Computational thinking

→ Ability to understand how problems will be solved by computer.

→ It increases with time when you see more & more examples.

Computational thinking (CT) is the mental skill to apply concepts, methods, problem solving techniques, and logic reasoning, derived from computing and computer science, to solve problems in all areas, including our daily lives.[1] In education, CT is a set of problem-solving methods that involve expressing problems and their solutions in ways that a computer could also execute.[2] It involves automation of processes, but also using computing to explore, analyze, and understand processes (natural and artificial).[3][4][5]

Computer Programming

Computer programming is the process of performing a particular computation (or more generally, accomplishing a specific computing result), usually by designing and building an executable computer program. Programming involves tasks such as analysis, generating algorithms, profiling algorithms' accuracy and resource consumption, and the implementation of algorithms (usually in a chosen programming language, commonly referred to as **coding**).[1][2]

→ Analysis
→ Generating Algorithm
→ Profiling Algo
→ Coding

Algorithm / Algo

In mathematics and computer science, an **algorithm** (/ˈælgərɪðəm/ (listen)) is a finite sequence of rigorous instructions, typically used to solve a class of specific problems or to perform a computation.[1] [redacted]

Real life example for understanding purpose

#1 How to make Maggi

Steps ① Put water
② Put maggi
③ Boil it for 3 minutes.

Recipe ≃ Algo

Recipe → Food

Algo → Computer problems

Algo can be represented by

1) English Statements

2) Flowchart

3) Code

Programming language

→ It's an language which both computer & human can understand

example - Python, Java, C++, Javascript

Language computer can understand feasible

↓

Binary language

Language only human understand

↓

English, Hindi, etc.

Computer Program/Code

A **computer program** is a sequence or set of instructions in a programming language for a computer to execute. Computer programs are one component of software, which also includes documentation and other intangible components.[1]

Programming Language = Python

Code ✓

```
list_of_numbers = [1,2,3,30,20,15,23]

for no in list_of_numbers:
    if no%2 == 0:
        print(f'Even={no}')
    else:
        print(f'Odd={no}')
```

```python
list_of_numbers = [1,2,3,30,20,15,23]

for no in list_of_numbers:
    if no%2 == 0:
        print(f'Even={no}')
    else:
        print(f'Odd={no}')
```

Piece of code

```
Odd=1
Even=2
Odd=3
Even=30
Even=20
Odd=15
Odd=23
```

Output

Q1. Why do we use computers? What is a computer? Why did we invented copy-paste machine (pun intended :))

→ To do computations which you & me can not do.

→ Make life Easy → Solving Problems

Q2. How do I know whether a problem can be solved by a computer?

→ 5+3+2

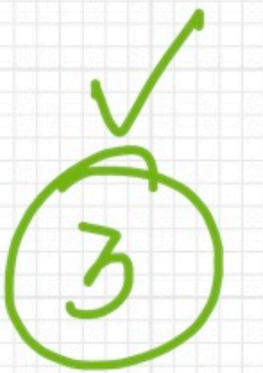
→ 8 +2 = 10

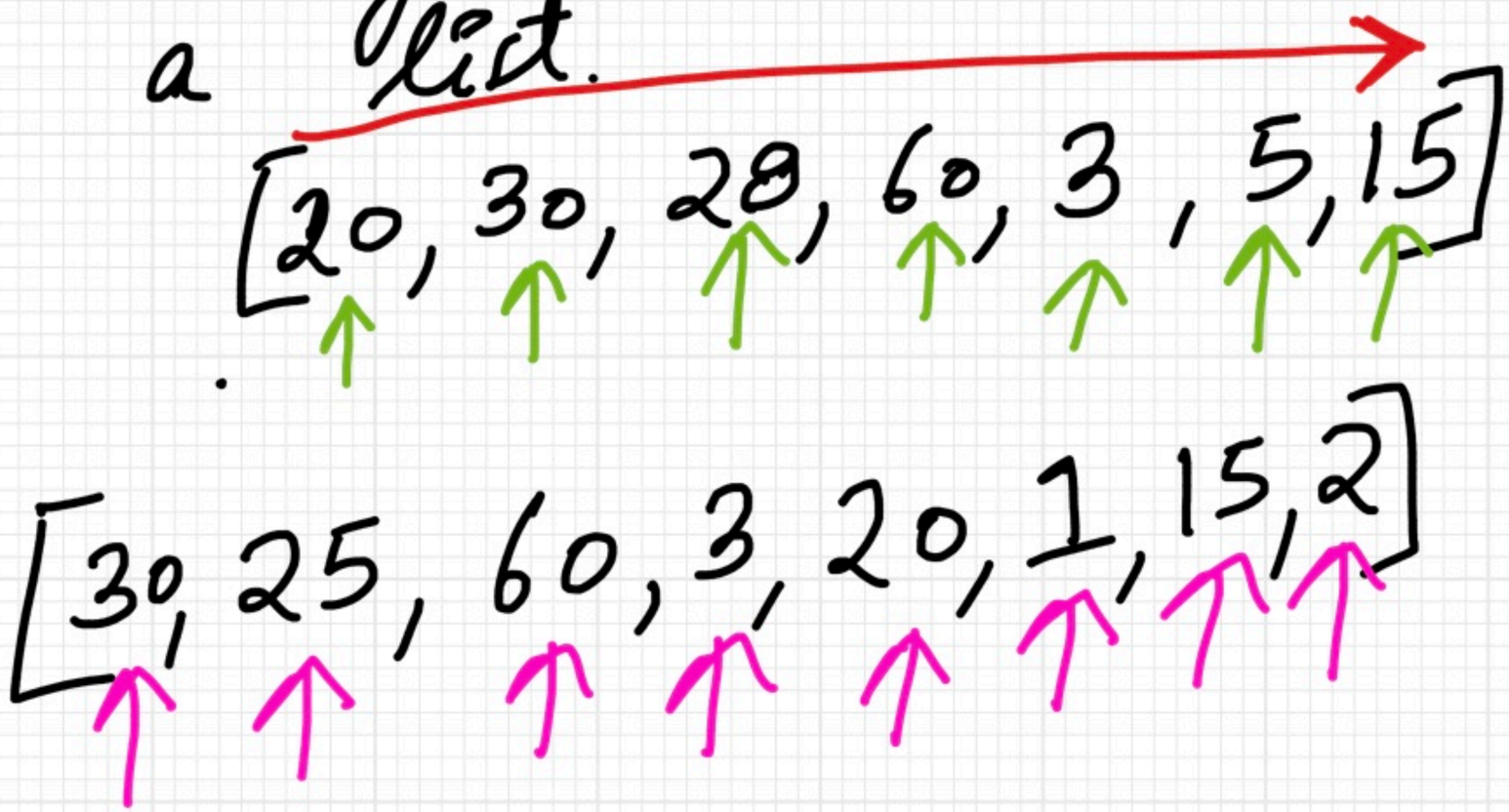
5+ 3+2

→ 5 + 5 = 10

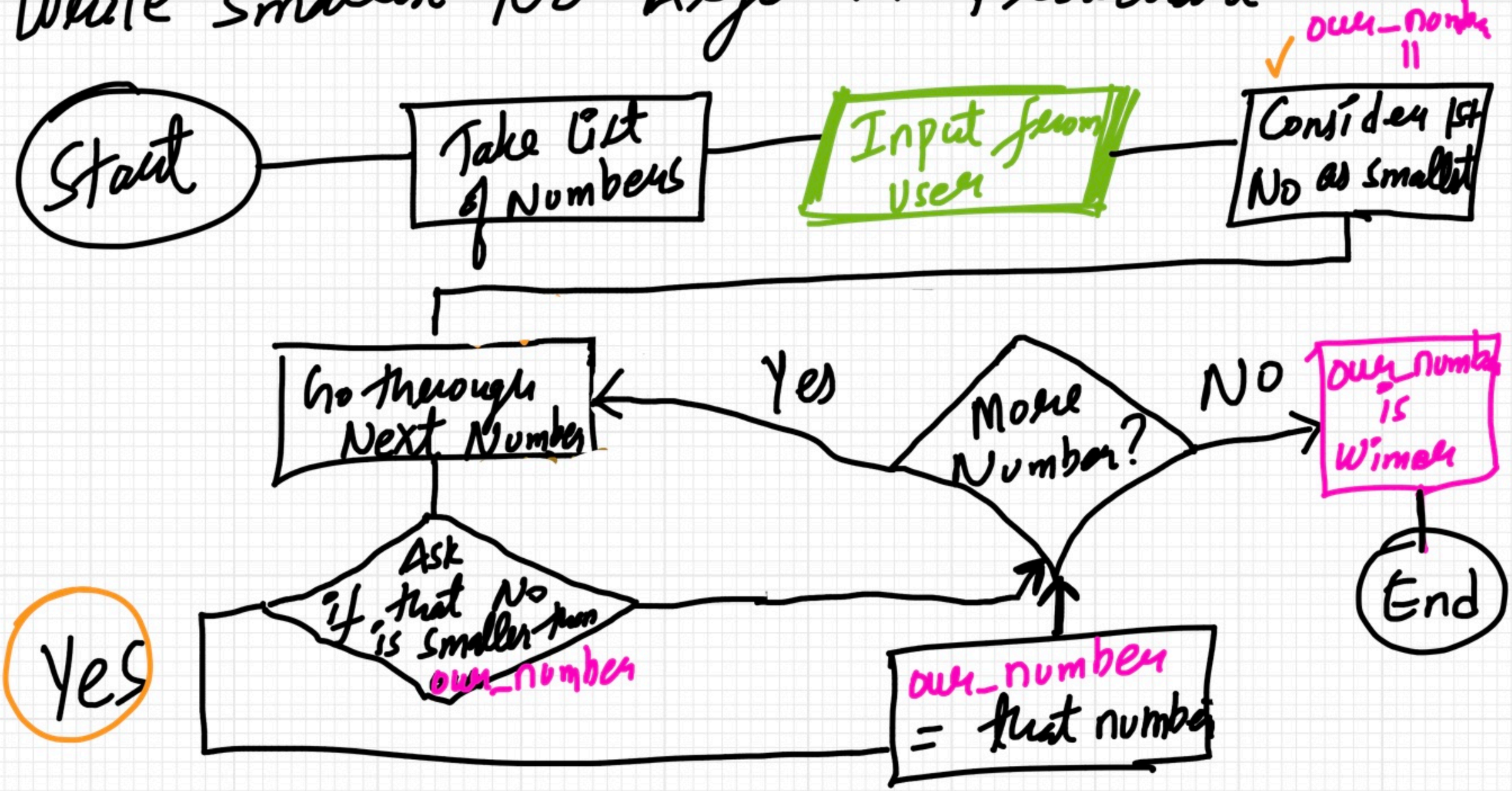
→ Computational Thinking Ability
≃ Knowledge with time & Practice.

Problem 1) Finding out smallest Number from a list.





Write Smallest No Algo in Flowchart



Let's use jupyter notebook (IDE like Software) to write computer program (code) & execute (run) it.

```
def find_smallest_no(list_of_numbers):
    our_number = list_of_numbers[0]

    for no in list_of_numbers:
        if no < our_number:
            our_number = no

    print(our_number)
```

Python Code

```
find_smallest_no([20,30,28,60,3,5,15])
```

output {

```
3
```

```
find_smallest_no([30,25,60,3,20,1,15,2])
```

output {

```
1
```

Problem 2) Arranging numbers in Ascending order.

H.W) Flowchart for the above algo & discussed.

Python program to arrange number (using previous knowledge)

```
list_of_numbers = [23,15,50,65,53]
ascending_order_list = []

while len(list_of_numbers)!= 0:
    smallest_number = find_smallest_no(list_of_numbers)
    ascending_order_list.append(smallest_number)
    list_of_numbers.remove(smallest_number)

print(ascending_order_list)
```

```
[15, 23, 50, 53, 65]
```