******
Day 2/180
*****

Lec-2

Flowcharts & Pseudocode

Classwork

Simple flowchart questions

i/P : 2 3 , -7 9 , -25 -10
O/P : 5 , 2 , -35

I> Sum of two numbers :-

(Diagram. repre.) Flowchart | Pseudocode (Textual Representation of algo/code)

Start

Read a, b

Sum = a + b

Print Sum

End
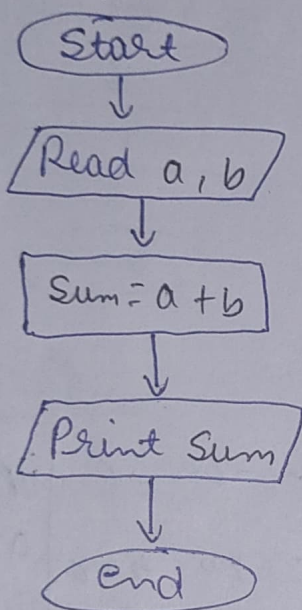
Step 1 :- Start (optional step)

Step 2 :- Read a and b

Step 3 :- Sum = a + b

Step 4 :- Print sum

Step 5 :- Exit

*> Flowcharts and Pseudocode are language independent.

NOTE :- Debug / Dryrun / verify your Flowcharts & pseudocode with bunch of examples / input values

i/P → 2   3

a = 2, b = 3
Sum = $\overset{a}{2} + \overset{b}{3}$
Sum = 5
print 5
⤷ O/P

i/P → -7, 9.

a = -7, b = 9
Sum = $\overset{a}{-7} + \overset{b}{9}$
Sum = 2
print 2
⤷ O/P

i/P -25, -10

a = -25, b = -10
Sum = $\overset{a}{-25} + \overset{b}{(-10)}$
Sum = -25 - 10
Sum = - 35
print -35
⤷ O/P

2> Sub two no. :-

i/p → 2 3 , 3 2 , −20 6 , 15 −3 , −20 −6
o/p → −1   1   −26   18   −14

Flowcharts

Start
↓
Read a, b
↓
Sub = a − b
↓
Print Sub
↓
End
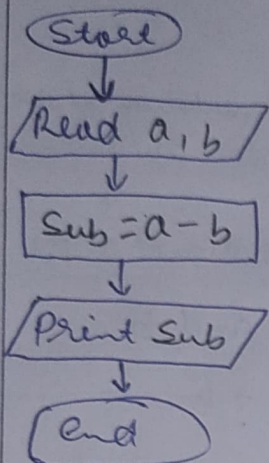
Pseudocode

Step 1: Start

Step 2: Read a, b

Step 3: Sub = a − b

Step 4: Print Sub

Step 5: Exit

Dry Run

i/p → 2 3	i/p → 3 2	i/p → −20 , 6	i/p → −20 −6
a = 2 b = 3	a = 3 b = 2	a = −20 b = 6	a = −20 b = −6
Sub = a − b	Sub = 3 − 2	Sub = −20 − 6	Sub = −20 −(−6)
= 2 − 3	Sub = 1	Sub = −26	= −20 + 6
Sub = −1	print 1	print −26	Sub = −14
print −1 o/p			print −14

Homework (write Flowchart & Pseudocode)

i) Product of 2 no.

ii) Divide 2 no.

Homework Solⁿ

i) Product of two no:- $\begin{cases} \text{i/P} \rightarrow: 2 & 3, & 5 & -2, & -8 & -2 \text{)} \\ \text{O/P}: 6 & & -10 & & 16 \end{cases}$

Flowchart

Start
↓
Read a, b
↓
Mul = a * b
↓
print Mul
↓
end
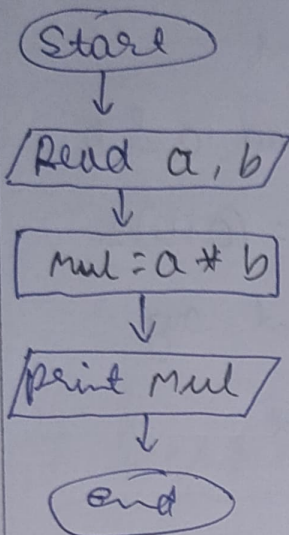
Pseudocode

Step 1: start

Step 2: Read a and b

Step 3: Mul = a * b

Step 4: Print Mul

Step 5: End

Dry Run

i/P → a = 2 b = 3

Mul = 2 * 3

Mul = 6

print 6

---

i/P → a = 5 b = -2

Mul = 5 * (-2)

Mul = -10

print -10

---

i/P → a = -8, b = -2

Mul = -8 * (-2)

Mul = -16

Print -16

---

ii) Divide of two no:-

i/P → 4 2, -20 5, -8 -4   6   0

O/P → 2   -4   2   can't divide by zero
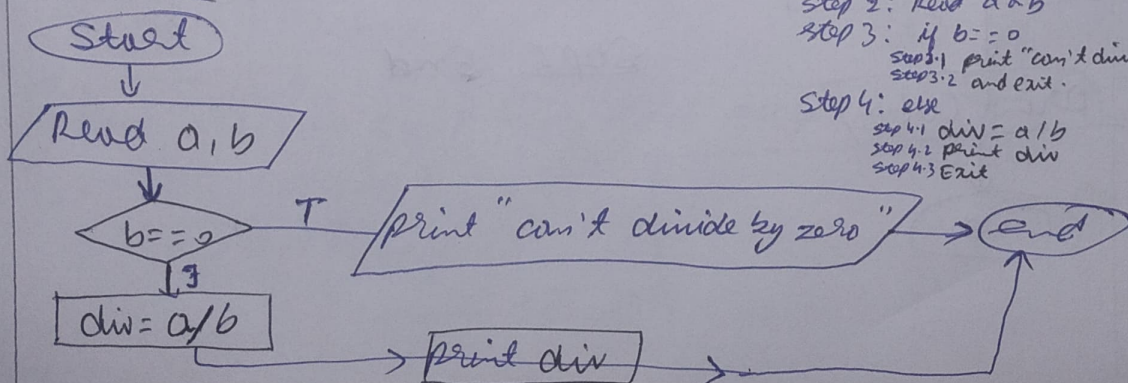
Pseudocode

Step 1: Start
Step 2: Read a & b
Step 3: if b = = 0
   Step 3.1 print "can't divide by zero"
   Step 3.2 and exit.
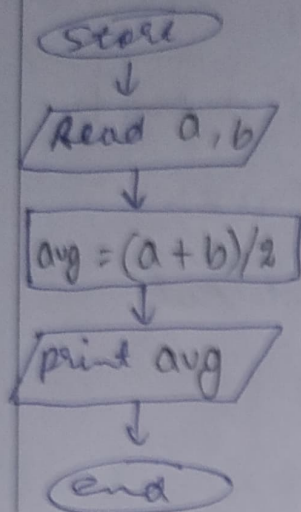Step 4: else
   Step 4.1 div = a/b
   Step 4.2 print div
   Step 4.3 Exit

Flowchart

Start
↓
Read a, b
↓
b = = 0 —T→ print "can't divide by zero" → end
↓ F
div = a/b
→ print div →

3) _Avg of 2 no._

$\left\{ avg = \dfrac{\text{Sum of observation}}{\text{Total no. of observation}} \right\}$

Flowchart   _Pseudocode_

Start

↓

Read a, b

↓

$avg = (a+b)/2$

↓

print avg

↓

End

Step1: Start

Step2: Read a & b

Step3: $avg = (a+b)/2$

Step4: print avg

Step5: Exit

---

4) _Find cube of a number_

i/p → $n = 3$ $n = 5$ $n = 7$

o/p → • 27 $(3 \times 3 \times 3)$ 125 $(5 \times 5 \times 5)$ 243 $(7 \times 7 \times 7)$

↳ cube= $n \times n \times n$ ↳ cube= $n \times n \times n$ ↳ cube= $n \times n \times n$

Flowchart   _Pseudocode_

Start

↓

Read n

↓

Cube= $n \times n \times n$

↓

Print cube

↓

End
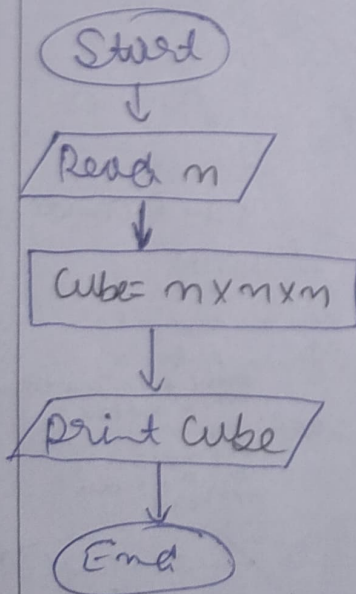
Step1: Start

Step2: Read n

Step3: cube= $n \times n \times n$
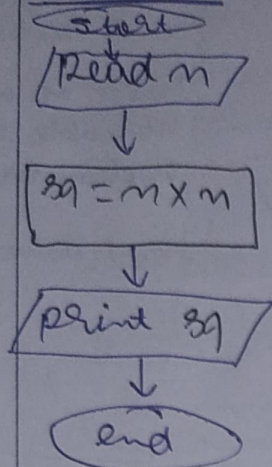
Step4: Print cube

Step5: End

5) Find square of a number:-

$IP \rightarrow m = 5$ $m = 7$ $m = 12$

$OIP \rightarrow$ $25 (5 \times 5)$ $49 (7 \times 7)$ $144 (12 \times 12)$

$Gsq = m \times m$ $Gsq = m \times m$ $Gsq = m \times m$

Flowchart

Start

Read m

↓

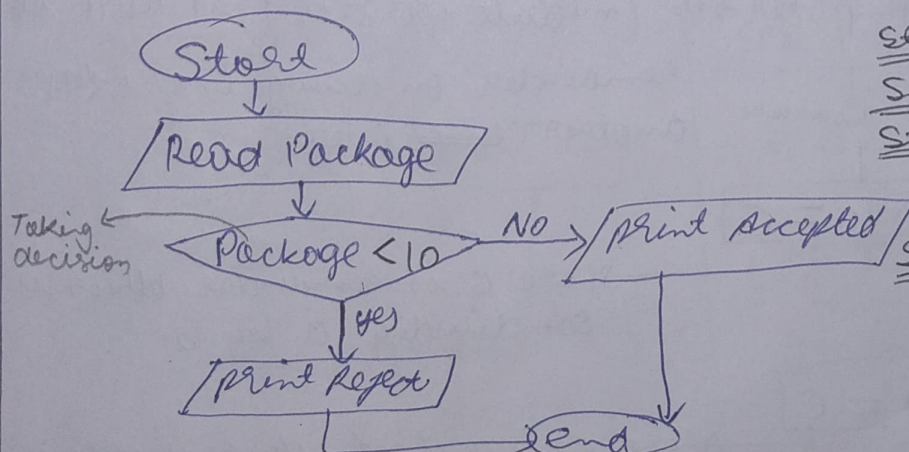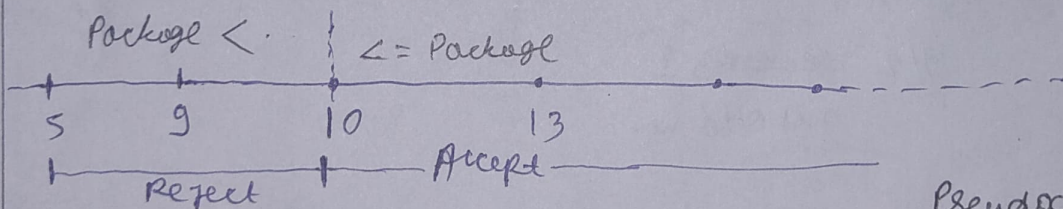$sq = m \times m$

↓

print sq

↓

end

Pseudocode

Step 1 : Start

Step 2   Read m

Step 3 $sq = m \times m$

Step 4   print sq

Step 5   End

---

* # Decision making in Flowcharts

Realife Example - 1

MNC — Interview crack

$i/P \rightarrow$ Package = 13   Package = 10   Package 9   Package 5

$O/P \rightarrow$ Accept   Accept   Reject   Reject

Package < · $\leq$ = Package

```
+----+----+----+----+----·----·----·--- ~ ~
5    9    10   13
```

Reject   Accept

Start

↓

Read Package

↓

Taking decision → Package < 10 → No → Print Accepted
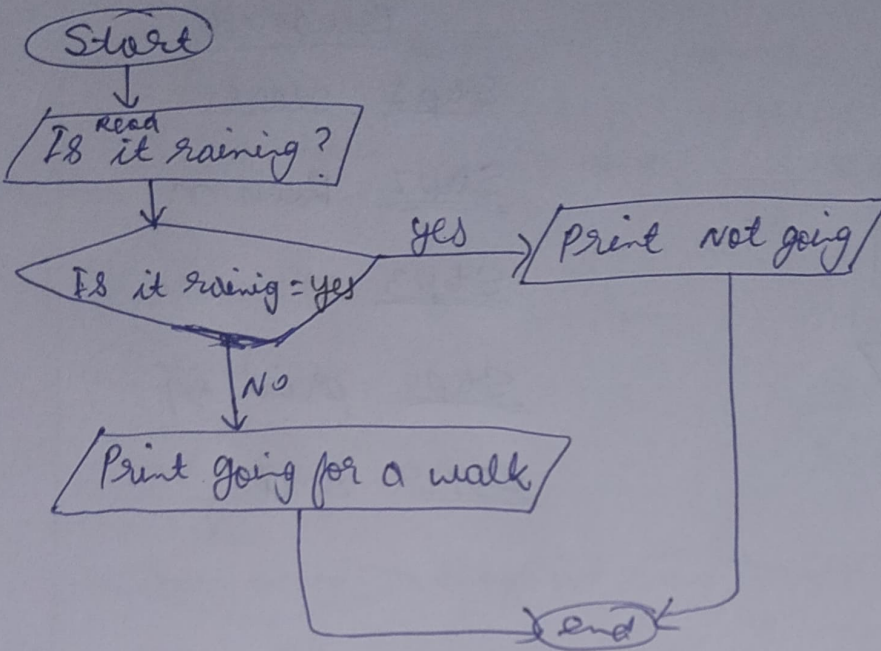
↓ yes

Print Reject

→ end

Pseudocode

St 1 : Start

St 2 : Read Package

St 3 : if Package < 10

   St3.1 Print Reject & exit

St 4 else

   St4.1 Print Accepted & exit

Realbife Example - 2

Start

Read
Is it raining?

Is it raining = yes → yes → Print Not going

No

Print going for a walk

end

* value → Yes or No
is a Boolean value

In CPP we use boolean
variable to read
boolean value

bool value = [→ true
                → false

6) Check a given is even or odd

i|P → m = 4   m = 7   m = -36   m = 27
o|P → even   odd   even   odd

m → 4
4/2 remainder 0
4 is an even no.

m → 7
7/2 remainder 1
7 is odd no.

So in mathematics
given: m → remainder 1 means odd no.
m/2 →
→ 0 means even no.

* In computer → % (modulo operator) is used to find
remainder on diving one integer with
another integer

Integer   Integer value
value   ↓
  ↓
a % b = C

→ here C is remainder which is obtained
on dividing a by b

a / b = C

→ here c is quotient which is obtained on
dividing a by b

% ⟶ modulo operator. It gives remainder on division

/ ⟶ division operator- It gives quotient on division

$n \to 4$ $n \to 5$ $n \to 36$

$4 \% 2 = 0$ $5 \% 2 = 1$ $-36 \% 2 = 0$

∴ 4 is even no.   ∴ 5 is odd no.   ∴ -36 is even no.

So in computer

given : ⟶ n

$n \% 2 \to$ 0 (even no.)
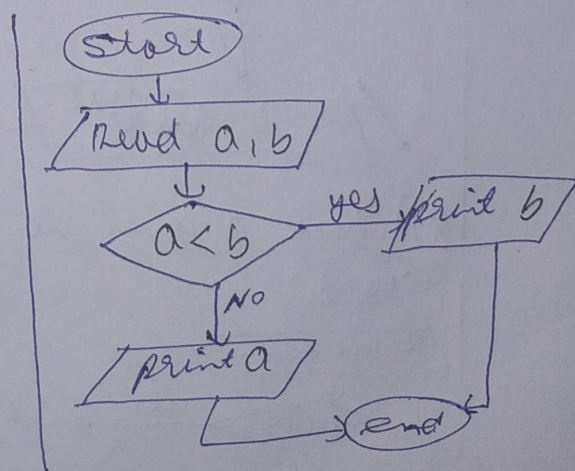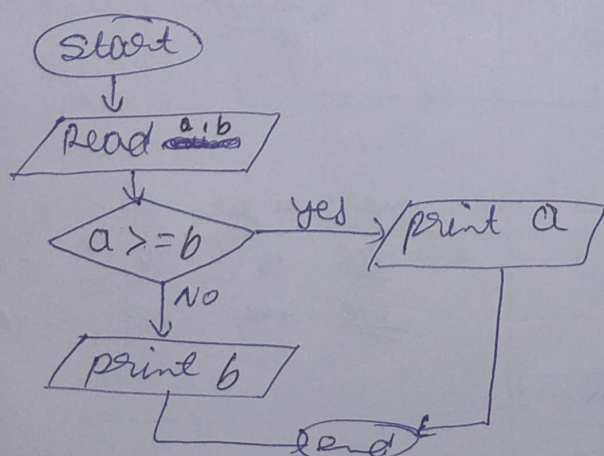   ⟶ 1 (odd no.)

Flowchart
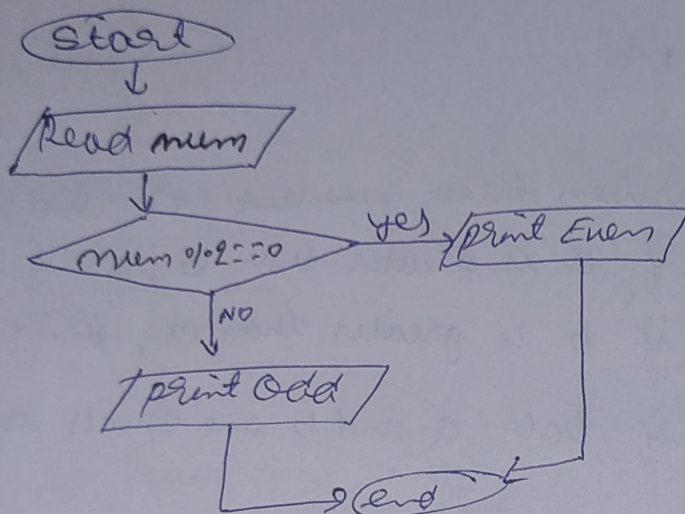
Pseudocode

St1 Read num

St2 if num % 2 == 0

St2.1 Print Even & exit

step3 else

St3.1 print Odd & exit

NOTE:- == is called "equal to operator"

= is called "assignment operator"

7) Check greater of two no. (a and b)

i/p → 2   3 ,   8   -7   /   -9   -25 ,   3   3 , -6 -6

o/p → 3   8   -9   3   -6

Pseudocode

St 1:- Start
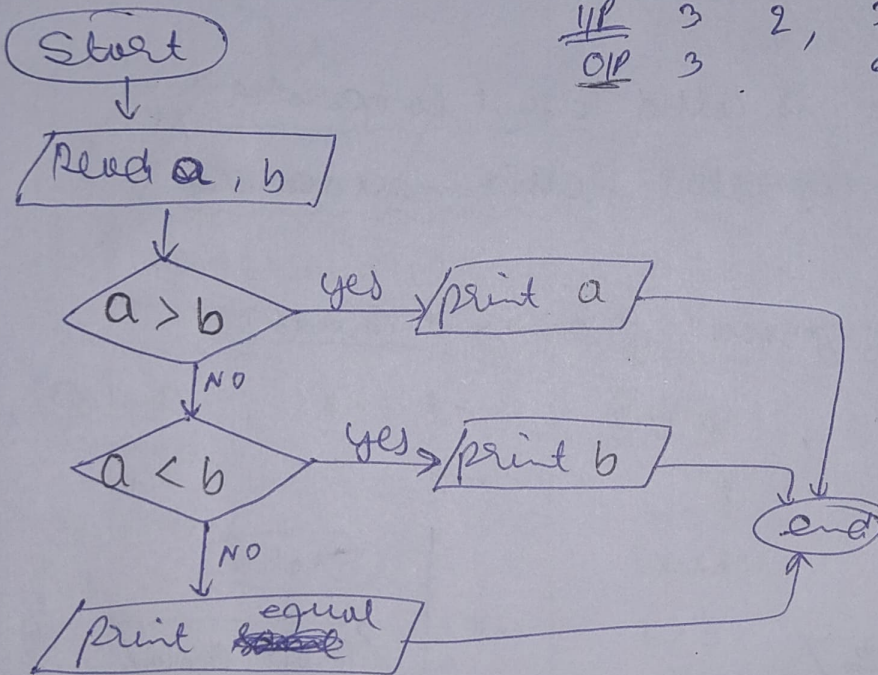St 2:- Read a & b
St 3:- if a < b
   St 3.1:- print b & exit
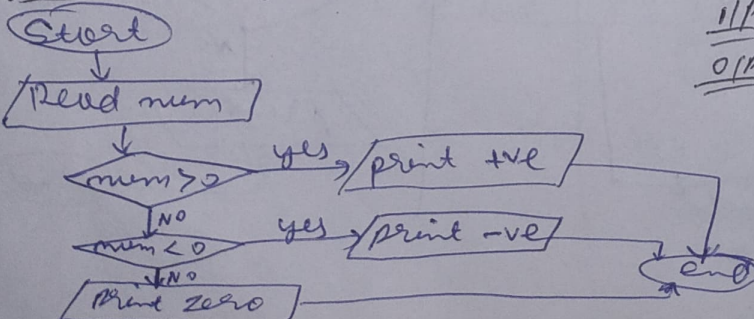St 4:- else
   St 3.2:- print a & exit

Homework

Problem statement:- Given two number a and b
   if a is greater than b, print a,
   if b is greater than a, print b,
   if both a and b are equals print equal

i/p	3	2,	3	3,	3	4
O/P	3		equal		4	

Start

↓

Read a, b

↓

a > b — yes → print a

↓ NO

a < b — yes → print b

↓ NO

Print ~~equal~~ equal

end

8)) Check a given no. is +ve, -ve, or zero

Start

i/p	3	4	-6	0
O/P	+ve	+ve	-ve	zero

↓

Read num

↓

num > 0 — yes → print +ve

↓ NO

num < 0 — yes → print -ve

↓ NO

Print zero

end

* Loops in Flowcharts

9)) Print a given no. 'n' times

i/P→   mem = 3   n = 2   mem = 3   n = 5
o/P→   3   3   3 3 3 3 3
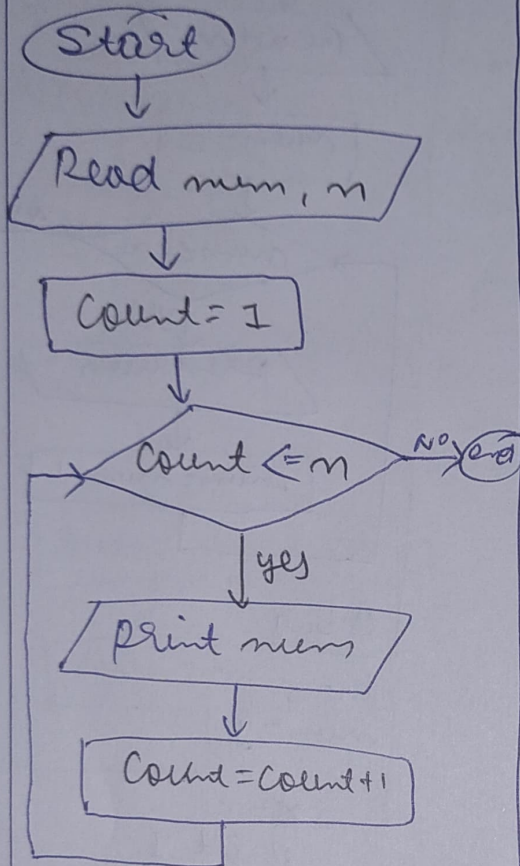
Left flowchart:

Start
↓
Read mem, n
↓
Count = 1
↓
Count <= n —— NO → end
↓ yes
Print mem
↓
Count = Count +1

Right flowchart:

Start
↓
Read mem, n
↓
Count = 1
↓
Count > n —— yes → end
↓ NO
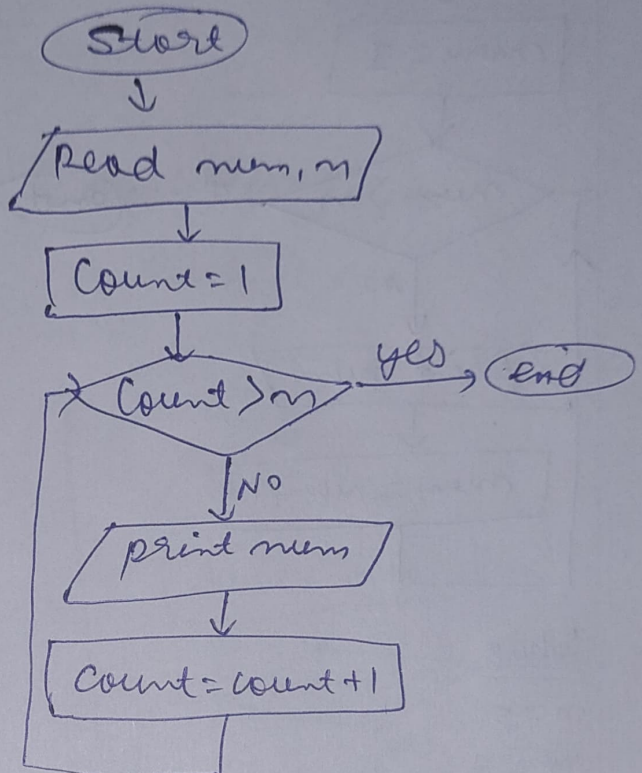print mem
↓
Count = Count +1

Debug (left)

mem = 3
n = 5
Count = 1

1 <= 5
  print 3
  Count = 2

2 <= 5
  print 3
  count = 3

3 <= 5
  print 3
  Count = 4

4 <= 5
  print 3
  count 5

5 <= 5
  print 3
  Count = 6

6 <= 5
  (No) end

Debug (right)

mem = 3
n = 5
count = 1

1 > 5
  print 3
  Count = 2

2 > 5
  print 3
  count = 3

3 > 5
  print 3
  count = 4

4 > 5
  print 3
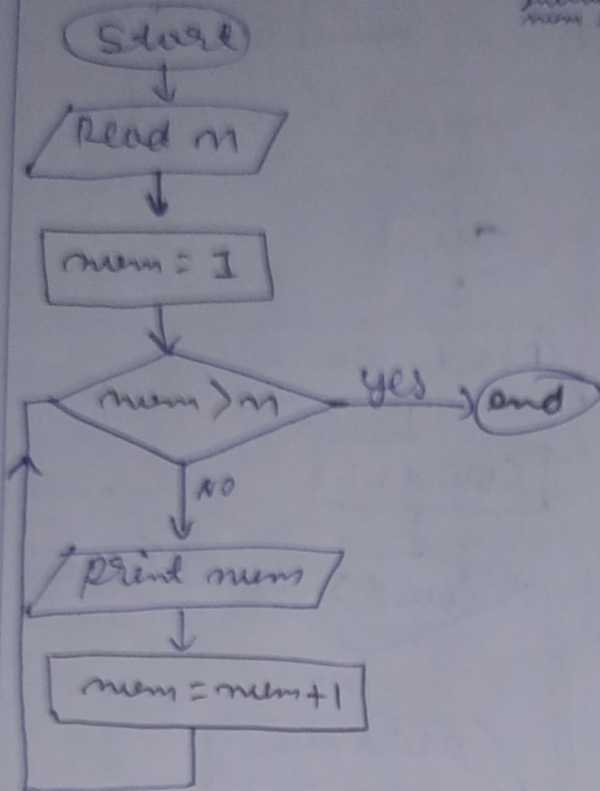 count = 5

5 > 5
  print 5
  count = 6

6 > 5
  yes → end

10) Print n natural no:-

i/p → m = 5   m = 10
o/p → 1 2 3 4 5   1 2 3 4 5 6 7 8 9 10 11 bstep
   initial   initial
   mem is none   mem is none
                                    mem = 10
                                    mem+1

Left flowchart:

(Start)
↓
/ Read m /
↓
[mem = 1]
↓
< mem > m > —yes→ (end)
↓ NO
/ Print mem /
↓
[mem = mem + 1]
(loop back)

Right flowchart:

(Start)
↓
/ Read m /
↓
[mem = 1]
↓
< mem ≤ m > —No→ (end)
↓ yes
/ print mem /
↓
[mem = mem + 1]
(loop back)

Debug (left)

m = 5
mem = 1

1 > 5
   print 1
   mem = 2

2 > 5
   print 2
   mem = 3

3 > 5
   print 3
   mem = 4

4 > 5
   print 4
   mem = 5

5 > 5
   print 5
   mem = 6

6 > 5
   yes end

Debug (right)

m = 5
mem = 1

1 ≤ 5
   print 1
   mem = 2

2 ≤ 5
   print 2
   mem = 3

3 ≤ 5
   print 3
   mem = 4

4 ≤ 5
   print 4
   mem = 5
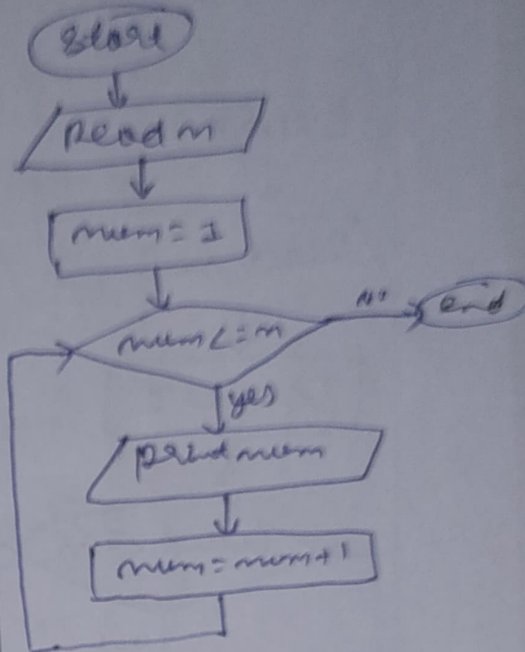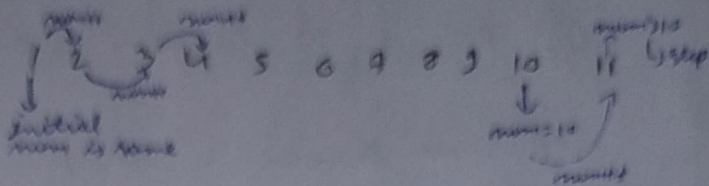
5 ≤ 5
   print 5
   mem = 6

6 ≤ 5
   No → end

11) Sum of n natural no.

i/p → m = 5
o/p → 15   (1 + 2 + 3 + 4 + 5)

i/p → m = 10
o/p → m = 55   (1 + 2 + 3 + ... + 10)

↓
initially num
~~n~~ start
from 1

↓
initially num
~~n~~ start
from 1

initially sum = 0

sum+num
num = 1 → sum = 0 + 1 = 1
↓++
num = 2 → sum = $\overline{1 + 2}$ = 3   num+num
↓++
num = 3 → sum = $\overline{3 + 3}$ = 6
↓++
num = 4 → sum = $\overline{6 + 4}$ = 10
↓++
num = 5 → sum = $\overline{10 + 5}$ = 15
↓++
num = 6, > 5 __stop__ (end)

num > n ✗

(Start)
↓
/Read m /
↓
[Sum = 0, num = 1]
↓
◇ num > n ◇ —yes→ (end)
↓ NO
[Sum = Sum + num]
↓
[num = num + 1]

(Start)
↓
/ Read n /
↓
[Sum = 0, num = 1]
↓
◇ num ≤ n ◇ —No→ (end)
↓ yes
[Sum = Sum + num]
↓
[num = num + 1]
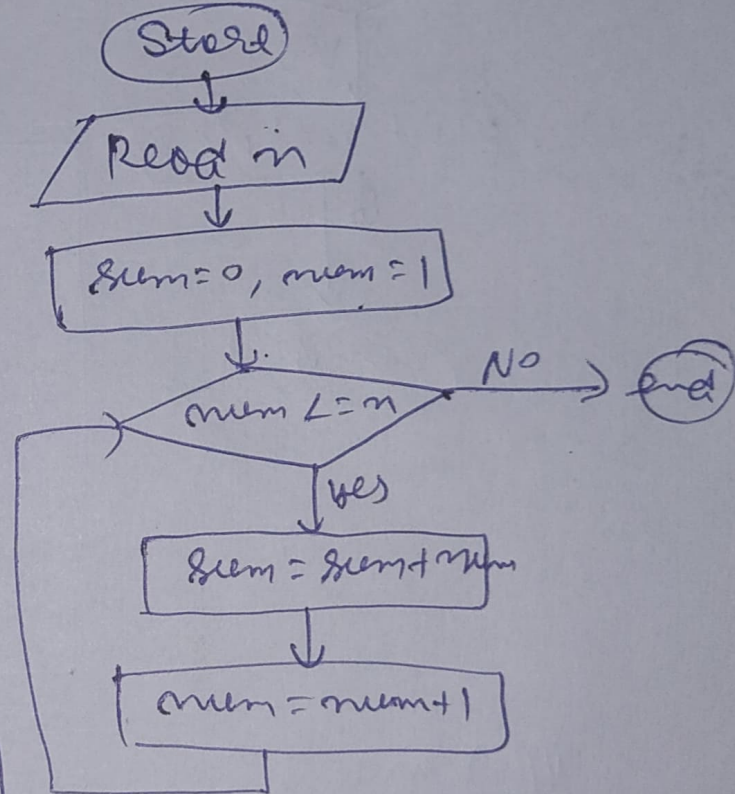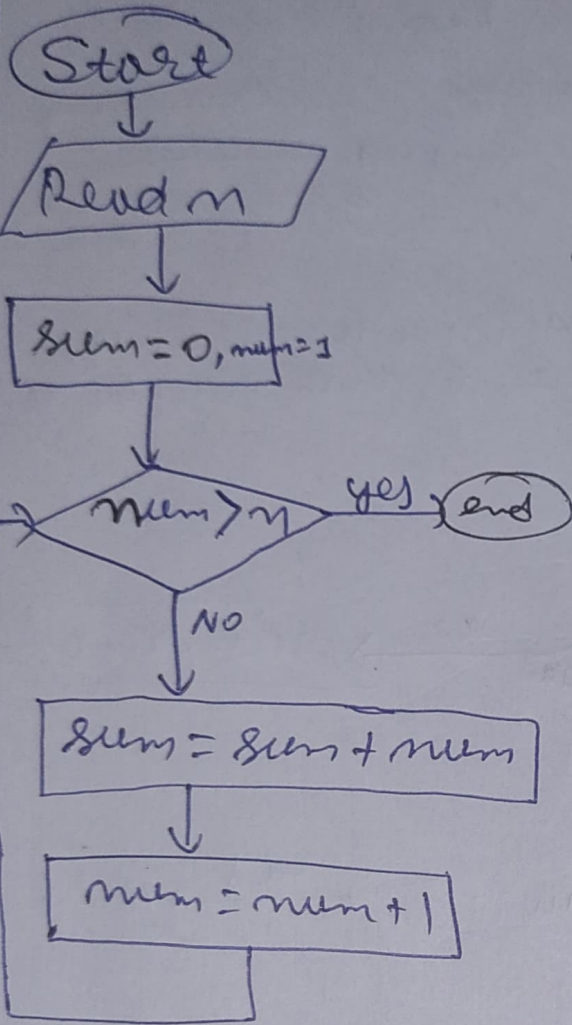
12) Check a given no. is prime or not.

i/P → n = 2   n = 4   n = 7   n = -3

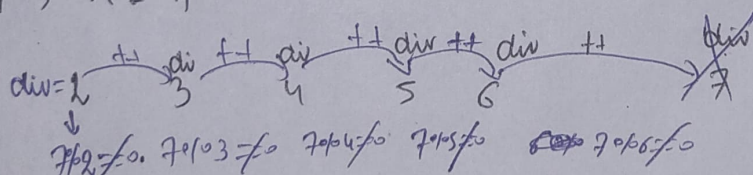o/P → prime no.   not prime   prime   not prime

       ↳ The no. which has only two factors 1 and the no. itself

Rage of prime no.

never a prime no. | prime no.

—1   0   1   2   3   4   5
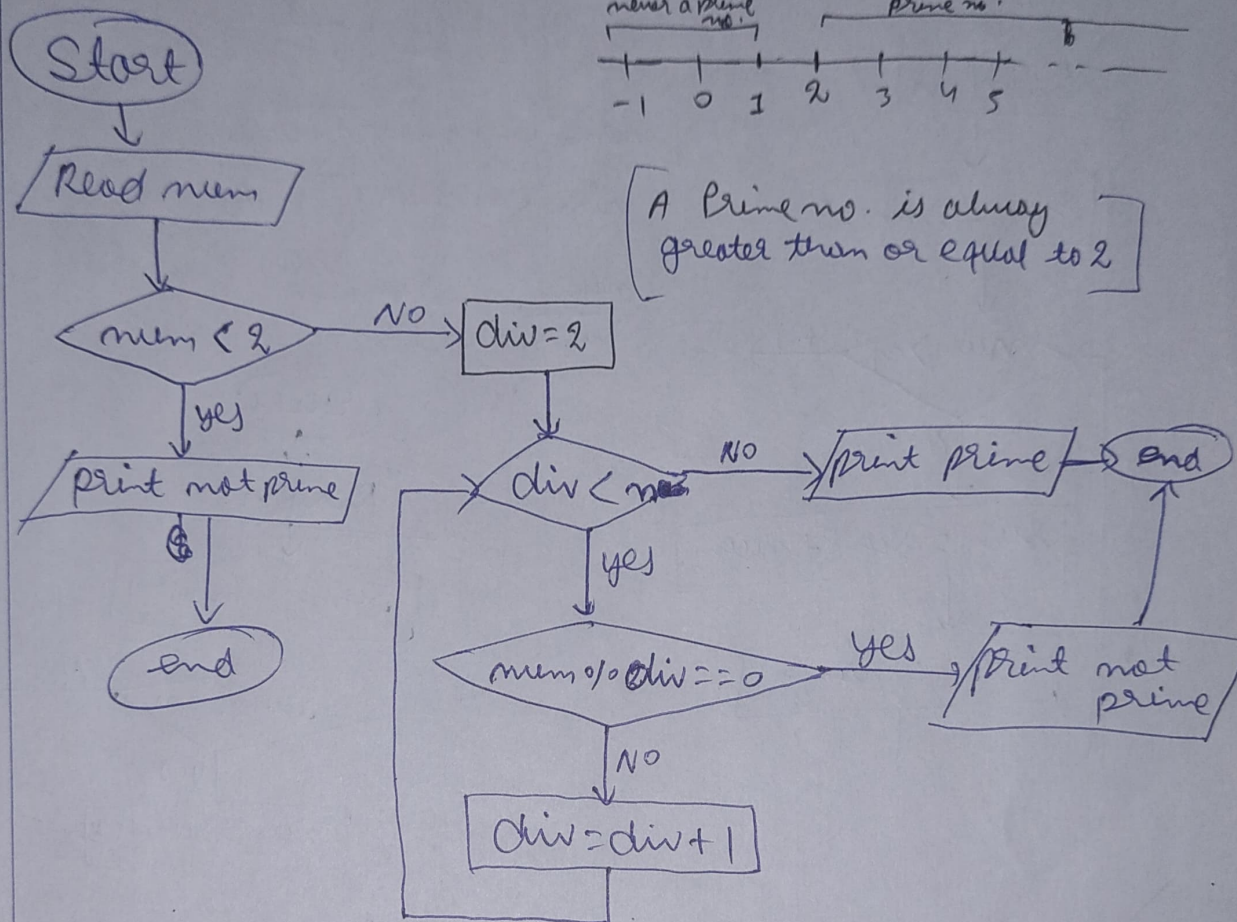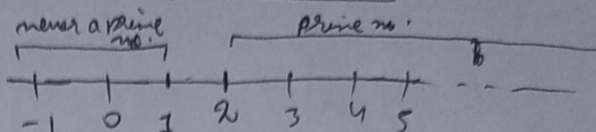
[A Prime no. is always greater than or equal to 2]

Start

Read num

mum < 2 — NO → div = 2

yes

Print not prime

End

div < num — NO → Print prime → End

yes

num % div == 0 — yes → Print not prime

NO

div = div + 1

i/P → n = 7

div = 2 → div ++ → 3 → div ++ → 4 → div ++ → 5 → div ++ → 6 → div ++ → div 7

7/2 ≠ 0. 7/3 ≠ 0 7/4 ≠ 0 7/5 ≠ 0 7/6 ≠ 0

divide the num from [2 to num—1] & if the num is not divisible in this range then we can say that the given num is prime

for 7 → check 2 to 6
10 → check from 2 to 9
15 → check 11   2 to 14