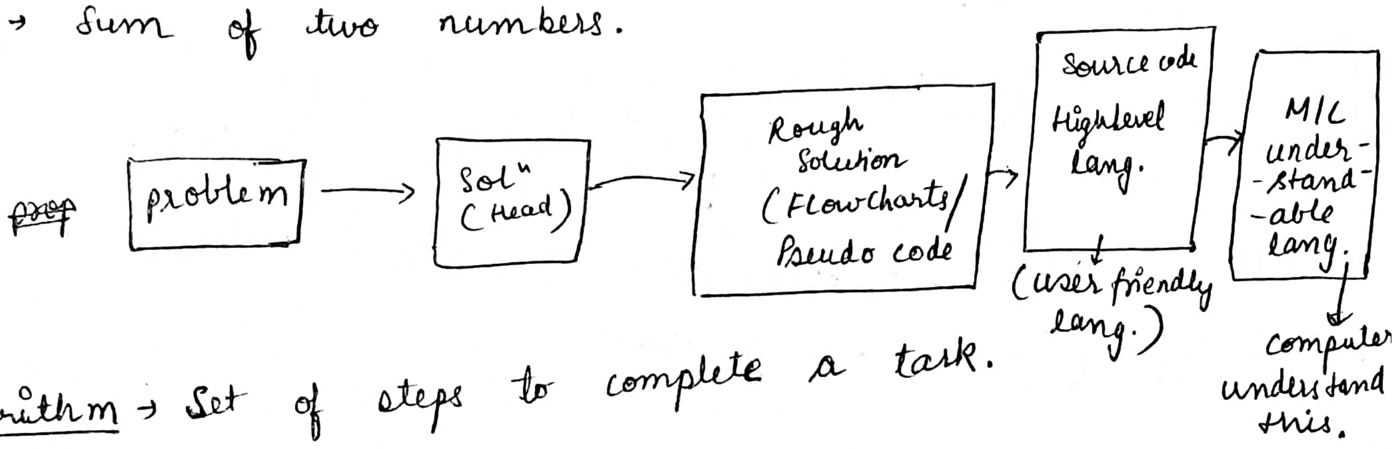
Lec-01 Programming Fundamentals 27/Jan

Thought process to solve a problem -

→ Understand the problem
→ i/p values
→ Find a approach

Eg→ Sum of two numbers.



~~prop~~
problem
Solⁿ (Head)
Rough Solution (Flowcharts/ Pseudo code
Source code
Highlevel Lang.
(user friendly lang.)
M/C under-stand-able lang.
computer understand this.

Algorithm → Set of steps to complete a task.

⊙ Using computer to solve a problem →

Suppose we want to find if a no. is prime or not.
A no. is prime if it only divided by 1 a itself (completely).

eg = 13.
let's check →

13/1 = 13
13/2 = 6.5
13/3 = 4.33
13/4 = 3.25
13/5 = 2.6
- - -
- - -
13/13 = 1

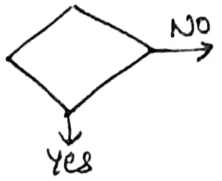
13 is only divided by 1 and 13.
So it is a prime no.

This is manual.
But what if the number is too greater. Are we going to do this?
So that's why we use computer.

⊙ Flowchart → Flowchart is a graphical representation of an algorithm.

Components →

- Terminator → ⬭ , used to show the start or end of an algo.
- Input/Output → ▱ , used for input and output.
- Process block → ▭ , used for calculation and initalization
- Decision Making Block →



, Decision are made (Yes or NO).

- Arrows → →, ←, ↑, ↓ , arrows are used to show the flow.
- Connector → ○ , it shows functions.

① Pseudo Code → Generic way of representing your approach/ algo in textual form.

Fake (नकली)

→ Not understandable by computers.

Eg→ Add 2 numbers.

```
① Start Program
② Enter two numbers , a & b
③ add the numbers together
④ print sum
⑤ End program.
```

There is not only single ~~an~~ pseudo code of any problem.

- Subtract two number.

```
① Read a, b
② diff = a - b
③ print diff.
```

- Print the multiplication of two numbers.

```
① Read a, b
② mul = a*b
③ print mul
```

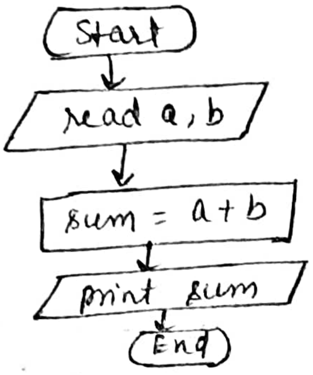
~~④~~

- Find avg of two numbers.

```
① Read a, b
② avg = (a+b)/2.
③ print avg.
```

Let's practice flowcharts ->

① Add 2 numbers.



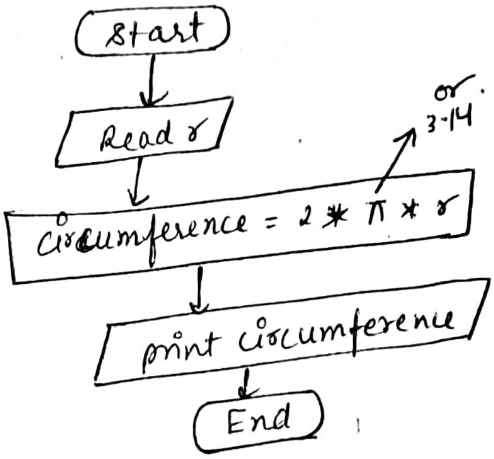
pseudo code-

```
→ read a, b
→ sum = a+b
→ print sum.
```

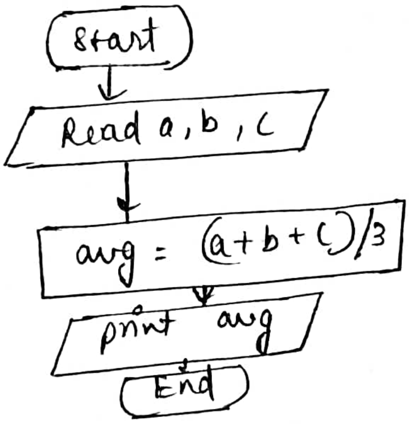
② circumference of a circle - $2\pi r$

pseudo code→

```
→ read r
→ cir =: 2* 3.14 * r
→ print cir
```



③ Avg of three numbers.



pseudo code-

```
→ read a, b, c
→ avg = (a+b+c)/2
→ print avg.
```

④ Check a number is even or odd.

% → remainder operator.

2%1 = 0

3% 2 = 1 ←

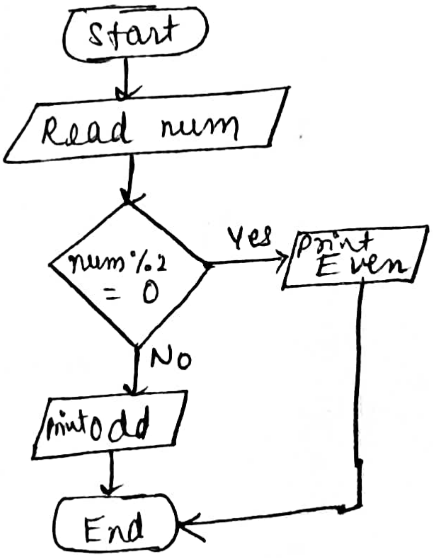
2)3 L1
2
1 =

n%2 = 0 → even

n%2 = 1 → odd

pseudo code→

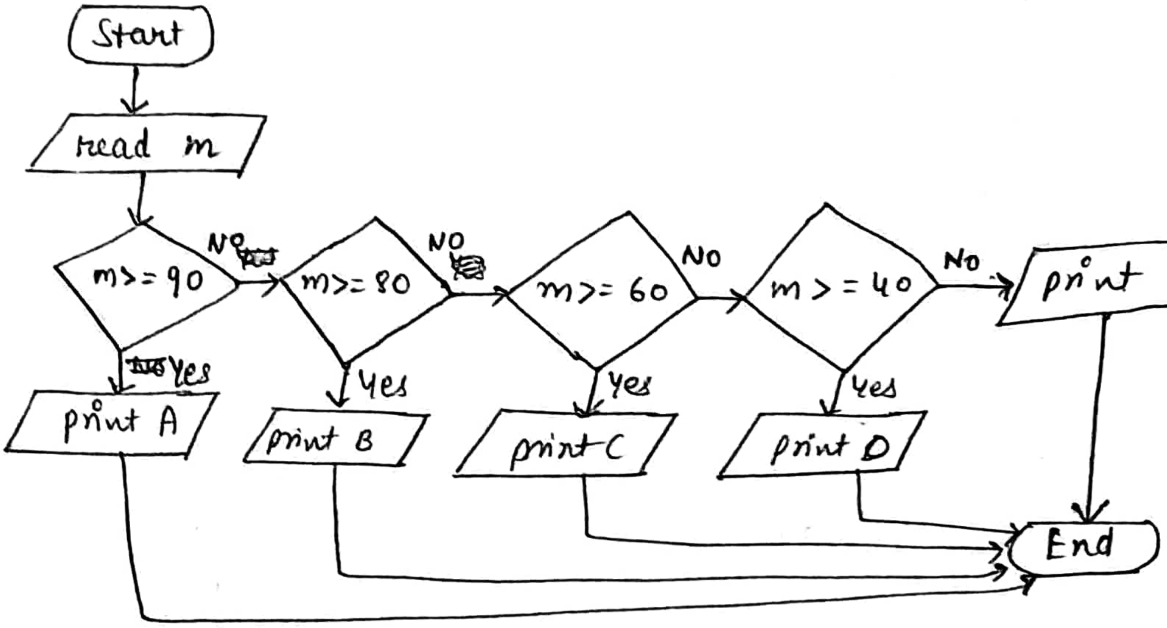
```
→ Read num
→ if num %2 = 0
      print even
  else print odd
```



- Student & Grade Flowchart

marks >=90 → A , >= 80 → B , >= 60 → C, >=40 → D

< 40 → F



Pseudo code -

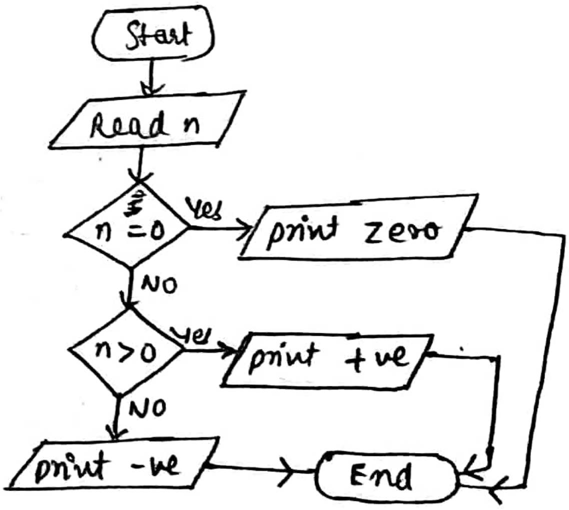
```
Read m
if m >= 90
        print A
else if m >= 80
        print B
else if m >= 60
        print C
else if m >= 40
        print D
else    print F.
```

- Check number is +ve, -ve or 0.

+ve → >0
-ve → <0
0 → =0

Pseudo Code -

```
Read n
if n = 0
    print zero
else if n > 0
    print +ve
else
    print -ve
```

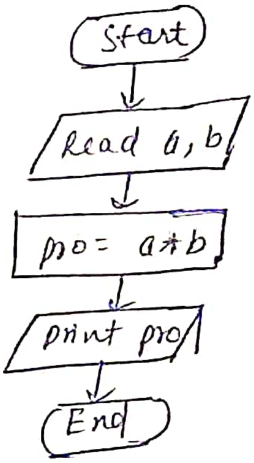


Problem Solving

- **Multiply 2 no. by taking input.**

pseudo code -

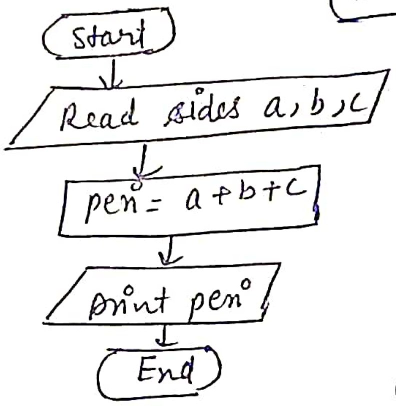
```
→ read a, b
→ pro = a * b
→ print pro.
```



- **perimeter of a triangle.**

pseudo code

```
→ read a, b, c
→ peri = a+b+c
→ print peri
```

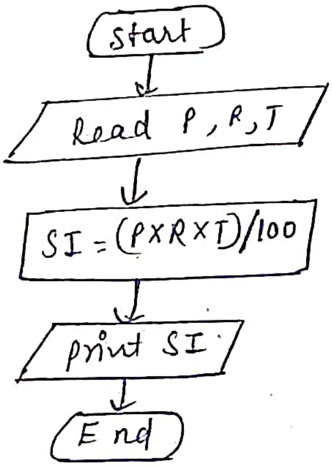


- **Find Simple Interest**

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

pseudo code -

```
→ read P, R, T
→ SI = (P × R × T) / 100
→ print SI.
```

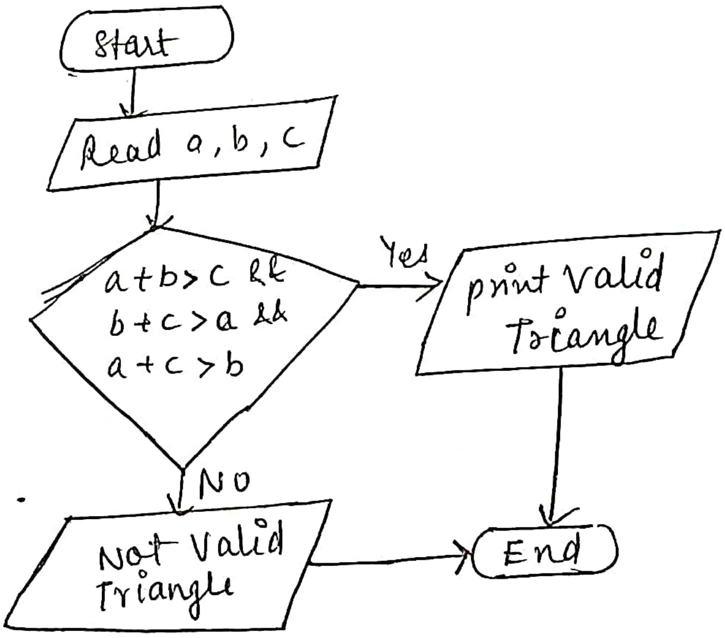


- Valid Triangle or Not

The condition of valid triangle is that the sum of any two sides is always be greater than the third side.

pseudo code -

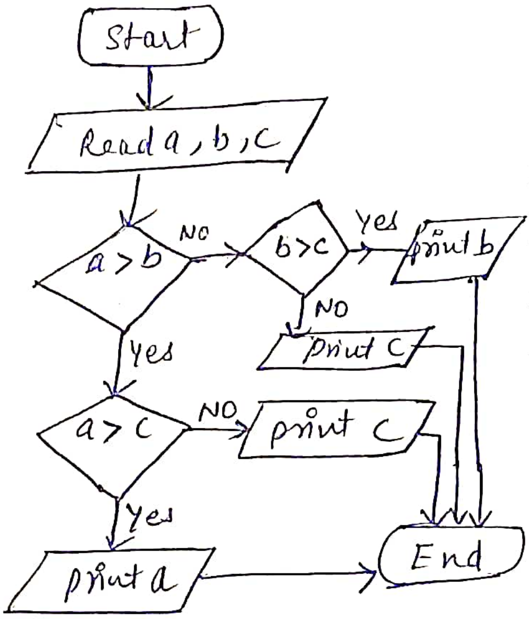
```
→ read a, b, c
→ if a+b > c &&
     b+c > a &&
     a+c > b
     print valid Triangle
→ else
     print Not a
           valid Triangle.
```



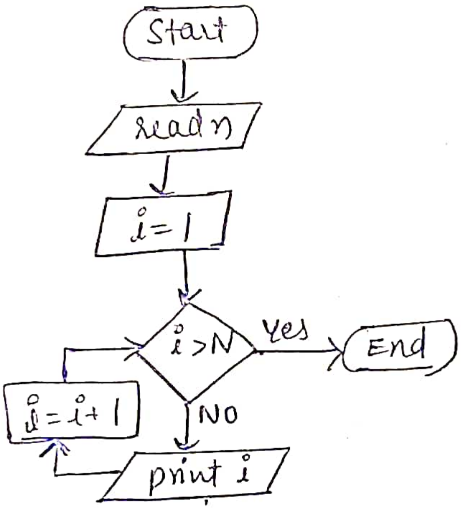
- Print greatest no. of 3 numbers.

Pseudo code

```
→ read a, b, c
→ if a > b
      if a > c
          print a
      else
          print c
→ else
      if b > c
          print b
      else
          print c
```



- Print counting from 1 to N →



pseudo code -

```
① → read n
② → i = 1
③ → if i > n
        then exit.
④ → else
        print i
        i = i + 1
        go to step 3
```

n = 4

i = 1, n = 4

1 > 4 → F

print 1 ✓

i = 2,

2 > 4 → F

print 2

i = 3

3 > 4 → F

print 3

i = 4

4 > 4 → F

print 4

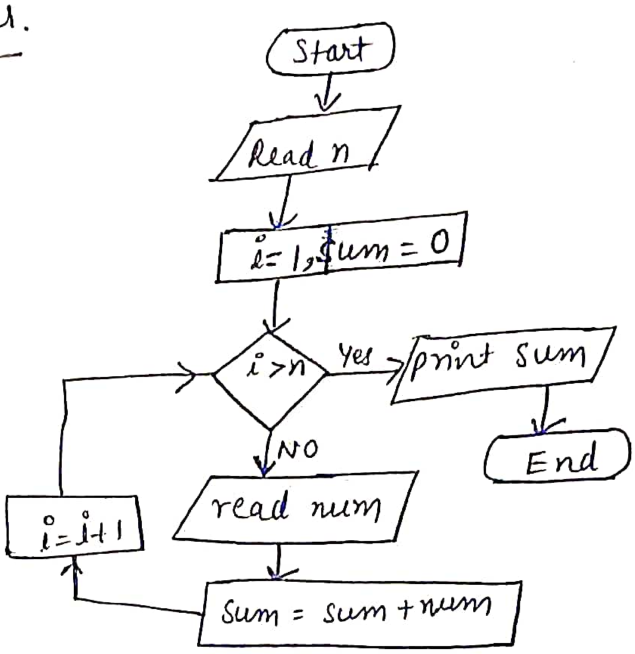
i = 5

5 > 4 → T → **End.**

- Add n numbers from user.

pseudo code –

```
① → Read n
② → i=1, sum=0
③ → if i>n
        print sum
        Exit
④ → else
        read num
        sum = sum + num
        i = i + 1
        go to step ③
```



dry run –
n=4

i=1, sum=0, 1>4 → NO → ~~print~~ read num
num = 1 → sum = 0+1 = 1

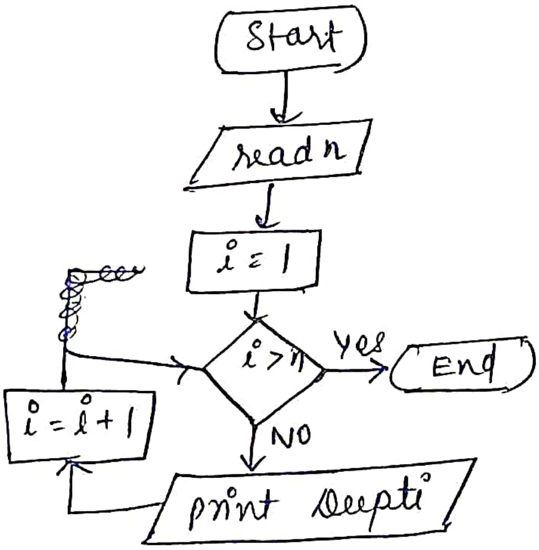
i=2, sum=1, 1>4 → NO → read num
num = 2 → sum = 1+2 = 3

i=3, sum=3, 3>4 → NO → read num
num = 5 → sum = 3+5 = 8

i=4, sum=8, 4>4 → NO → read num
num = 8 → sum = 8+8 = 16

i=5, sum=16, 5>4 → yes → End of loop
print sum.

- Print your name N Times.



pseudo code –

```
① → read n
② → i=1
③ → if i>n
        exit
④ → else
        print Deepti
        i= i+1
        go to step ③
```

n=3, i=1 1>3 → NO → print Deepti, i = 1+1 = 2

n=3, i=2 2>3 → NO → print Deepti, i = 2+1 = 3

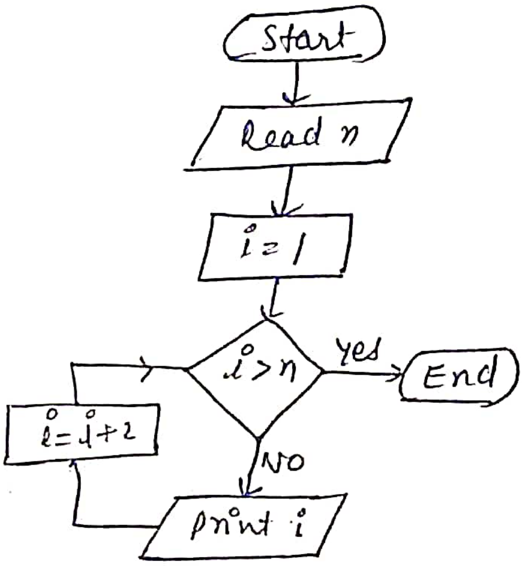
n=~~4~~3, i=3 3>3 → NO → print Deepti, i = 3+1 = 4

n=3, i=4 4>3 → yes
↳ exit.

⊙ Printing 1 to N odd numbers.

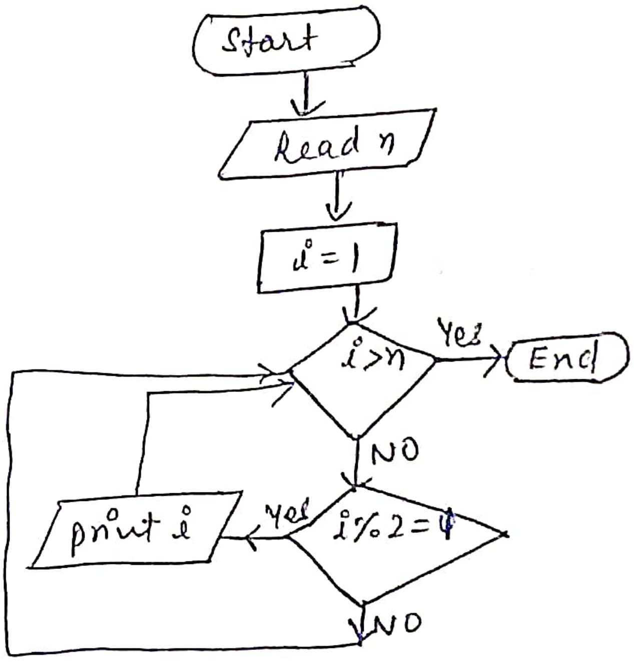
pseudo code-

```
① → read n
② → i=1
③ → if i>n
        exit
④ → else
        print i
        i = i+2
        go to step 3
```



(i = i+2 because if we add 2 to an odd number, we get an odd number (next odd number)).

Another way:→



pseudo code -

```
① → Read n
② → i = 1
③ → if i>n
        Exit
④ → else
        if i%2 = 1
            print i
            go to step③
        else
            go to step③
```