ Print A,B.
- ④ $A = A+B$.
- ⑤ $B = A-B$.
- ⑥ $A = A-B$
- ⑦ Print A,B.
- ⑧ End.

* Flowchart :-

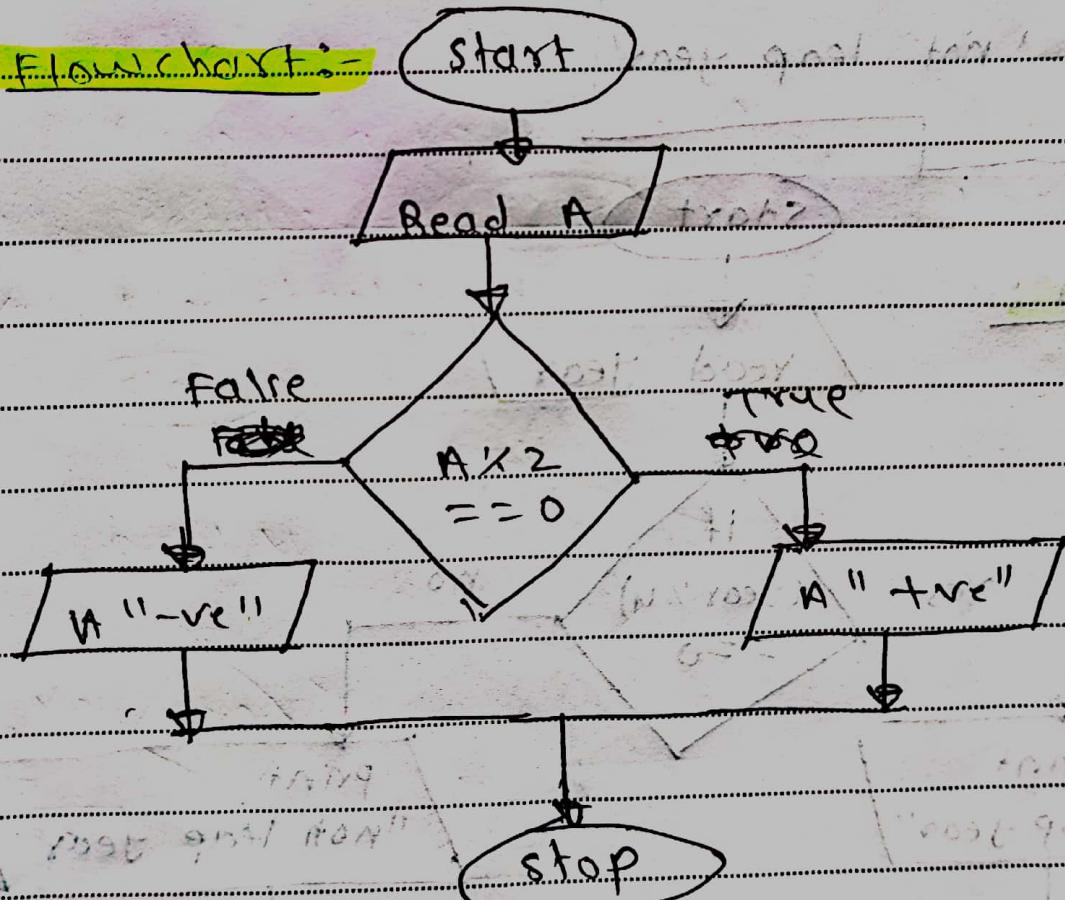


5. How to check whether the given number is positive or negative in Java?

* Algorithm:-

- ① Start.
- ② Read A.
- ③ If $A \times 2 = 0$, print A "positive".
- ④ Print A "negative".
- ⑤ End.

* Flowchart:-

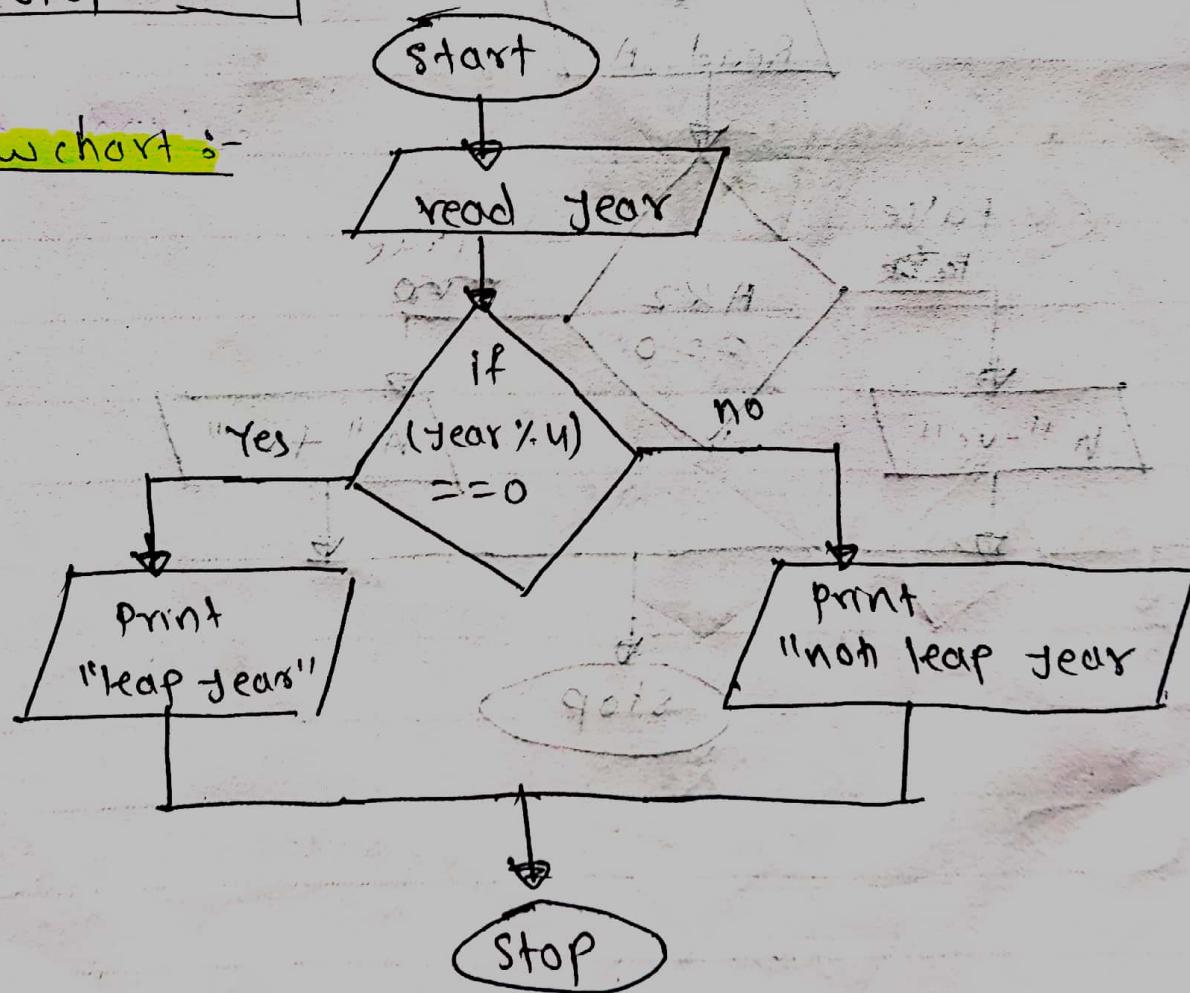


6. Write a Java program to find whether a given num. is leap year or NOT.

* Algorithm :-

- ①. start.
- ②. Read year
- ③. rem = year % 4.
- ④. if (rem == 0) then.
 print 'leap year'
else
 print 'not leap year.'
- ⑤. stop.

* Flowchart :-

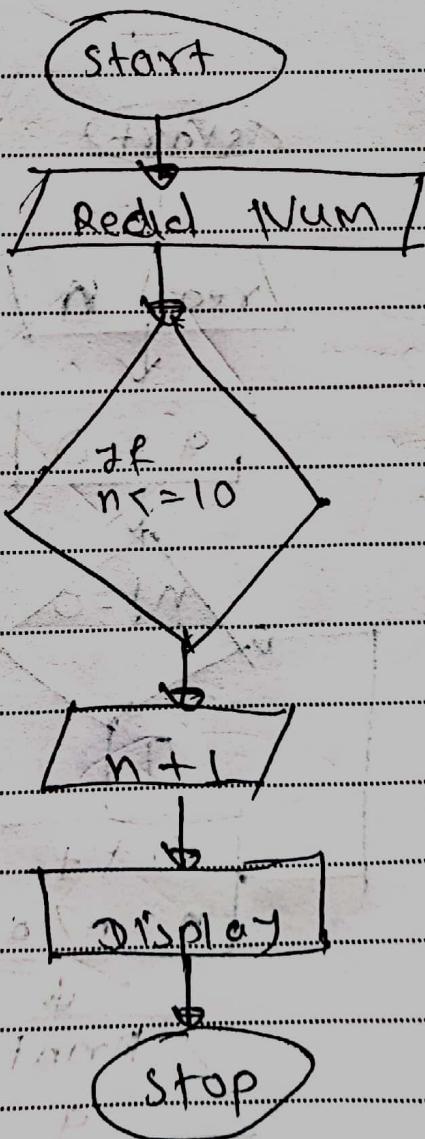


7. Java program to print 1 to 10 without using loop.

Algorithm :-

- ① Start.
- ② Read N.
- ③ If ($n \leq 10$) & then print ($n+1$)
- ④ Display
- ⑤ Stop.

Flowchart :-

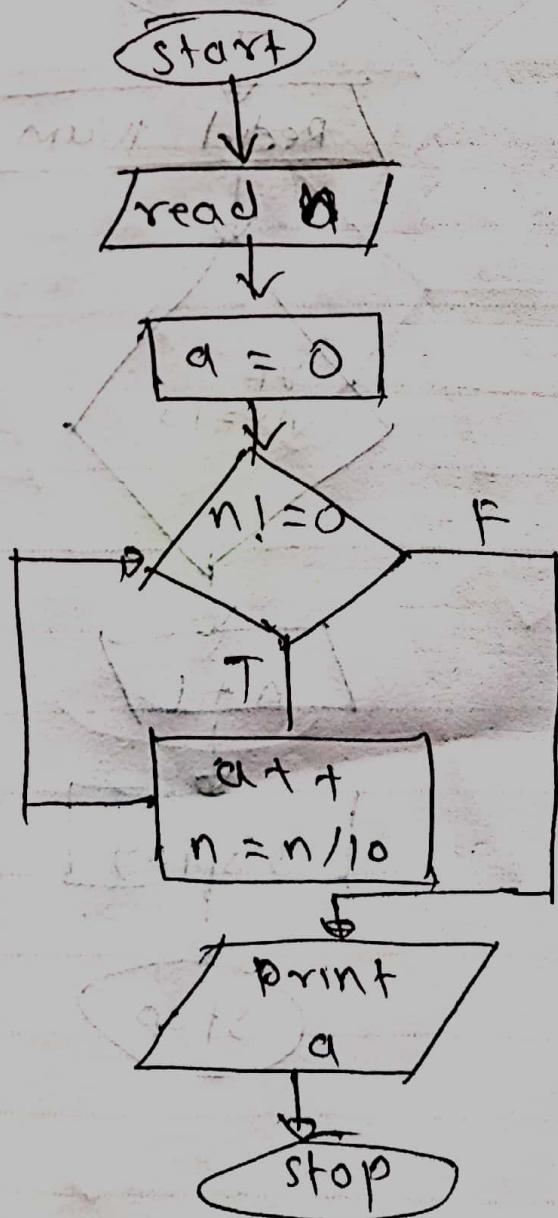


8. Print the Digits of a Given number.

* Algorithm :-

- ① start
- ② read n
- ③ a = 0;
- ④ while ($n \neq 0$)
 { a++
 $n = n / 10$
 }
- ⑤ display a;
- ⑥ stop;

* Flowchart :-

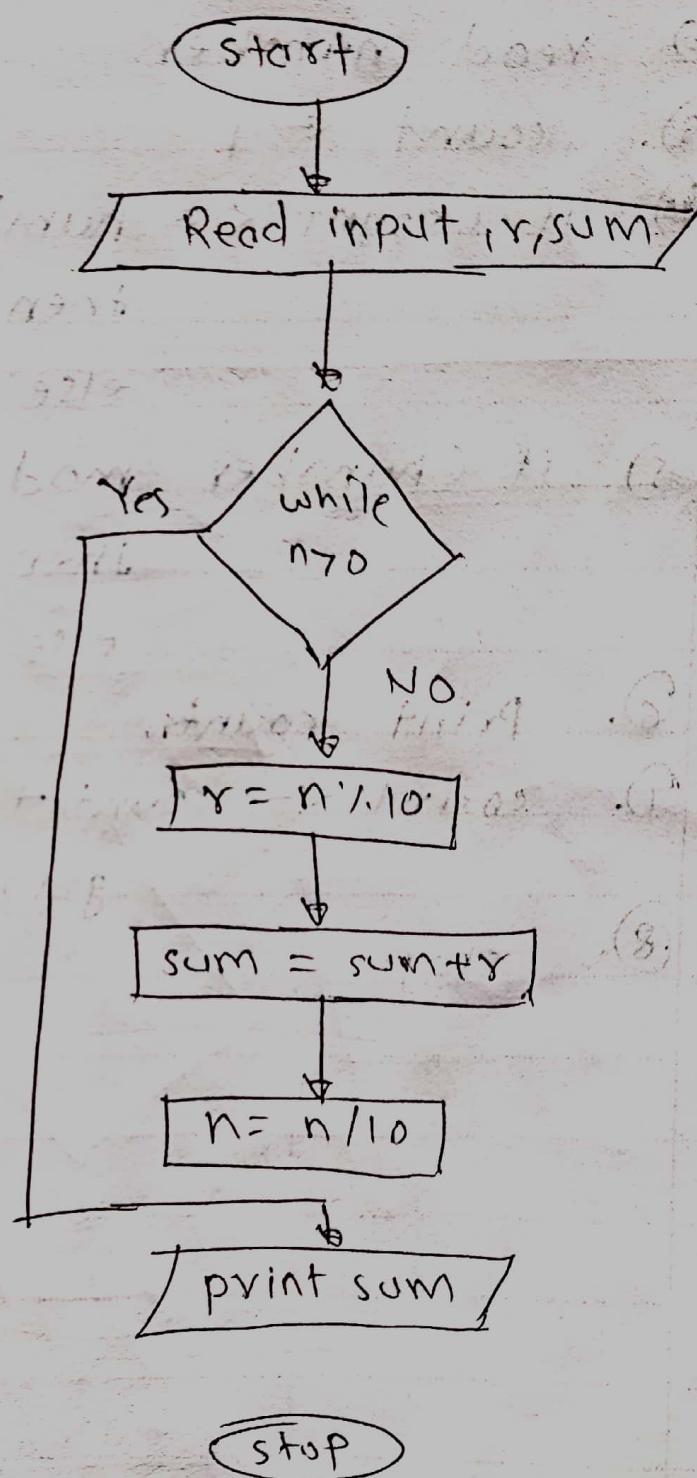


10. Find sum of digits of given numbers.

* Algorithm :-

- start
- read input
- three variables
input, r, sum
- while ($n > 0$)
 $r = n \% 10$
 $sum = sum + r$
 $n = n / 10$
- print sum
- stop.

* Flowchart :-

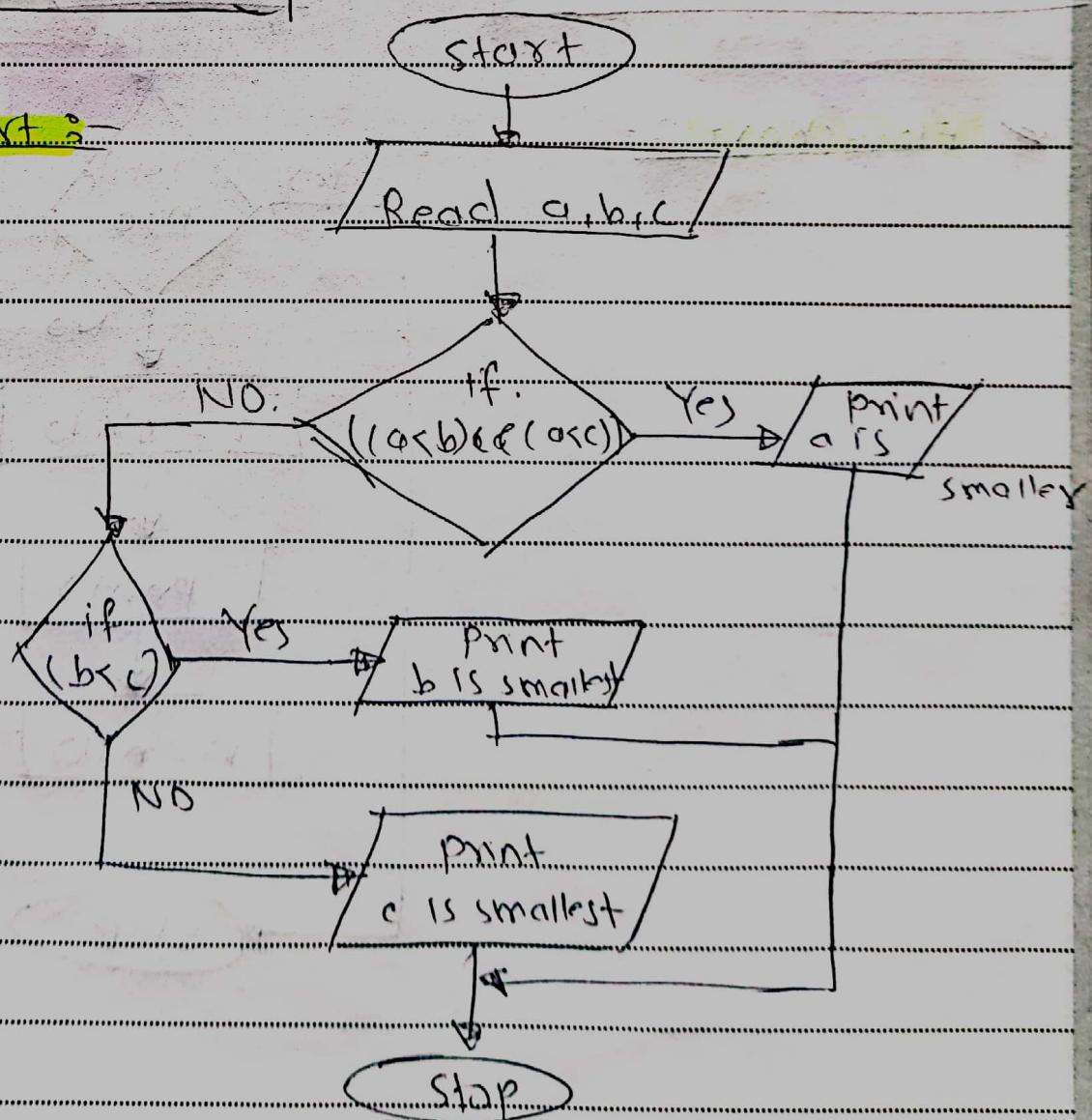


11. Find smallest of 3 nos. a,b,c.

* Algorithm :-

- start
- read a,b,c
- if $a < b \& a < c$
then a is smallest
- if $b < c$
then b is smallest
- else c is smallest
- stop

* Flowchart :-

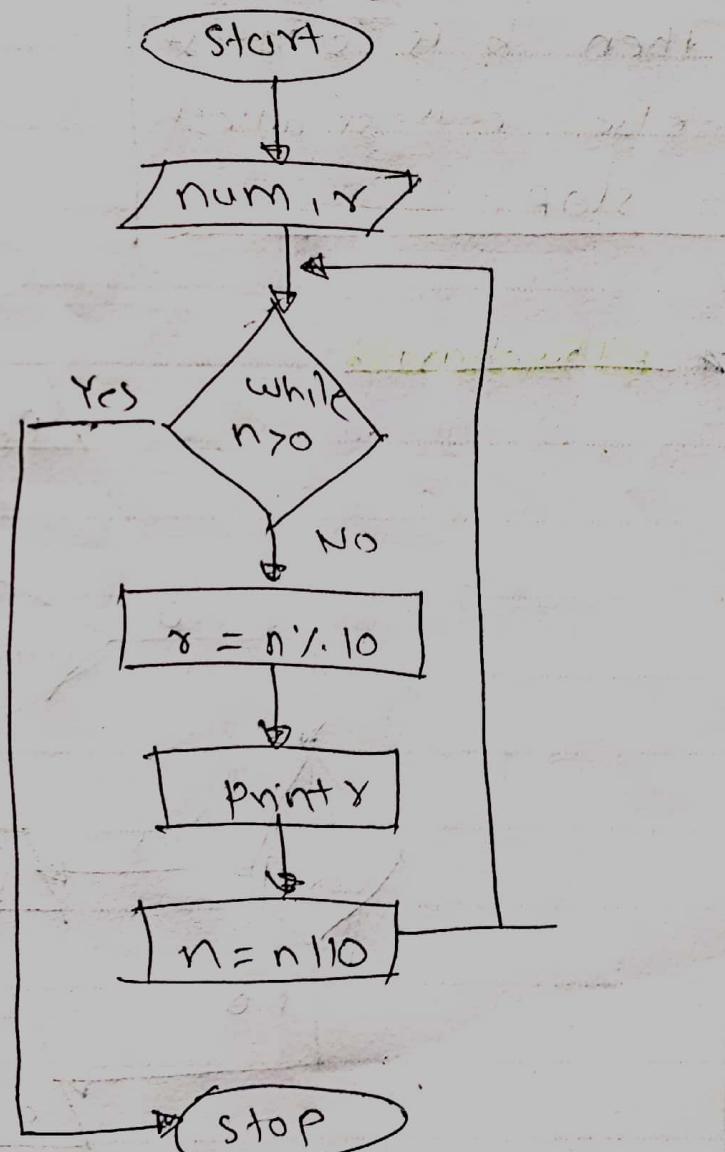


13). Reverse the given number

* Algorithm :-

- Start
- Read num
- declare num & r
- while ($n \geq 0$)
 $r = n \% 10$
print r
 $n = n / 10$
- stop.

* Flowchart :-



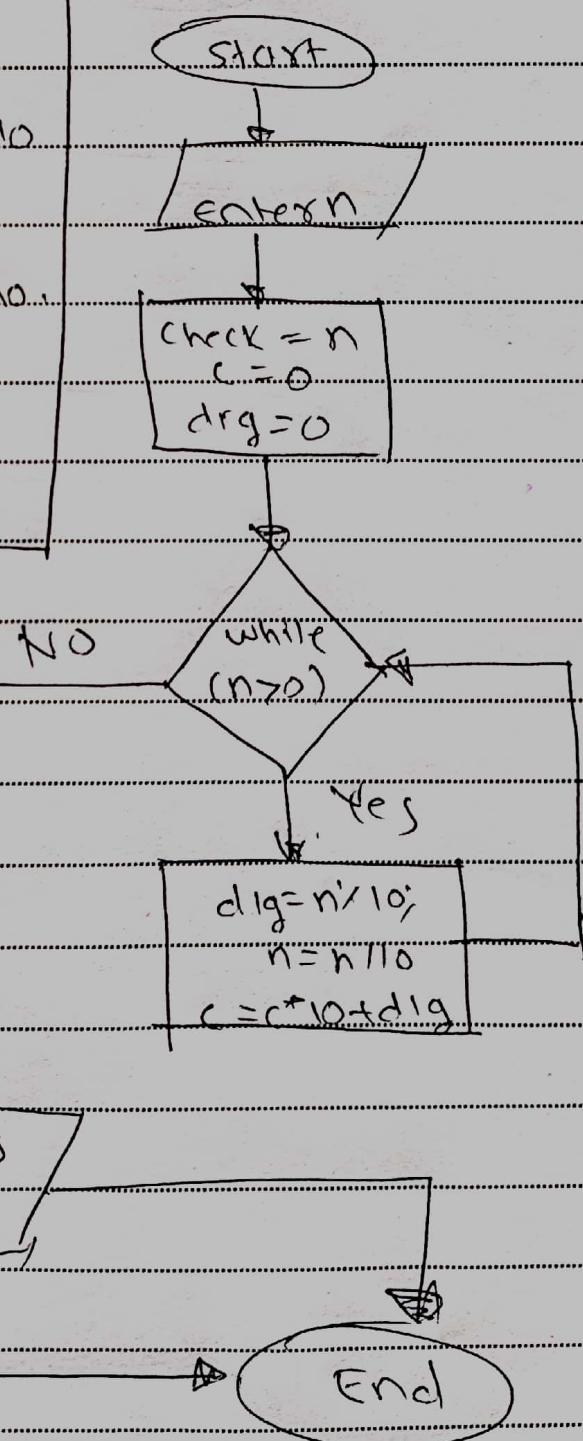
17

Given Number is a Palindrome or NOT

* Algorithm :-

- (1) Start
- (2) Enter n
- (3) check = n, c = 0, dig = 0
- (4) Enter while ($n > 0$), if condition true goto 5
or else goto 6.
- (5) $dig = n \% 10$, $n = n / 10$, $c = c * 10 + dig$
+ goto 4.
- (6) if (check == c), print no.
is Palindrome
- (7) else, print Is not
palindrome

* Flowchart :-

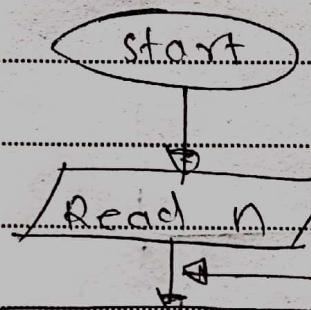


19.

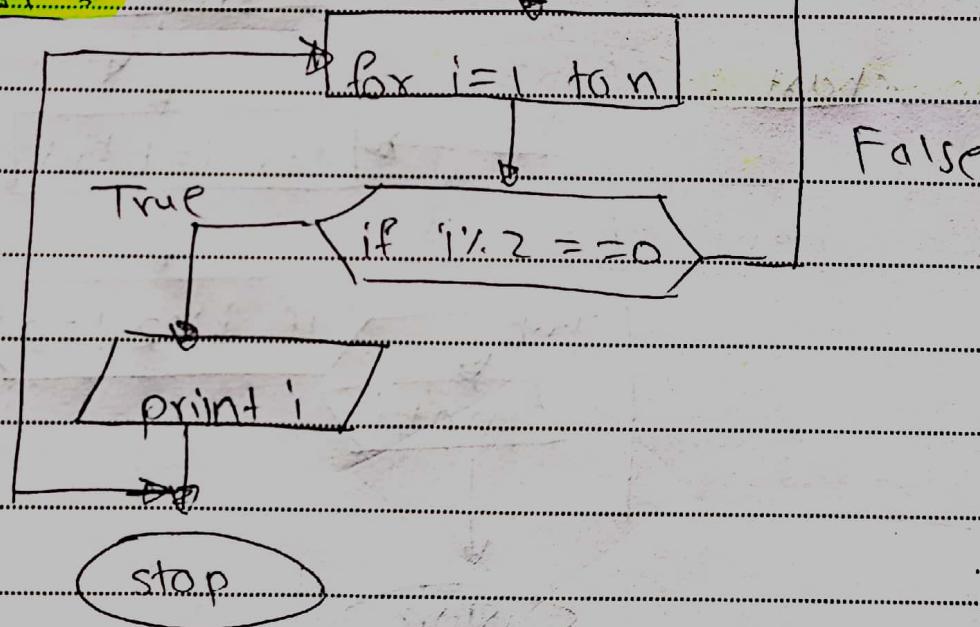
print even nos. series 2, 4, 6, ...

* Algorithm :-

- start
- input max.no. n
- for i=1 to n
- if $i \% 2 == 0$
- print i
- stop



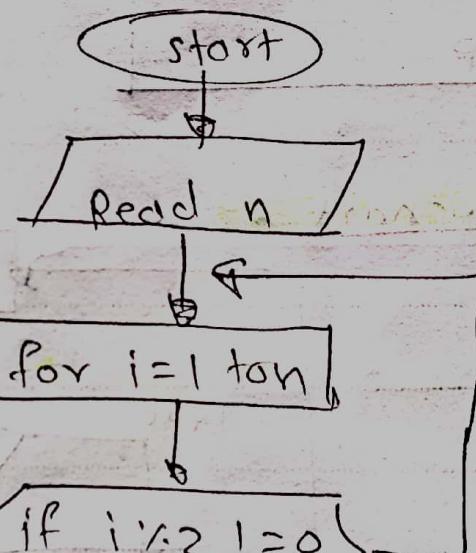
* Flowchart :-



20. Print odd nos. series 1, 3, 5, 7, ...

* Algorithm :-

- start
- input most.no. n
- for i=1 to n
if $i \div 2 \neq 0$
print i
- stop.



* Flowchart :-

