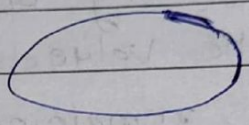
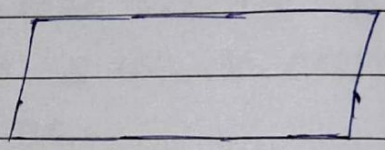
#3 Flow of Program:

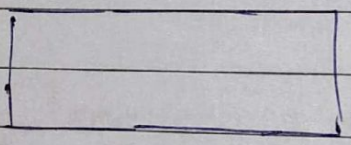
8 Aug 2024

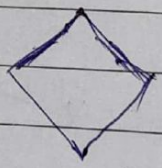
* Flowcharts:-

Flowchart is a basically a tree chart diagram which represent about How program is working, which steps are taken while program is running.

start / stop ⟶ 

Input / output ⟶ 

Processing ⟶ 

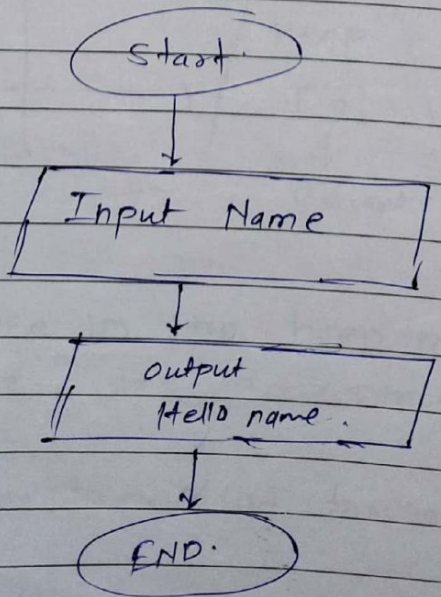
Condition ⟶ 

* Flow direction of Program:- ⟶

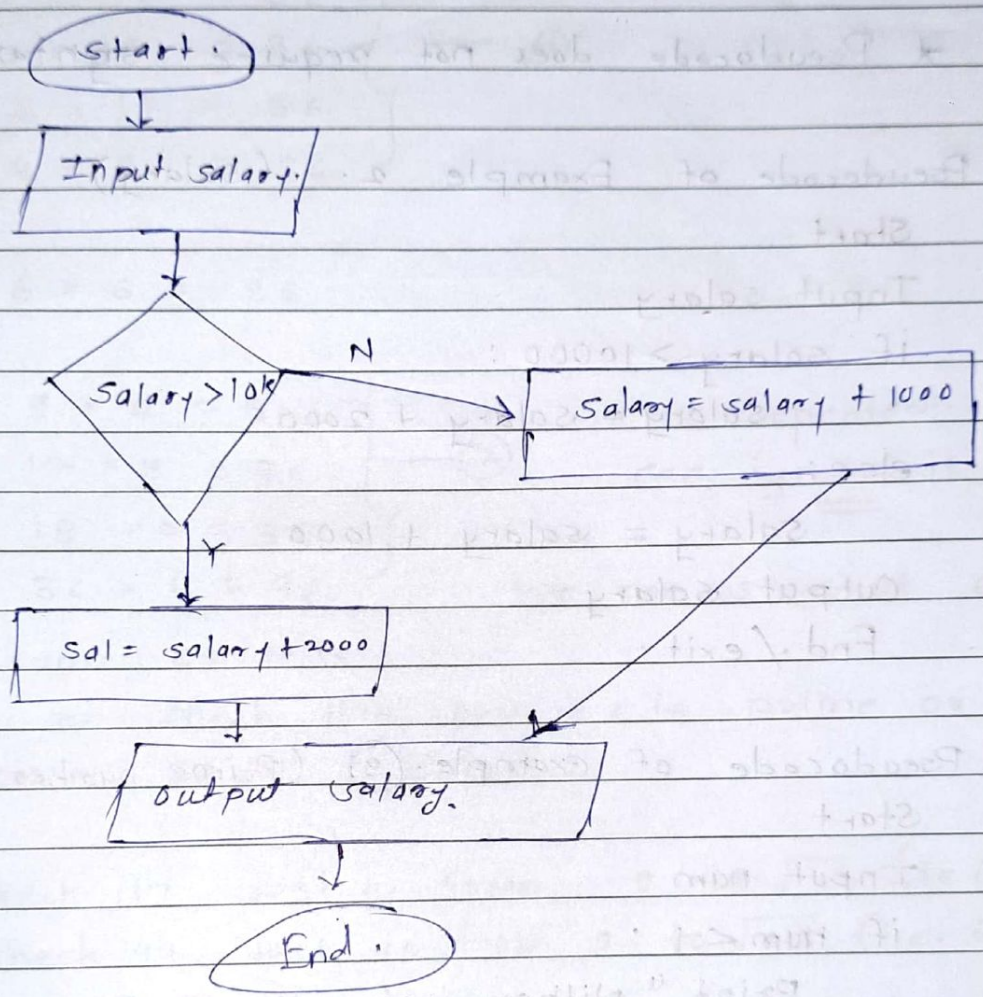
* Example:-

① Take a name & output Hello name

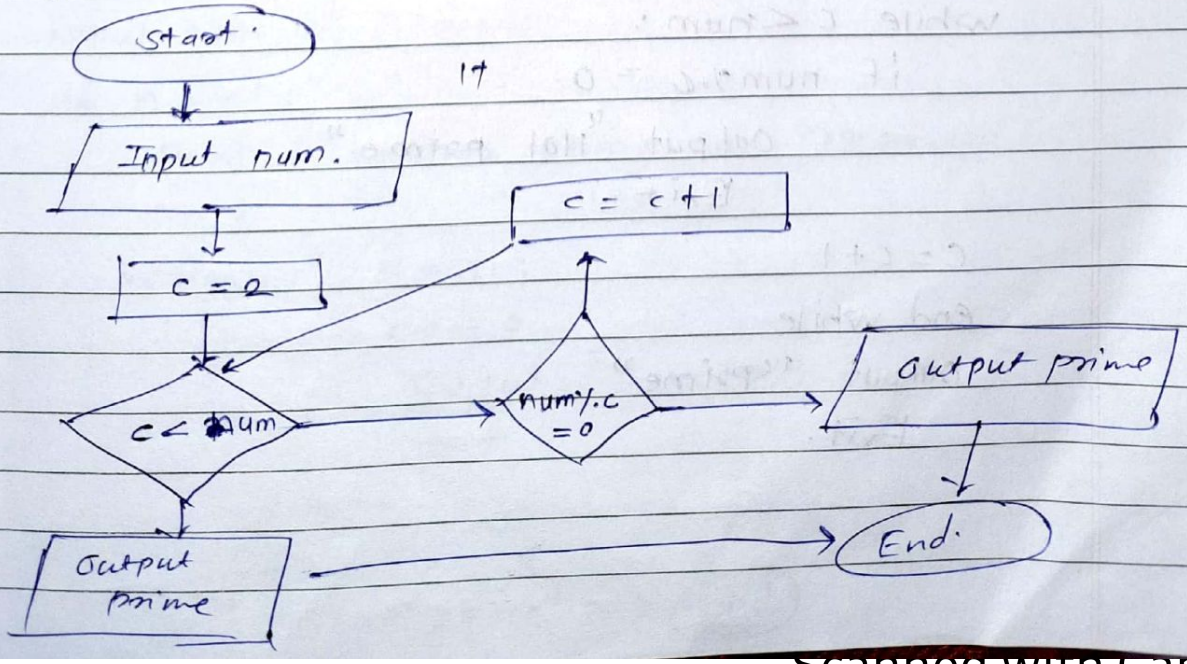
⟶



*. Example. 2. - Take input of salary. If the salary is greater than 10,000 add bonus of 2000, otherwise add bonus as 1000.



Ex (B) Input a number & print whether it is prime or not.



* Pseudocode :-

It is like rough code which represents how the algorithm of program works.

* Pseudocode does not require syntax.

* Pseudocode of Example 2. (salary)

```
Start
Input salary
if salary->10000 :
        salary = salary + 2000
else:
        salary = salary + 1000
Output salary
End./exit.
```

* Pseudocode of example. (3) (Prime numbers example).

```
Start
Input num
if num <= 1 :
    Print " Nither prime nor composite".
c = 2
while c <num :
    if num%c = 0
        Output "Not prime"
        Exit..
    c = c+1
end while
Output "prime"
    Exit.
```

* square root Method for finding num. is prime or not

→ Lets take a number 36.

$1 \times 36 = 36$.
$2 \times 18 = 36$
$3 \times 12 = 36$.
$4 \times 9 = 36$.

— ①.

$6 \times 6 = 36$

$9 \times 4 = 36$.
$12 \times 3 = 36$.
$18 \times 2 = 36$
$36 \times 1 = 36$.

— ②. it is repeated. so we can ignore it ② eq^n.

We can stop at 6×6.

As same as this

* We can check the number is prime or not. from ~~2~~ 2 to $\sqrt{num}$.

① To check 17, just go from 2 to $\sqrt{17}$. (ie (.4))
② To check 40, just go from 2 to $\sqrt{40}$ (i.e. 6).

* Pseudocode for this is :-

```
start
input num.
if n <= 1 :
    Print ("neither prime nor composite")
c = 2.
while c*c <= n :
    if n %. c == 0
        output "not Prime"
        exit.
    c += 1
end while.
Output "prime" ⟹ (end).
```