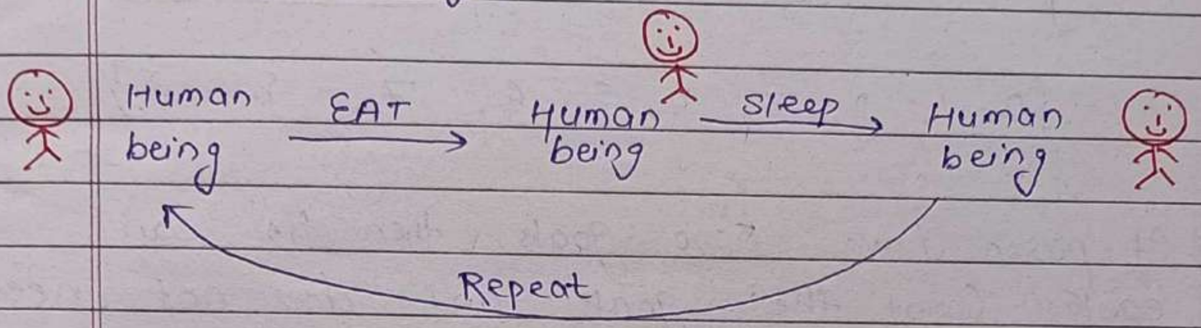
Lecture - 1

Introduction to programming

10,000 years ago,



→ These people does not know Counting, because at that time Couting is not discoverable.

→ But they also works for survival.
A man have 5 goats.
He Count the number of goats by picking a stone for one goat.

→ We he go to field With goats he filled his pocket With stones for one goat one stone.

When he Comes home he throw stone from pocket We one goat enter one stone throw.

At last if not any stone ~~is~~ in packet then all goats are safe.

If stone are kept in packet means Wild amimal can eat the goat.

After many years, Counting System Comes.

The first Counting system is decimal number.

{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

If person have 5000 goats, then he can easily Count the goats and does not need to carry any Stones.

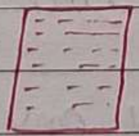
0 is discovered by Aryabhata.

At 1800, Some problem arises.
people started to doing bussiness and maintain a log book of sales or import - Export.

Let,
A bussinessman maintain 4 log book of his import - Export details.

Now, he want to calculate total profit / loss.

Bussiness man

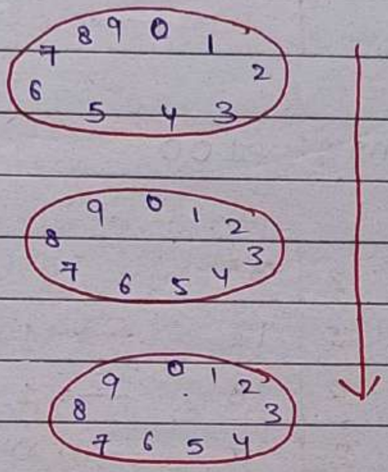
 × 4 = 4000 pages

log Book
1000 pages

Now he started Calculating,
There is a high probability to make mistakes while calculating.

Then, Charles Babbage Makes a Computer called as Mechanical Computer which only calculate.

Looks like



Size is very big
It is adjust in
1 room.

→ After some year, we found transistor.

→ At present computer can only understand binary language → 0 or 1.

0 → 0
1 → 1
2 → 10
3 → 11
4 → 100
5 → 101
6 → 110
7 → 111

→ [10110] → Convert into Decimal

$1\times2^4 + 0\times2^3 + 1\times2^2 + 1\times2^1 + 0\times2^0$

$16 + 0 + 4 + 2 + 0$

22 (Decimal)

→ Decimal to binary

4 →

		rem
2	4	
2	2	→ 0 ↑
2	1	→ 0
	0	→ 1

Ans : 100

35 →

2	35	
2	17	→ 1 ↑
2	8	→ 1
2	4	→ 0
2	2	→ 0
2	1	→ 0
	0	→ 1

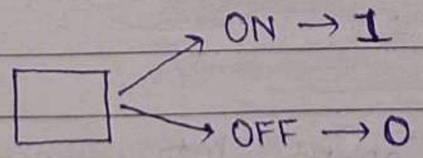
When 0 comes ← We stop.

Ans : 100011

* <u>Transistor</u> :

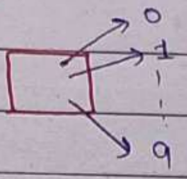
Before Transistor → Computer use decimal Number system

After Transistor → Computer use binary Number system.

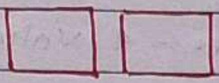


Transistor

Decimal :

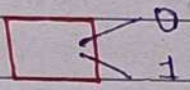


We can fill 10 Number in one place.

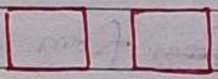


→ In this We can fill 10 x 10 = 100 Number.

Binary :

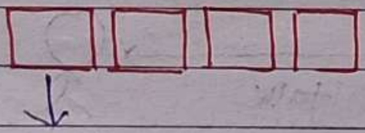


→ We can fill two types in one place



→ In this We can fill 2x2 = 4 types.

0 0
0 1
1 0
1 1

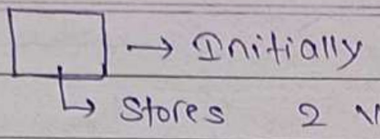


→ In this We can fill 16 types.

This one box is one tranisistor.

Moore tells that in every 2 year the capacity of transistor get doubled.

He predict this in 1956-57, and till now it is true.

→ 

After 2 year,

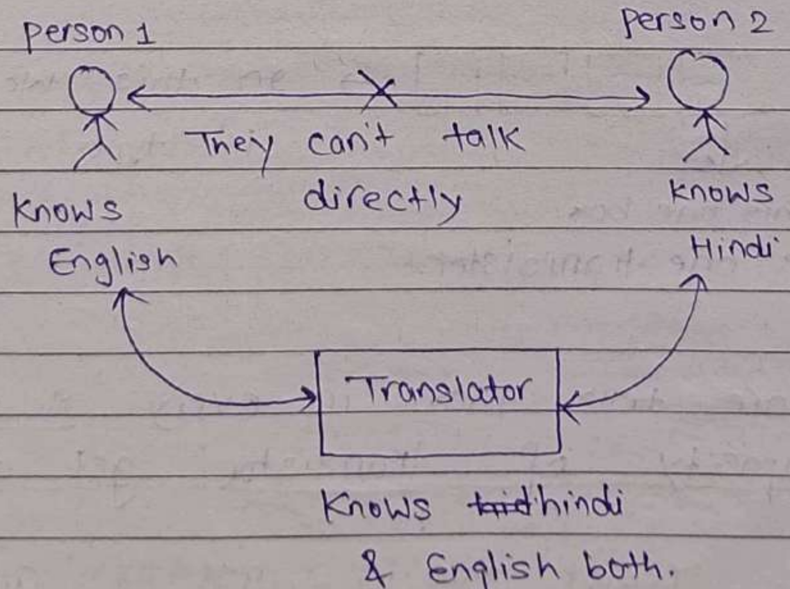
Now, It stores 4 values.

capacity of transistor is increased very much.

Computer can only understand Binary language.
At 1960-70 programmer can write Code only in Binary form.

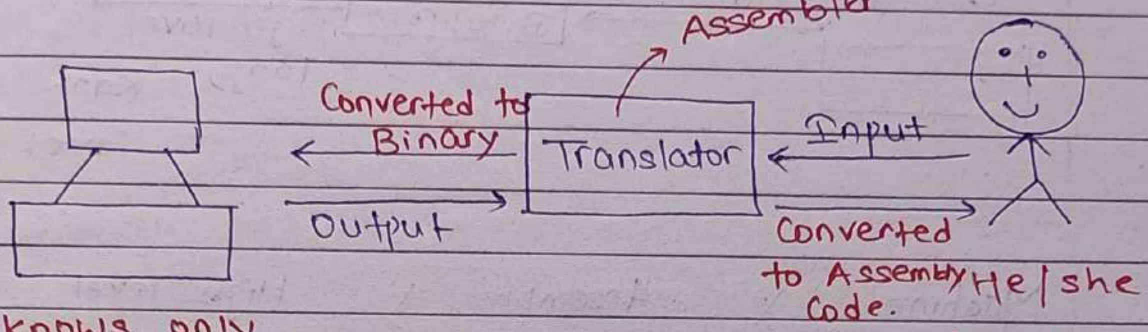
Writing code in Binary form is very complex.
While Writing Code in Binary ~~from~~ form there is a high probability of mistake.

So, it does not give Correct solution.



Translator can translate the language into known language of Both person.

Same in Computer World



Knows only Binary language.

He/she Knows Assembly language

Assembly lang: ADD 2 3

Computer does not understand Assembly language. so, a translator translate the Assembly Code into Binary Code.

Assembler can ~~t~~convert Assembly Code to Binary Code and vice versa.

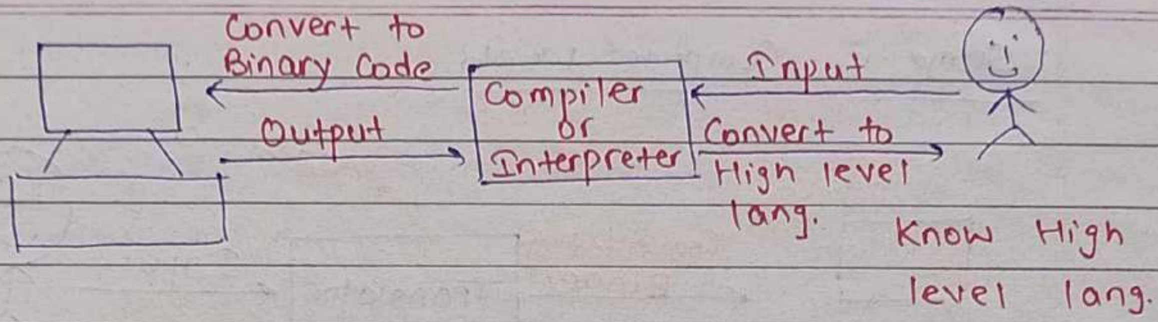
In Assembly Language the length of Code is very high.

To reduce the length of Code We interoduce high level language.

↳ Java, C++.

↳ similar to english.

High level language is programmer friendly.



Machine language > Assembly language > High level language

Execution time

* find sum of

$1 + 2 + 3 + 4 + \cdots + 100$

1st Approach:

$\underbrace{\underbrace{\underbrace{1 + 2}_{3} + 3}_{6} + 4}_{10} + \cdots + 100$

Ans = 5050

It takes more time to solve and take more space.

2nd Approach:

formula for first n natural number:

$$\frac{n(n+1)}{2} = \frac{\overset{50}{\cancel{100}} \times 101}{\cancel{2}}$$

$$= 5050$$

It take very less time and very less pace.

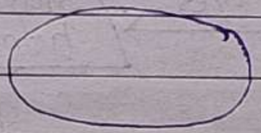
Time Complexity :

Total time taken by the algorithm with respect to input size.

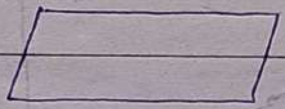
* **Flowchart :-**

A diagram that shows the connection b/w different stage of a process or part of system.

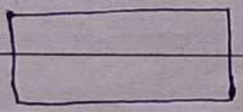
Components of flowchart



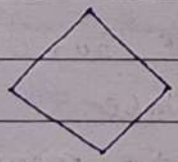
Start / End



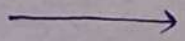
Input /output



process



Descision Making

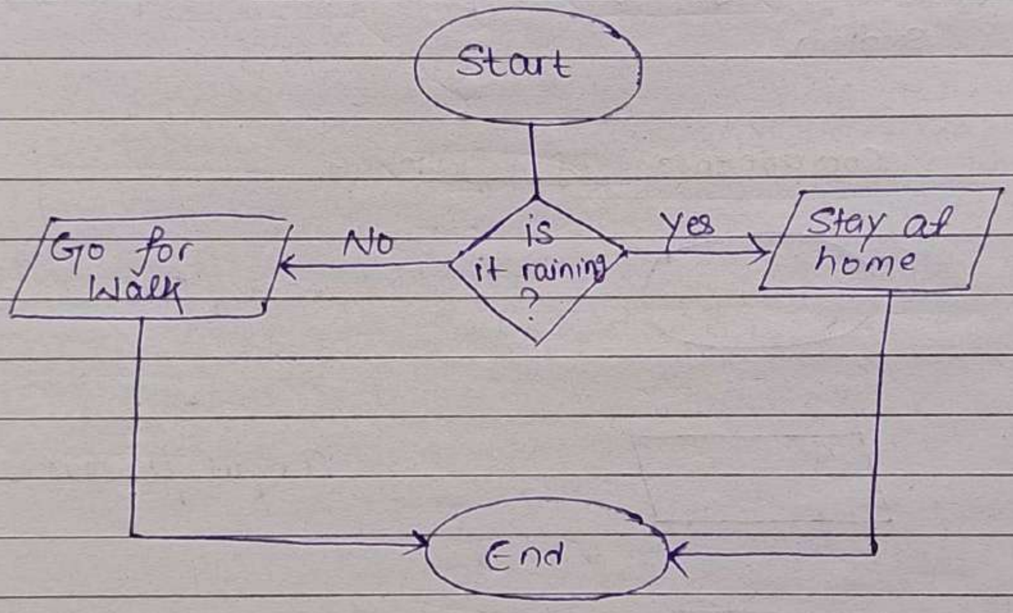


flow / Direction

Example - 1

* Decide you have to go out or not , if it is a Rainy Season.

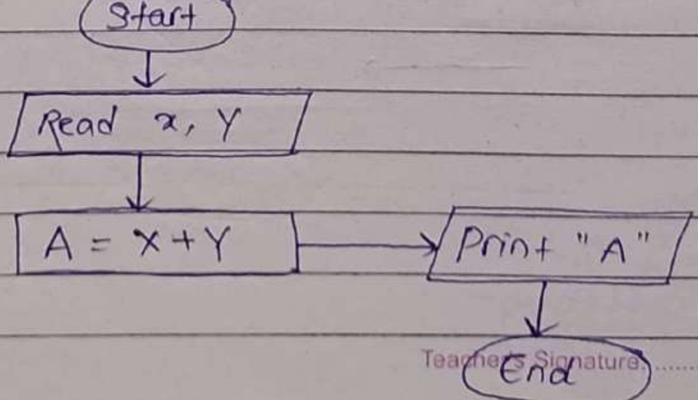
→ If raining, then stay at home
→ If not raining, then go for Walk



Example - 2

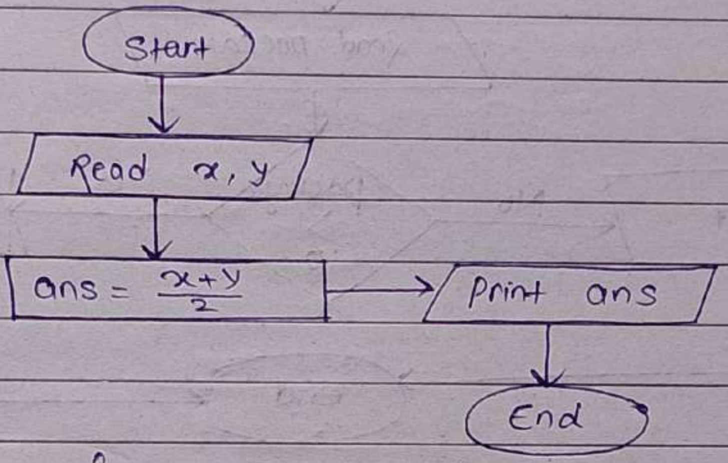
Add 2 number.

→ Take input of two number
→ sum both number
→ Return / show the result.



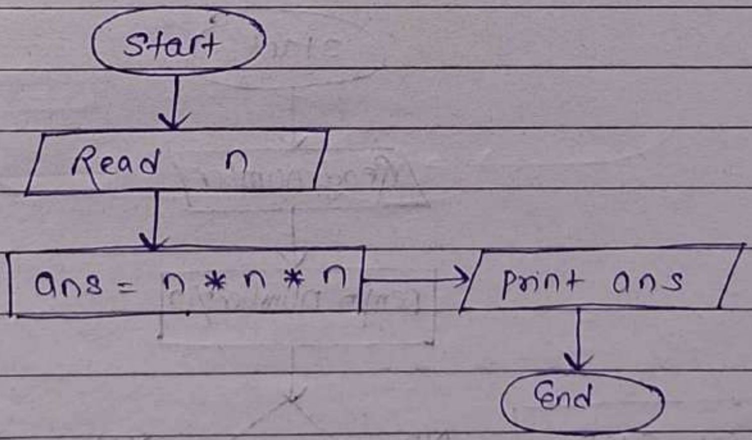
Example : 3 Average of 2 Number

pseudoCode
```
→ Take input of two number
→ sum both number and divide by 2
→ show the result.
```

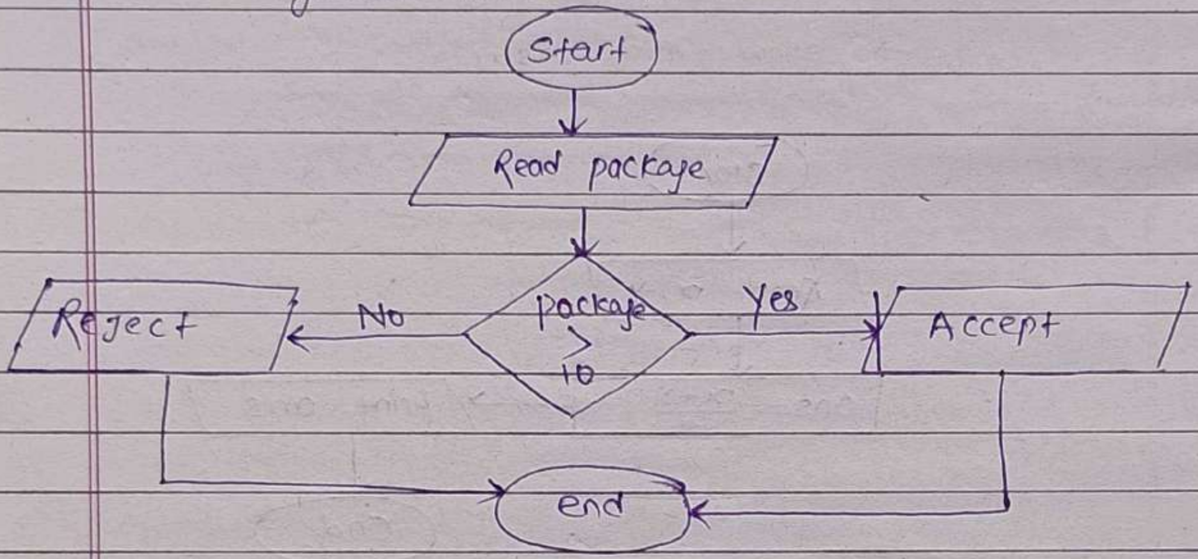


Example : 4 find the Cube of number

pseudoCode
```
→ Take input x
→ find Cube by num * num * num
→ print the result.
```



Example - 5 : You got an offerletter from X Company. You will accept offer only when package is greater than 10LPA.



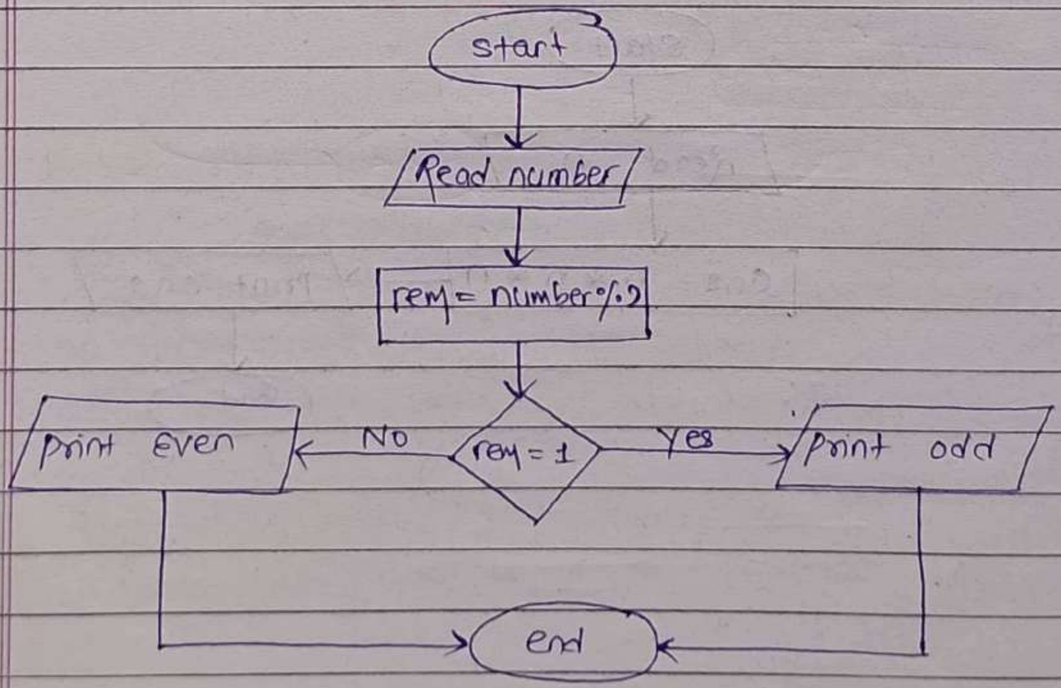
Example-6 Check number is Even or odd

→ Input number

→ If number % 2 is 0, then even

→ If number %.2 is 1, then Odd

→ end.



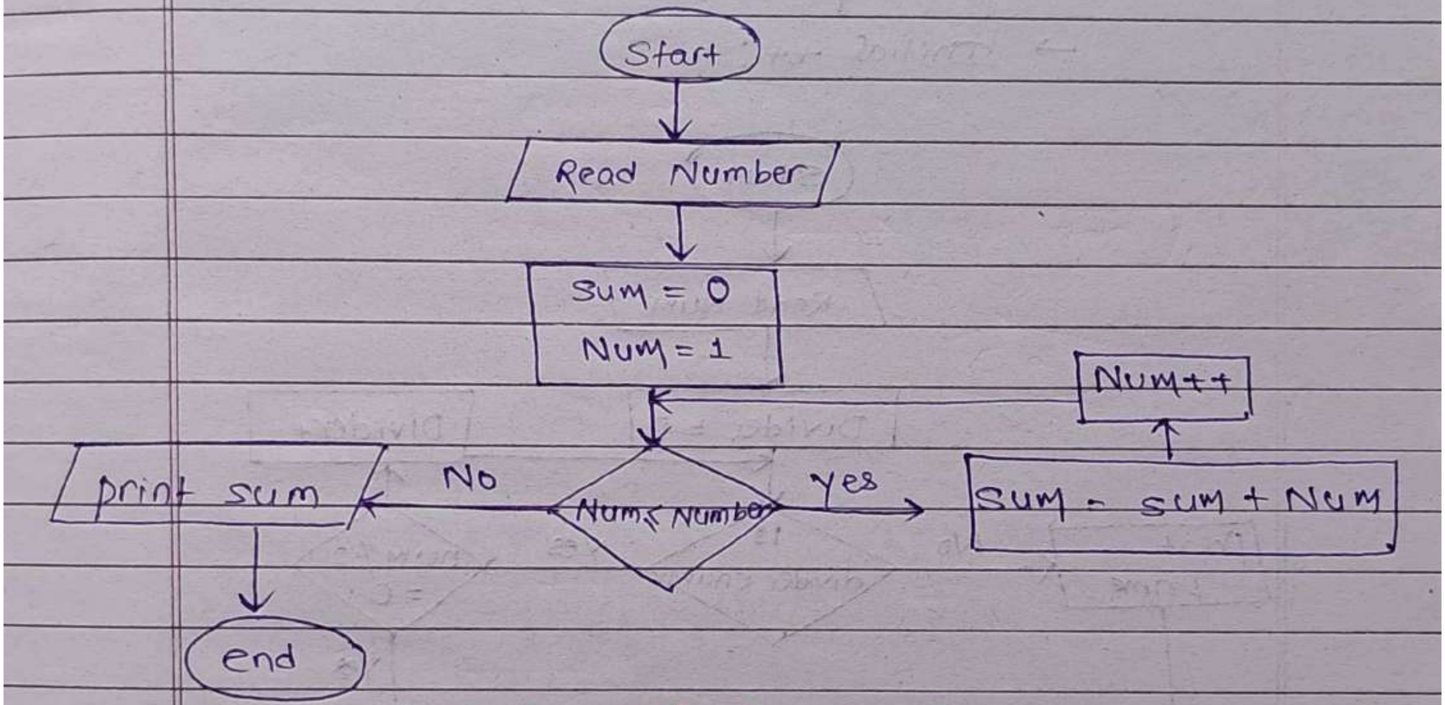
Example - 7 sum of n natural number

→ Take input

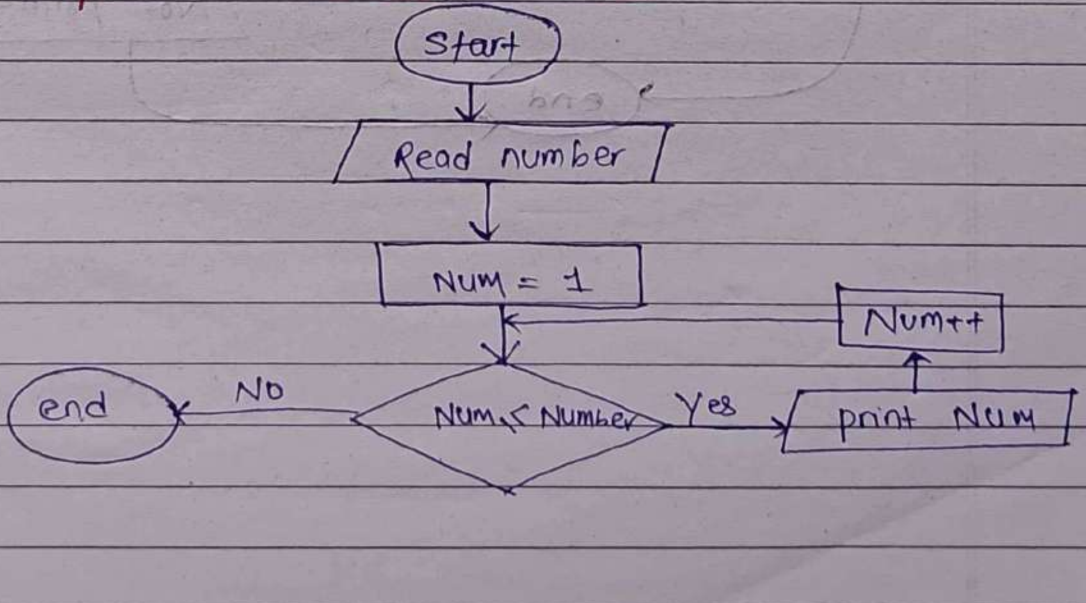
→ Take 1 by 1 number from 1.

→ Add the number in sum.

→ Repeat thės process until n reached.



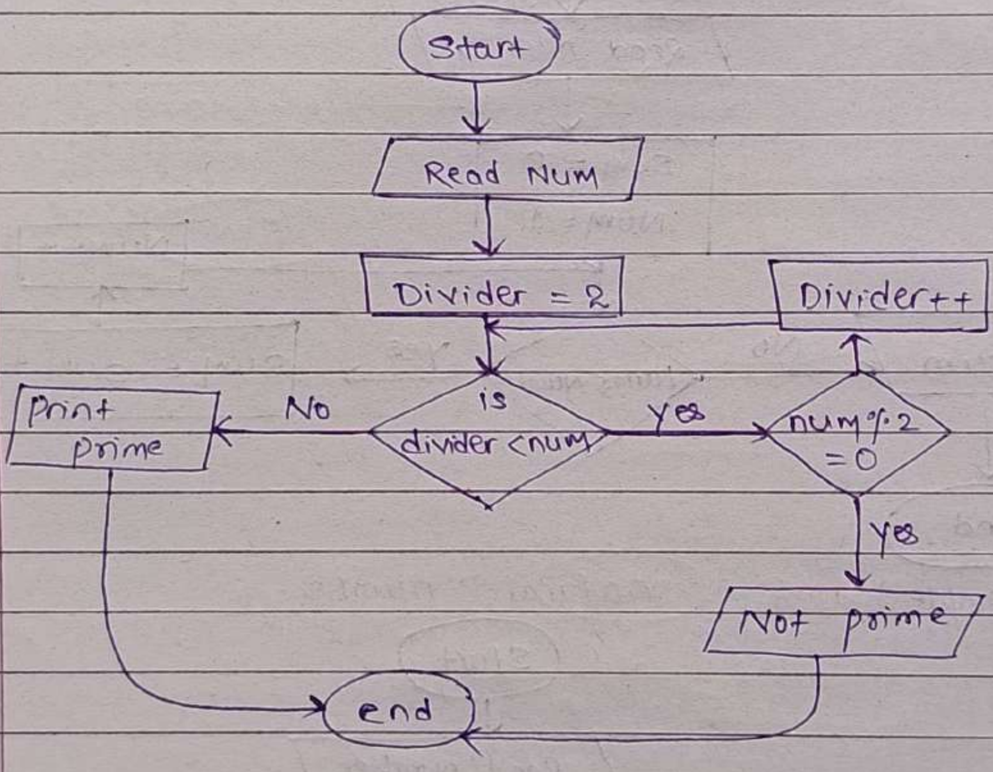
Example : print n natural number



→ Divided by two number and both are different.

* Check for prime number :

- → Take input n.
- → Take Initial = 2
- → check n % Initial == 0
- → of 0, then print Not prime, return.
- → else prime
- → Initial ++;


Start
Read Num
Divider = 2
Divider++
is
divider <num
No
yes
Print prime
num % 2 = 0
yes
Not prime
end