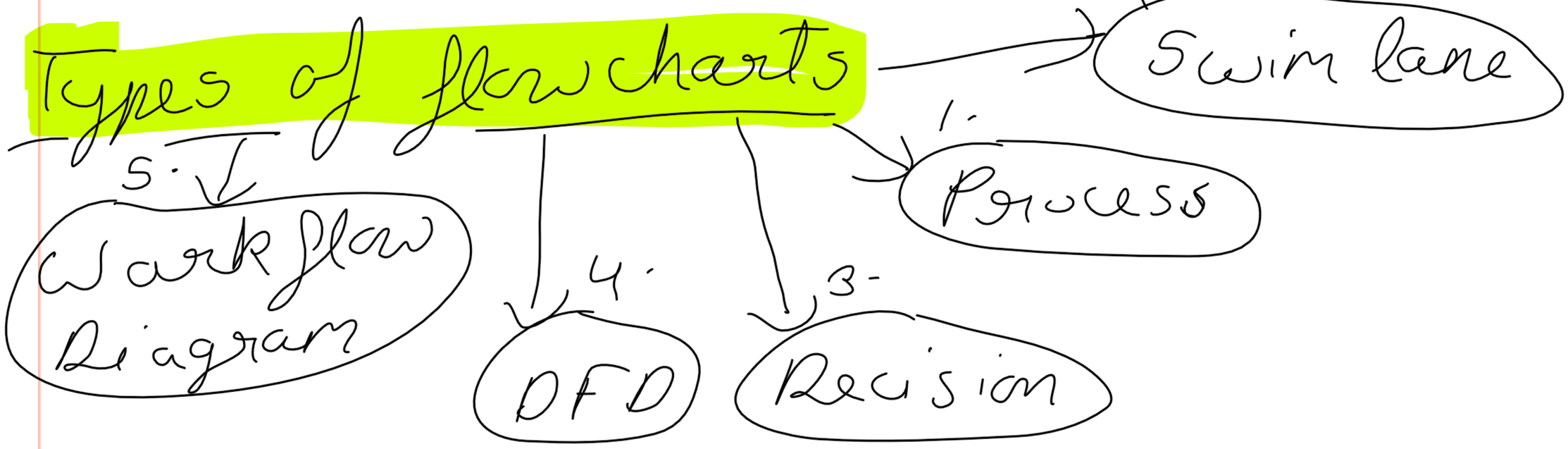
Introduction to flowcharts

(flowcharts) → diagrammatic way of expressing an algorithm.

Why use flowcharts? → Make complicated process simple

* universal visual language.
* great for documentation.
* helps to see problem clearly.

Types of flowcharts



Symbols used in flowchart Design

1.

Start/End

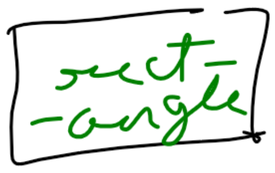
↳ shows start and end points

2.

Input/output

↳ gets required data

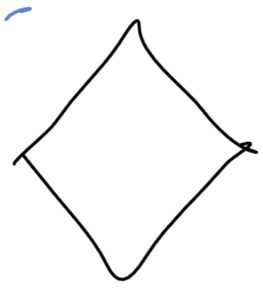
3.



Action/process

↳ indicates specify task

4.



Decision

↳ shows point where there is a choice.

5. → Arrow ⇒ shows flow of steps.

Rules for flowchart

1. (Start with 'start symbol') → 2. (End with 'end symbol')

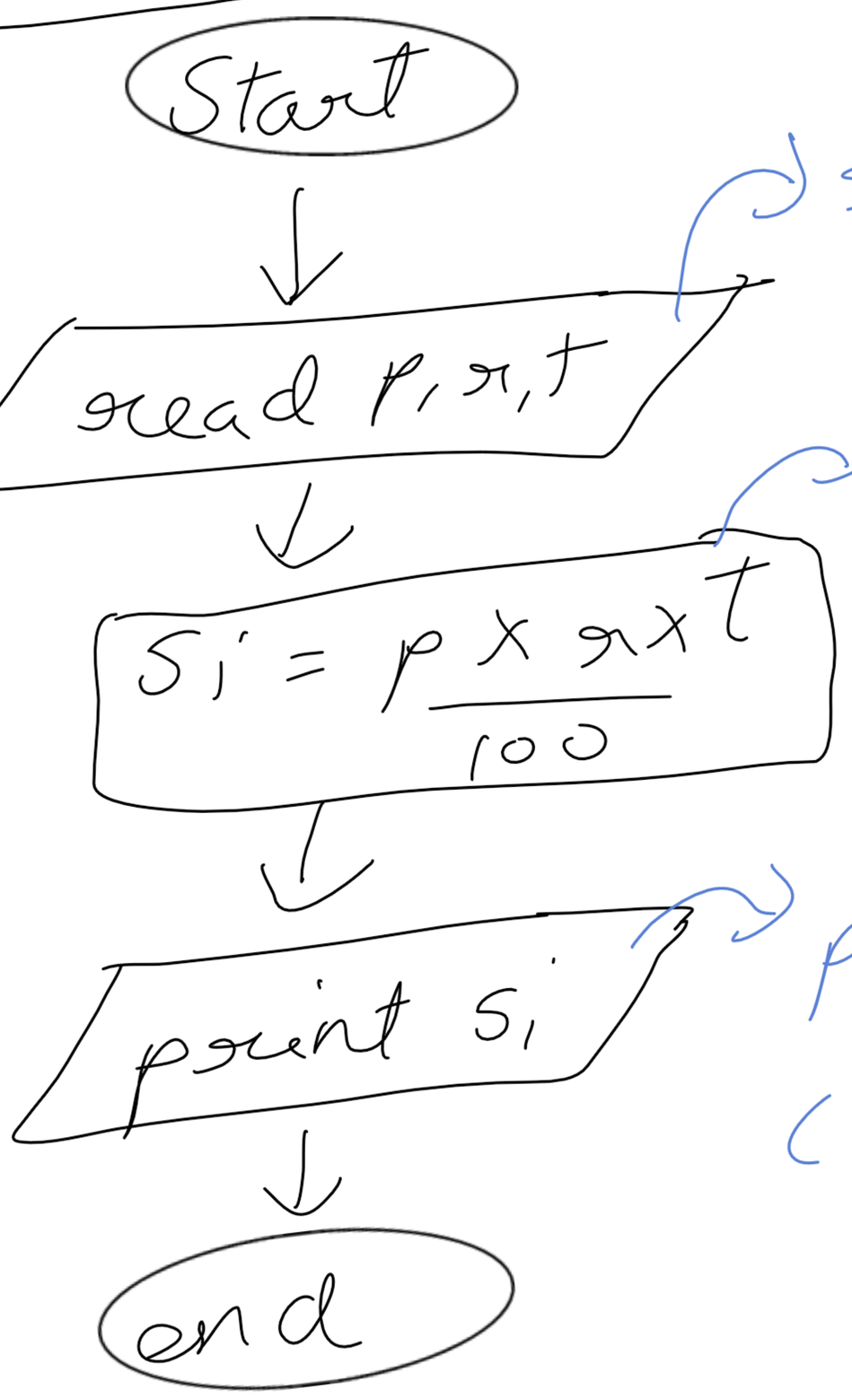
3. (Connect all symbol with arrow)

4. (Decision must have two or more path leading out)

5. (flow diagram flows from top to bottom)

Example of flowchart

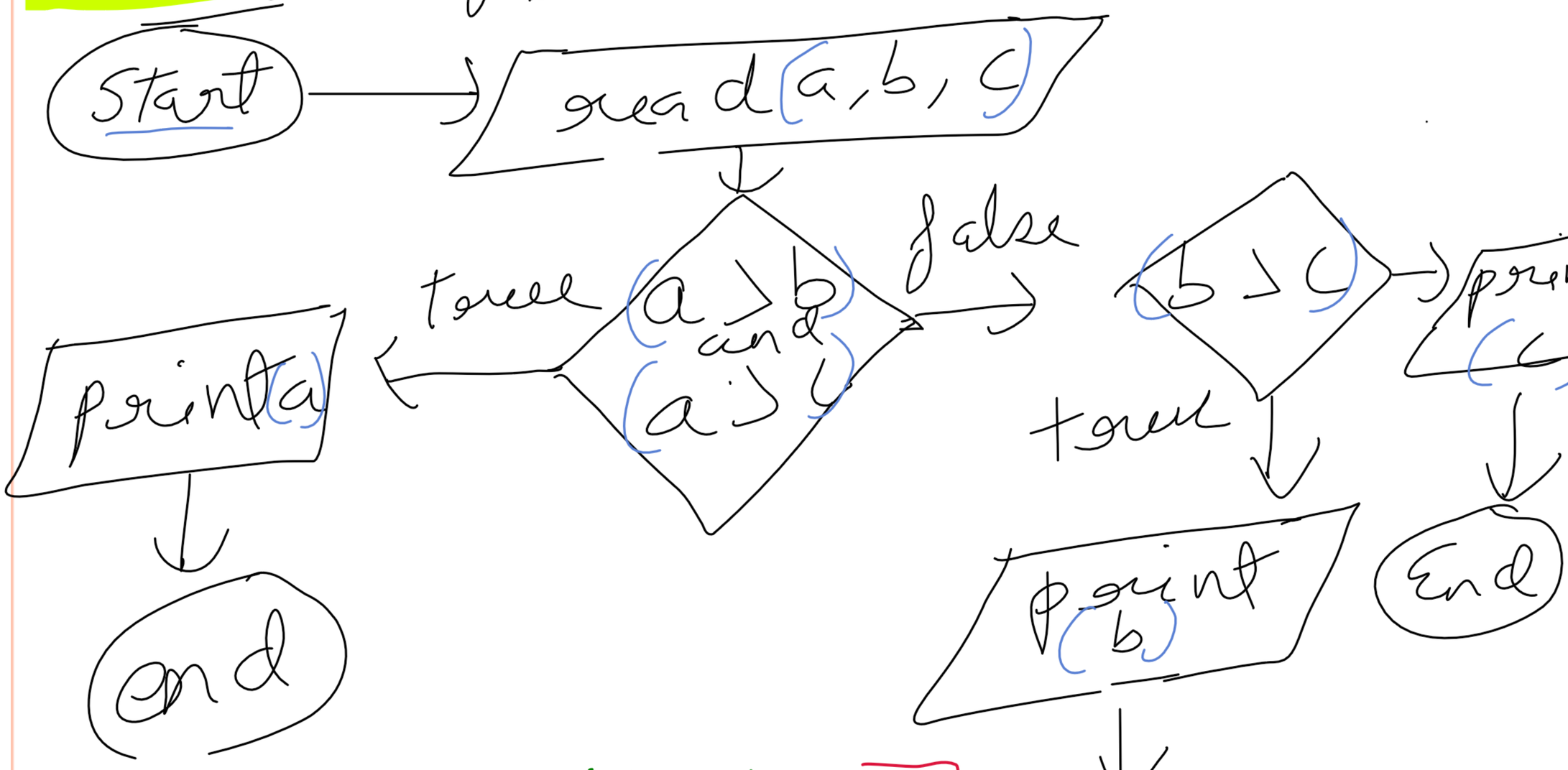
1. Simple Interest



let
p = 10
r = 2
t = 1
SI = 0.2
↓
10 x 2 x 1 / 100
= 0.2

[Use SI formula p*r*t]

2. largest of three Numbers :-



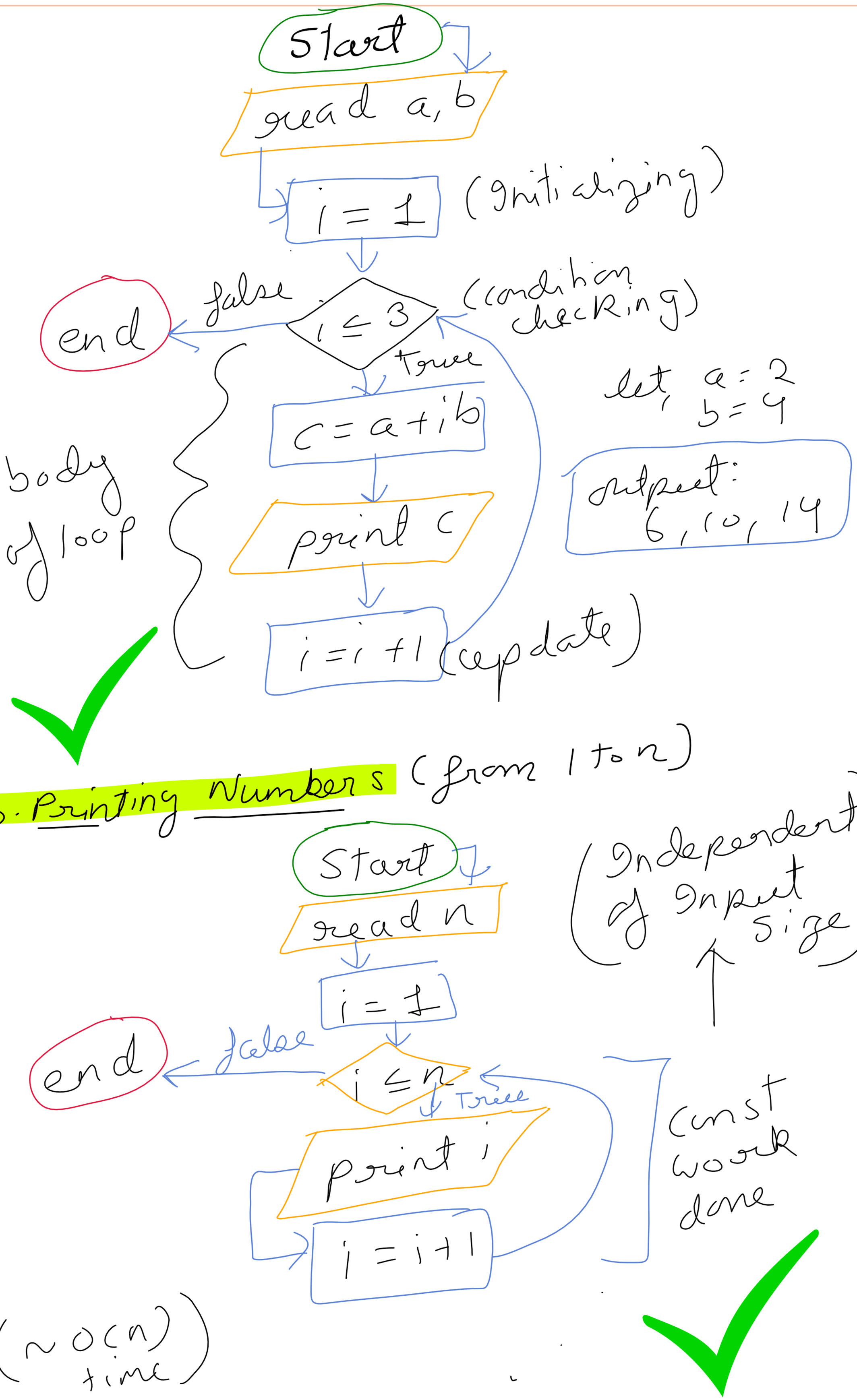
[firstly check if a is largest, if not then check b, if not then c is largest]

3. Point this series

Given 'a' and 'b', print the following :- [a+b, a+2*b, a+3*b].

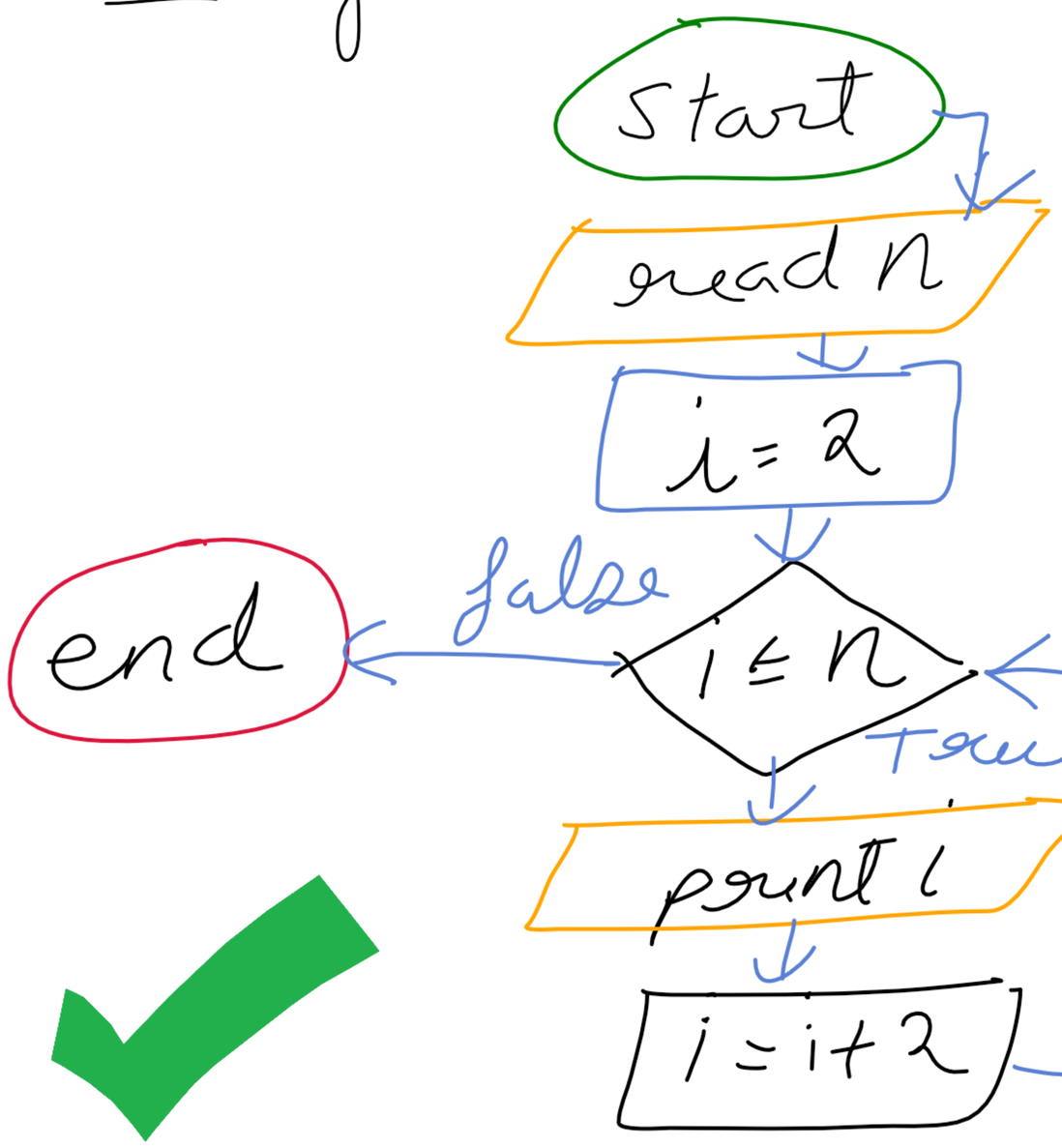
[loop → programming construct which allows to do the same task repeatedly while a condition is true] To create loop :-

* Initialize loop variable
* Condition checking.
* Body of the loop.
* Update loop Variable.



3. Printing Numbers (from 1 to n)

4. Printing Even Numbers (from 2 to n)

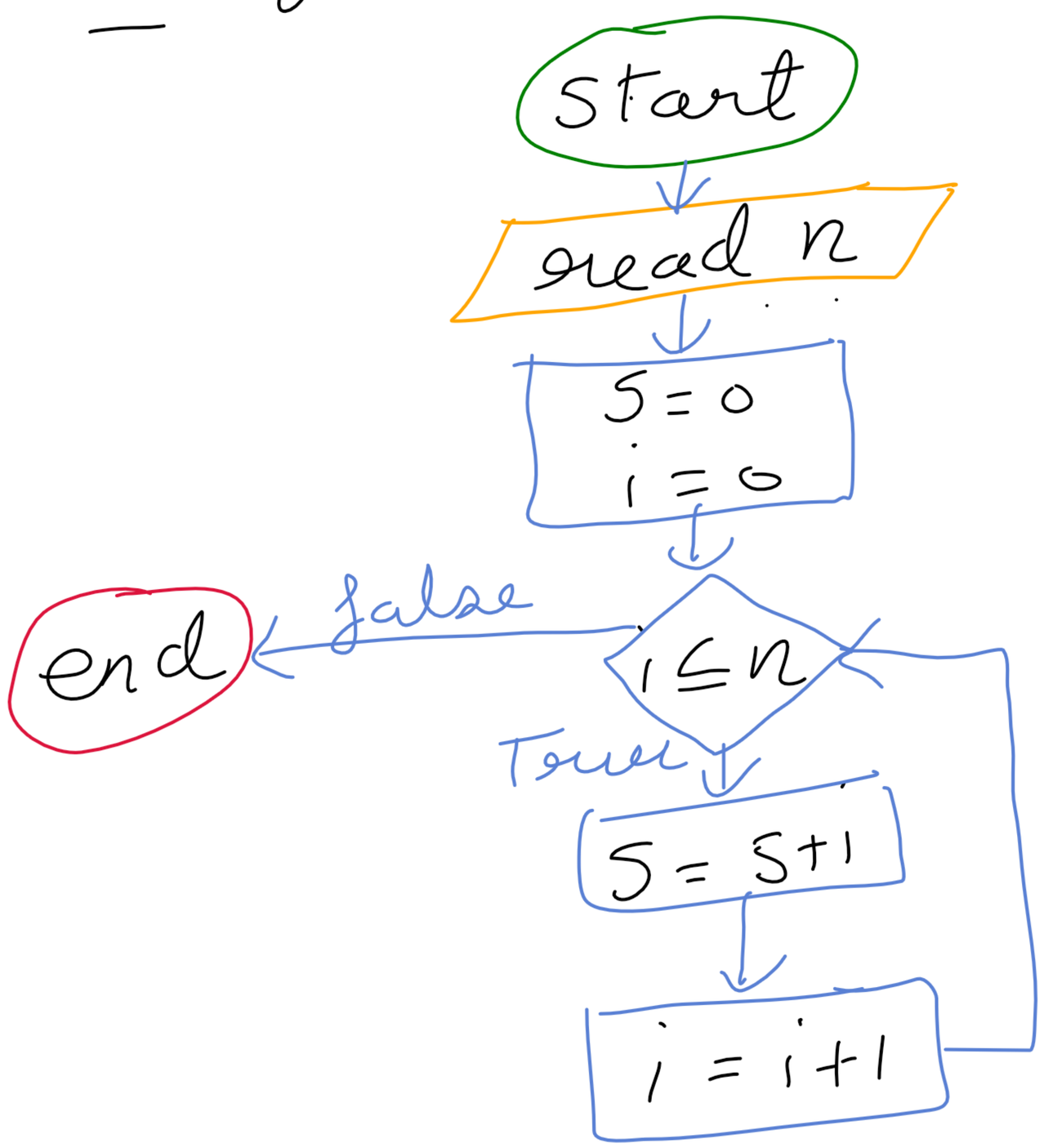


(time ~ O(N))

[# iterations = n/2]

(const work done)

5. Sum of 'n' natural numbers



(time: O(n))

$\left[\frac{n(n+1)}{2}\right]$

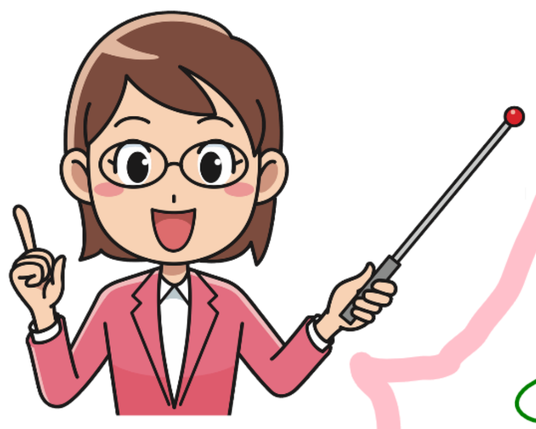
↓

This formula can be used

[Input : 10
Output : 55]

Create a 'S' variable, iterate through 1 to n, add iterator to 'S'

6. Check Prime Numbers



Assume 'n' is a composite Number,

By definition that means, 'n' can be factored into two int a and b.

$n = a * b$, where a and b are > 1 and $< n$.

$\therefore$ Case I: $a = b$

$\Rightarrow n = a^2$

$\Rightarrow a = \sqrt{n}$

Case II: $a \neq b$

$\Rightarrow n = a * b$

$\Rightarrow$ one factor must be $< \sqrt{n}$ and another $> \sqrt{n}$.

$a^2 \leq ab \leq b^2$

$a^2 \leq n \leq b^2$

$* \quad a \leq \sqrt{n} \leq b$

Range Optimized for checking prime number: $[2, \sqrt{n}]$

For eg: ① $n = 20$,

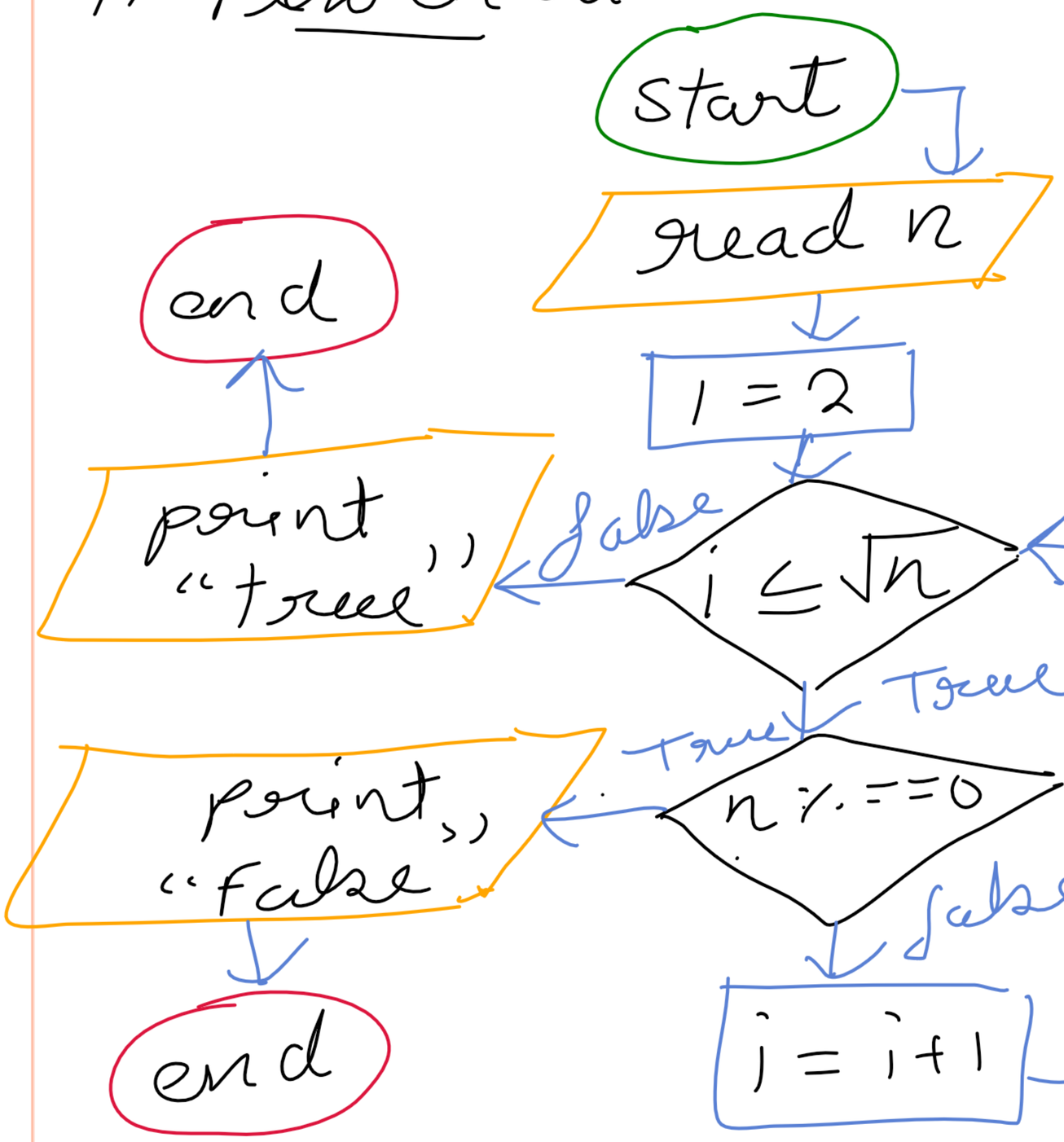
factors = 1 2 4 5 10 20

(check upto here)

② $n = 36$

factors = 1 2 3 4 6 6 9 12 18 36

Flow chart



(worst-case if n is prime)

[time ~ O(n)]

[# Iterations = 2 − √n
work done const]