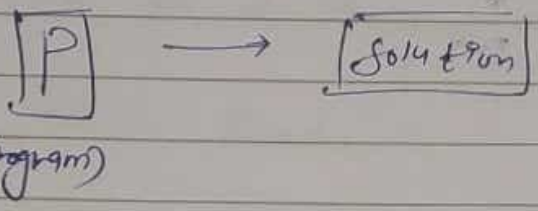
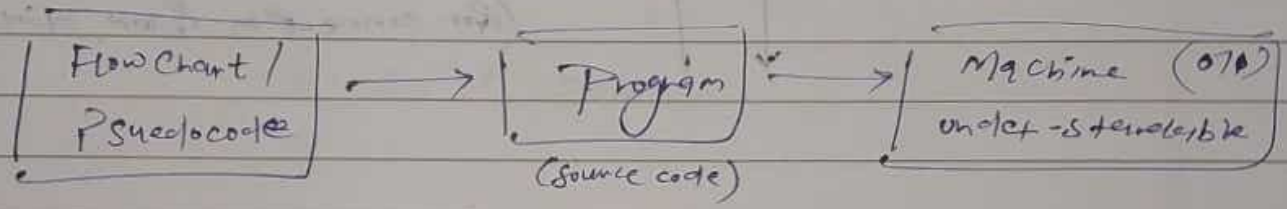
DSA 1 LB.

#LECTURE-1

Intro to Programming & Flowchart





Flowchart — Diagramatic approach (representation of a)

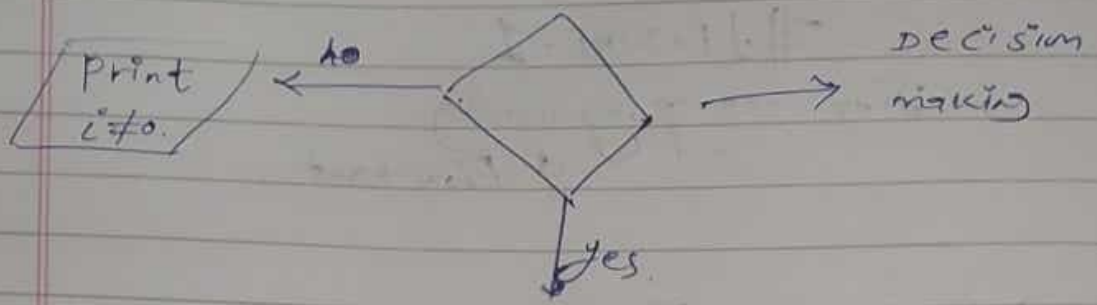
Components :-

1. Start/End → Termination
   - Start
   - End

2. Print "Hello" / Read a → Parallelogram
   - Input/output

3. i = i+1 → process

4



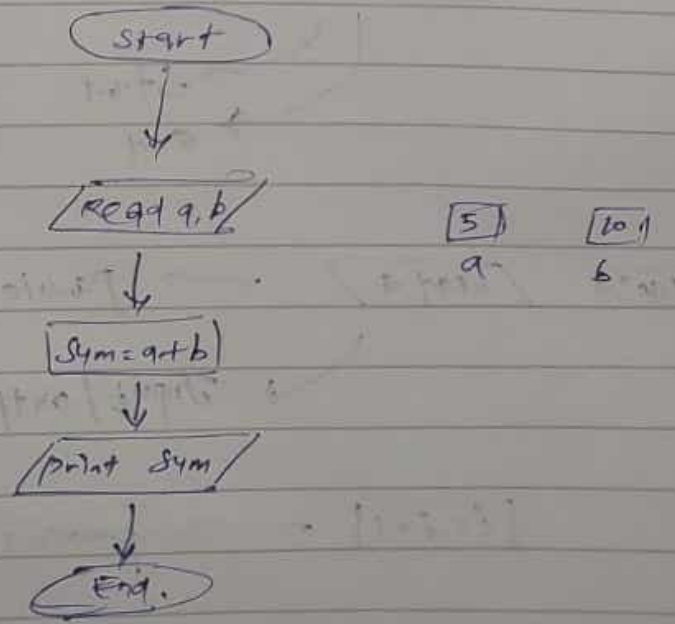
5


arrows
(for connection, flow, direction)

6

Connector.

◦ Sum of 2 no.



Pseudocode

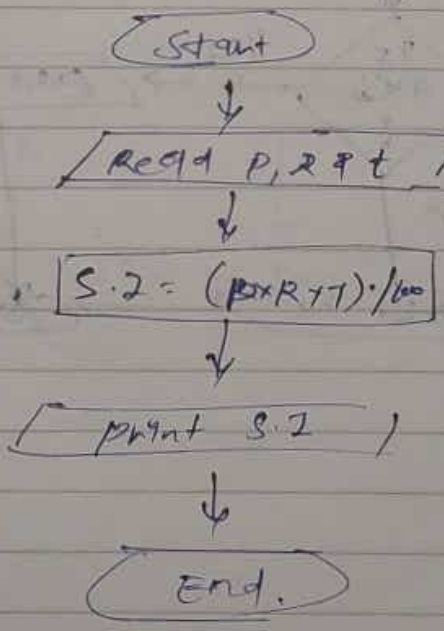
↳ Way of Representing Logic (Logic) — Generic

⇒ Sum of 2 no.

1. Read 2 no, a & b
2. Sum = a+b
3. print sum

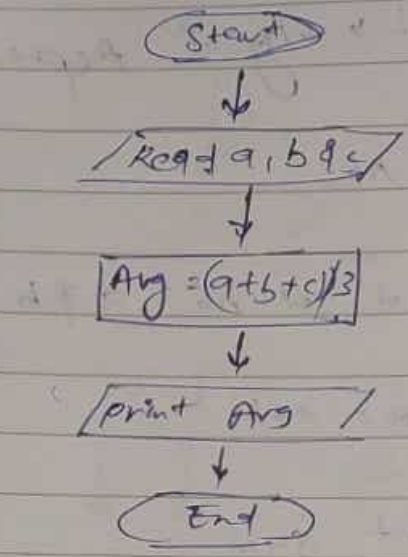
⇒ Simple interest Flowchart

$$S.I = \frac{P \times R \times T}{100}$$

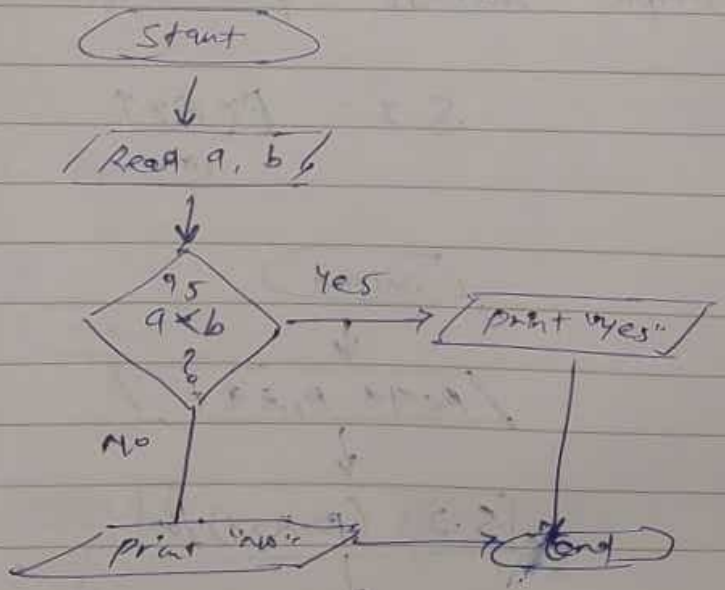


[1] P [2] R [3] T

→ Avg of 3 no.



=> a<b => "yes" or "no"

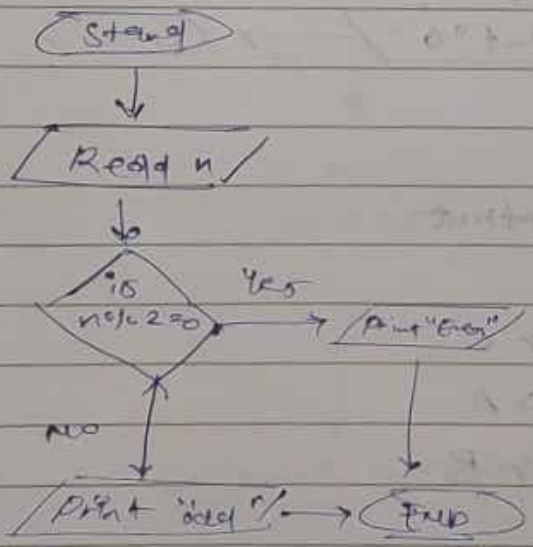


1. Read a & b

2. if a < b
   print "Yes"
   else
   print "No"

3. END.

⇒ Odd or Even



n = 3.

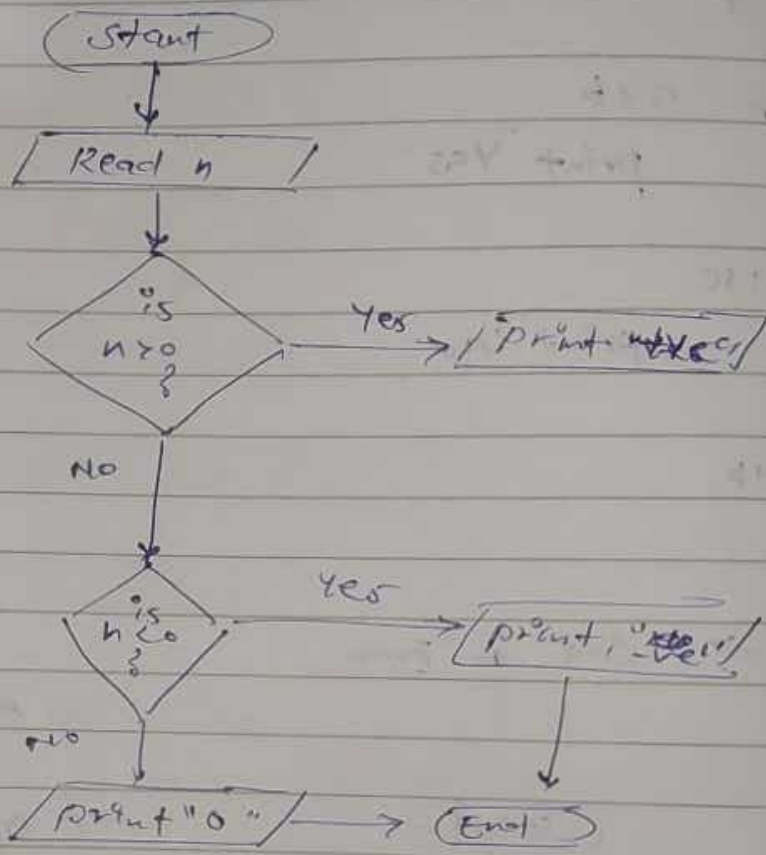
% Operator → Rem

5 % 3 → 2

6 % 2 → 0

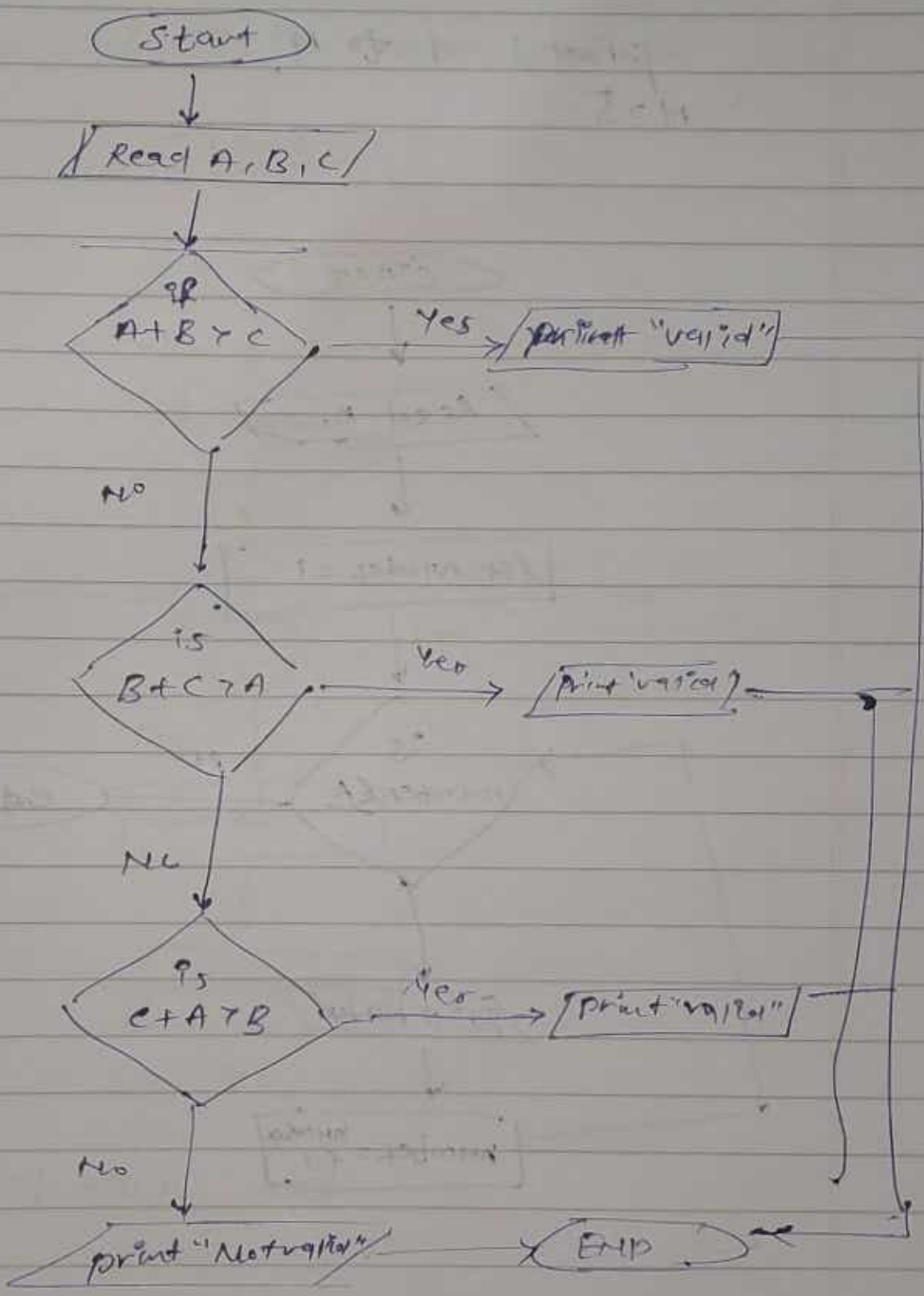
n % 2 = 0 → even

n % 2 = 1 → odd



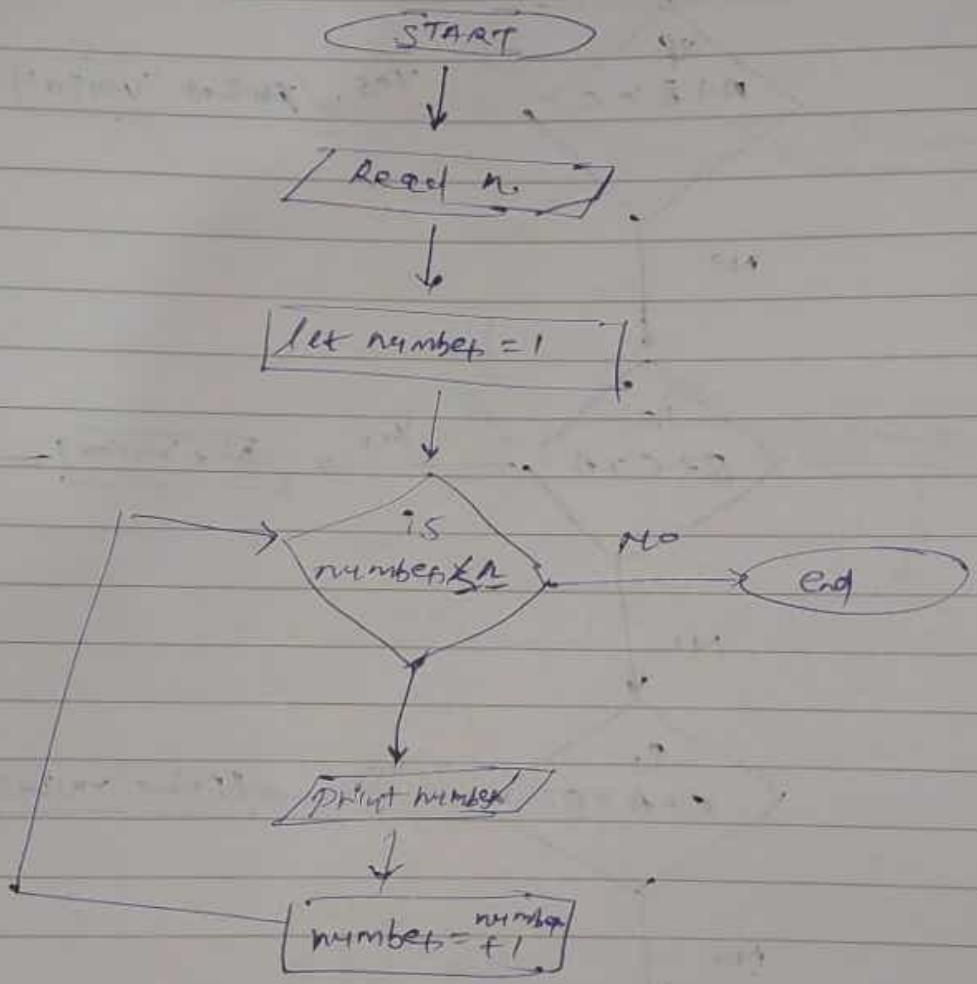
Valid triangle or not.

$$A + B > C$$
$$B + C > A$$
$$C + A > B.$$

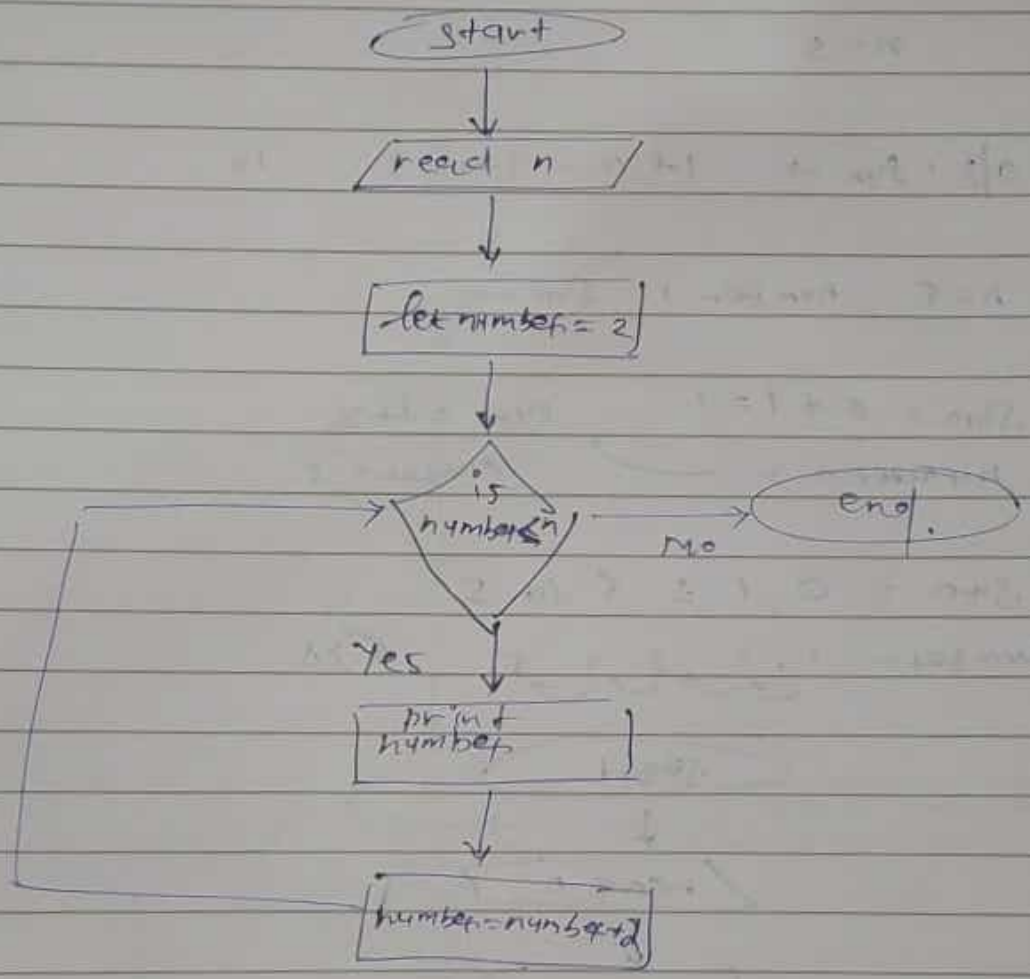

Start
Read A, B, C
If A+B > C
Yes
print "valid"
No
is B+C > A
Yes
print "valid"
No
is C+A > B
Yes
print "valid"
No
print "Not valid"
END

Print 1 to N.

N = 5.

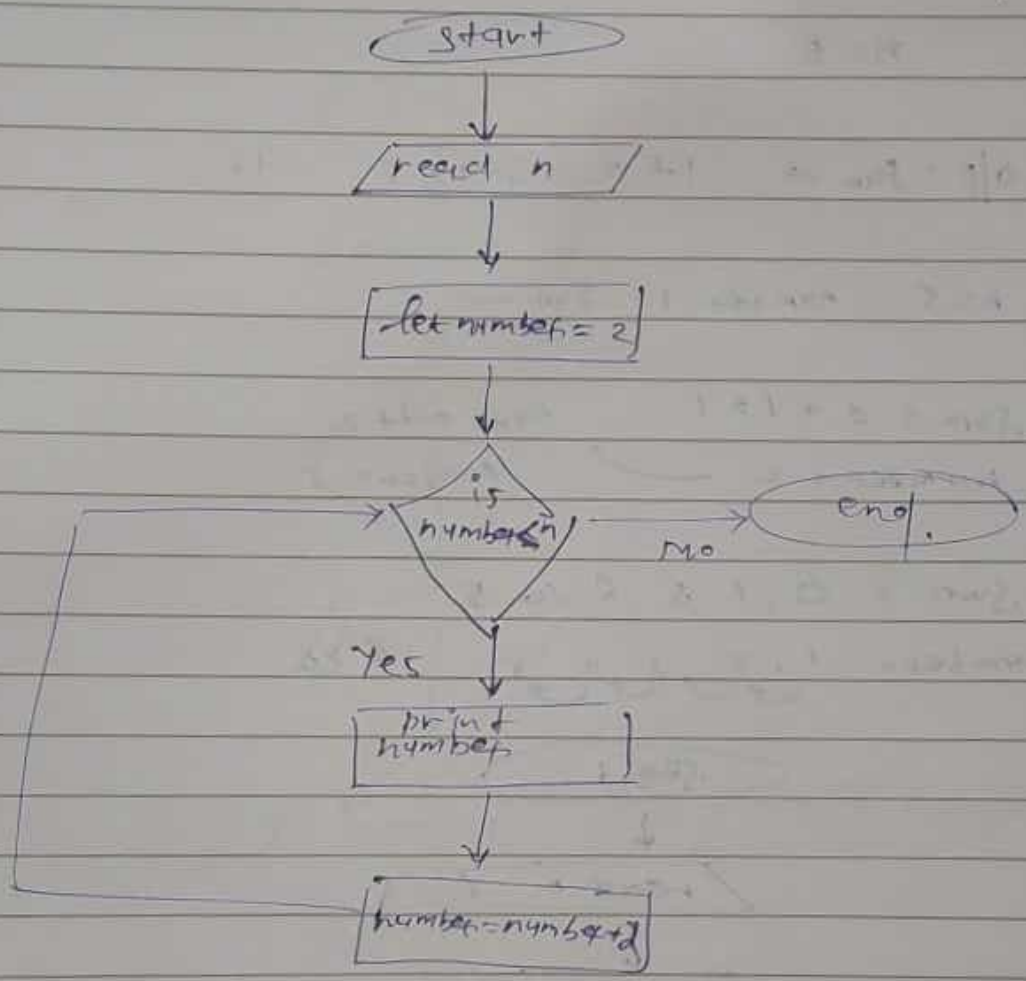


9. 1 to n even even sq. print



1 to n = odd

Q. 1 to n: even even no. print



1 to n = odd

q. find sum 1 to N (in digits)

n = 5

o/p = sum → 1 + 2 + 3 + 4 + 5 = 15

n = 5, number 1, sum = 0

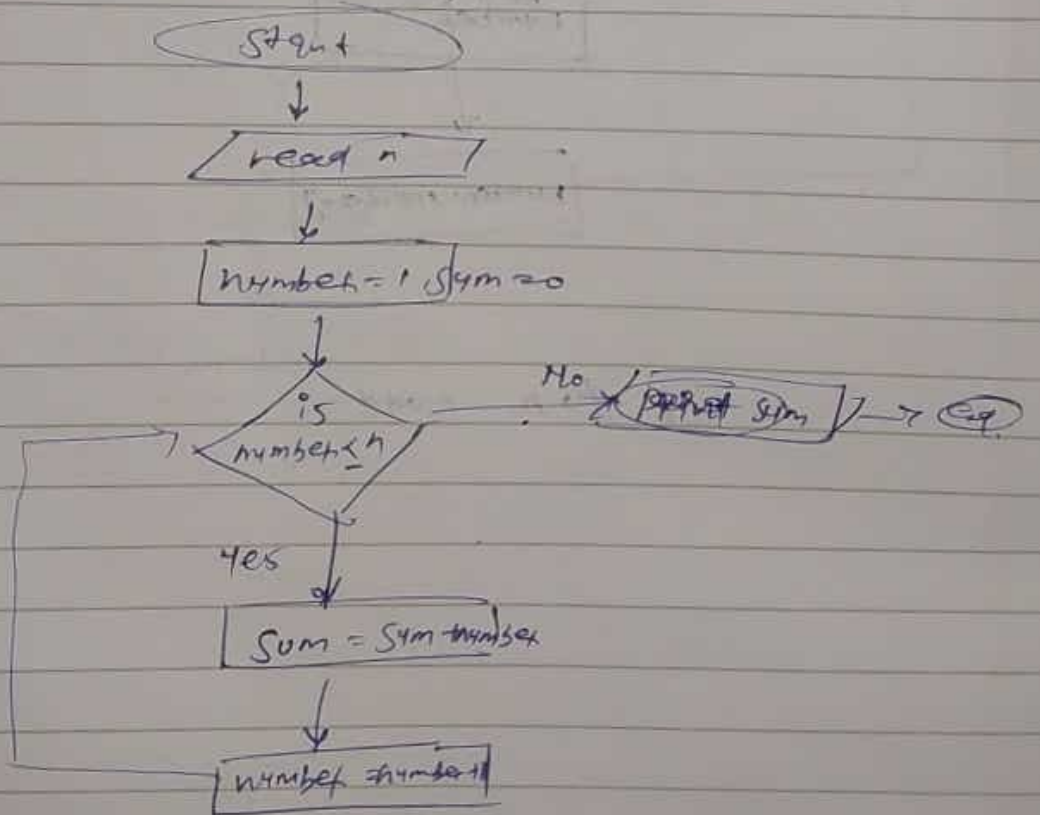
sum = 0 + 1 = 1

number = 2 → sum = 1 + 2

number = 3

sum = 0, 1, 3, 6, 10, 15

number = 1, 2, 3, 4, 5, 6, 7, 8



q find factorial

5! = 5 × 4 × 3 × 2 × 1

1 × 2 × 3 × 4 × 5

Q Check prime or not

n ⟶ prime

2 . . . . (n-1) ⟶ n%i == 0