

# Intro into Algorithms & Problem Solving

Monday, 21 July, 2025 8:50 PM

## Pseudocode :

It is a way to describe the program using human readable languages without strict syntax of programming languages.

What it is - It is a Step by step outline of how an algorithm works.

Written in Structured English using keywords like IF, ELSE, ELSEIF, RETURN

Not executable by a computer but helps to understand.

Why use it - Helps to plan before writing real code (acts as a bridge between idea and code).

Helps to communicate ideas to team members.

Easier to Design and Debug algorithms.

How to write - They begin with a START and ends with END.

Algorithm goes in between.

Eg:

```
START
    A=529
    B=256
    SUM=A+B
    Display SUM
END
```

## Flowchart:

It is the visual representation of a program or an algorithm. It uses signs or symbols to show how tasks, decisions and actions flow from one to the next (which makes it easier to understand complex programs).

Why use it - Clarifies processes for better understanding.

Identifies inefficiencies or bottlenecks (bottlenecks are the places which slow down a program).

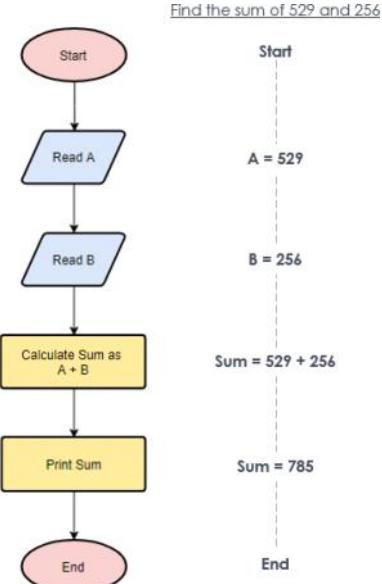
Improves communication with the team.

Helps in planning and decision making.

## Common Flowchart Symbols -

Symbol	Meaning
Oval	Start or End of a process
Rectangle	A process or action step
Diamond	A decision point (Yes/No, True/False)
Parallelogram	Input or Output (e.g., data entry)
Arrow	Direction of flow

Eg:



#### History :

Charles Babbage - The Father of the Computer (Designed the Analytical Engine).

Ada Lovelace - The First Programmer.

Alan Turing - The Theoretical Genius (Proposed the idea of a universal machine) (Turing test) (Enigma used to break down codes in WWII).

#### Turing Test:

It is the classic concept of AI proposed by Alan Turing in 1950. It was proposed as the imitation game.

How the test works - A human Judge is engaged in a text-based conversation with a human and a computer.

When the judge can't tell which is the computer, the machine wins.

It's about indistinguishability between a computer and a human, not perfect answers.

What it tests - Natural language understanding, Contextual reasoning, Human like (not smart but how well its human like).

Contextual Reasoning - The ability of humans to make sense of the information.

Program is a collection of computer instructions or operations, grouped together in logical manner.

Algorithm lists the steps involved in accomplishing a task.

It's of detailed, unambiguous and ordered instructions developed to describe the processes necessary to produce the desired output.

Define the problem -

Divide the problem into 3 separate components.

Input - List of source data provided to the problem.

Output - List of outputs required.

Processing - List of actions required to process the outputs.

These 3 can be presented in a Defining Diagram (IPO Chart).

<b>Input</b>	<b>Processing</b>	<b>Output</b>

Steps to Developing a Problem :

Define - Define the Problem

    Define the Problem using Defining Diagram (IPO Chart).

Outline - Outline the Solution

    Certain details are identified by analyzing the problem further.

    Like Major processing tasks, subtasks, control structures, variables, mainline logic.

Develop - Develop the outline into an algorithm

    Using Pseudocodes and Flowcharts.

Test - Test the algorithm for correctness

Code - Code the algorithm into a specific programming language

Run - Run the program on a computer

Document and maintain - Document and Maintain the Program

Six Basic Computer Operations :

Input - Receive Information

Output - Produce Information

Perform Arithmetic

Assign a value to a piece of data

Compare two pieces of information

Repeat a group of actions

### Relational Operations :

Used to compare numbers to determine relative order.

Operator	Action
<	Lesser Than
>	Greater Than
==	Equal To
<=	Lesser Than or Equal To
>=	Greater Than or Equal To
!=	Not Equal

### Relational Expressions :

Boolean Expressions - True or False

Eg:

```
12 > 5 is True
7 <= 5 is False
If x is 10 then
    x == 10 is True
    x != 8 is True
    x == 8 is False
```

### Logical Operations :

Used to create relational expressions from other relational expressions.

Operation	Action	Definition
and	AND	New relational expression is True if <u>both</u> expressions are True.
or	OR	New relational expression is True if <u>either</u> expression is True.
not	NOT	<u>Reverses the value</u> of an expression - True -> False False -> True

Eg: x = 12, y = 5, z = -4

(x > y) and (y > z)	True
(x > y) and (z > y)	False
(x <= z) or (y == z)	False
(x <= z) or (y != z)	True
NOT (x >= z)	False

### Truth Table :

AND

P	Q	P AND Q
T	T	T
T	F	F
F	T	F
F	F	F

OR

P	Q	P OR Q
T	T	T
T	F	T
F	T	T
F	F	F

### Selection Structure :

Simple IF Statement -

A choice is made between 2 alternate paths, depending on the result of a condition being true or false.

Represented by,

IF, ELSE, THEN, ENDIF

```
Eg: IF account_balance < 300 THEN
    service_charge = 5.00
ELSE
    service_charge = 2.00
ENDIF
```

#### Null ELSE Statement -

It is a variation of simple IF structure.  
Used when a task is performed when a particular condition is true.  
If the condition is false, then no processing takes place and the IF statement will be bypassed.

```
Eg: IF student_attendance == part_time THEN
    add 1 to part_time_count
ENDIF:
```

#### Combined IF Statement -

It contains multiple conditions.  
Each condition is connected using logical operators AND or OR.

```
Eg 1. IF ((condition_1) AND (condition_2)) THEN
    Action_1
ELSE
    Action_2
ENDIF
```

```
Eg 2. IF ((condition_1) OR (condition_2)) THEN
    Action_1
ELSE
    Action_2
ENDIF
```

#### Nested Selection (Nested IF Statements) -

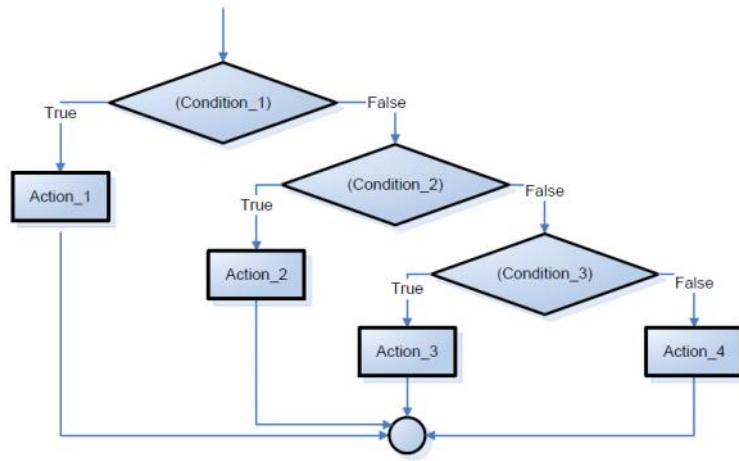
Occurs when the word IF appears more than once within an IF statement.

They can be considered as Linear and Non-Linear.

##### Linear -

Used when a field is being tested for various values and a different action is to be taken for each value.  
This form of nested IF is called linear, because each ELSE immediately follows the IF condition to which it corresponds.  
Comparisons are made until a true condition is encountered, and the specified action is executed until the next ELSE statement is reached.  
Linear nested IF statements should be indented for readability, with each IF, ELSE and corresponding ENDIF aligned.

```
Eg : IF condition
    statement
ELSE
    IF condition
        statement
    ELSE
        IF condition
            statement
        ELSE
            statement
        ENDIF
    ENDIF
ENDIF
```

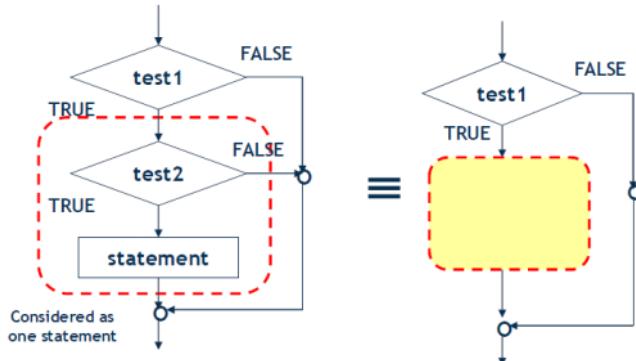


Non-Linear -

```

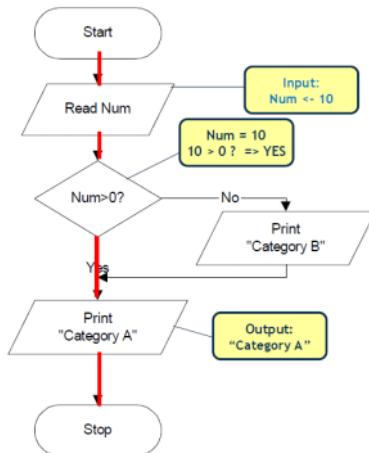
Eg : IF condition THEN
      IF condition THEN
          statement
      ELSE
          statement
      ENDIF
      ELSE
          statement
      ENDIF
  
```

### Nested if (if within if)



Trace -

Trace the flowchart to find its output.



## Repetition Structures :

Specifies a block of one or more statements that's repeatedly executed until a condition is satisfied.

Usually, loop has 2 important parts,

An expression that's tested for true/false.

A statement or block that repeated if the expression is true.

2 types of repetition or loop,

Per-test loop.

Post-test loop (Not in Python).

## Counters -

Can be used to control execution of a loop (loop control variable).

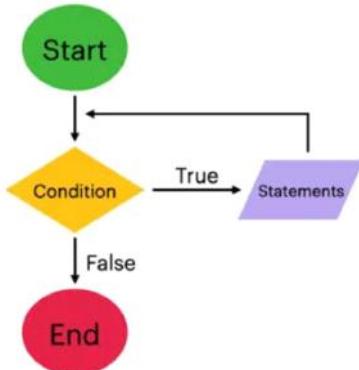
It increments or decrements each time the loop repeats.

Must be initialized before entering a loop.

## Pre-test loop (While & For loop) -

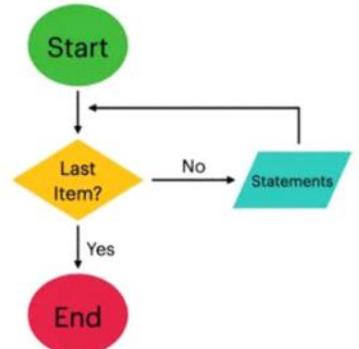
### While Loop -

```
WHILE condition  
    statement  
END WHILE
```



### For Loop -

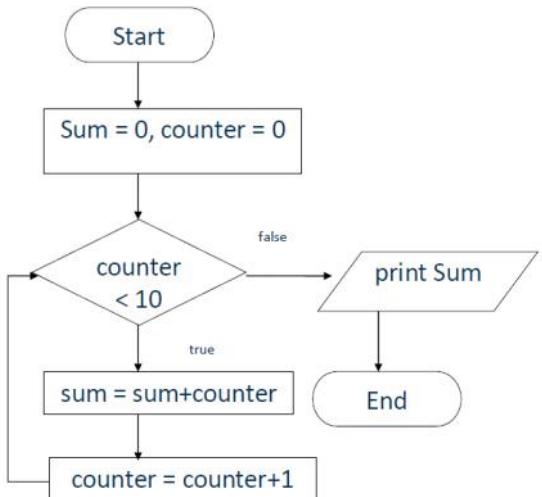
```
For condition  
    statement
```



## Post-test loop (Do while) -

Eg:

```
Start  
Set sum = 0, counter = 0  
Do  
    sum = sum + counter  
    counter = counter + 1  
    While (counter < 10)  
        Display sum  
    End
```



#### Sentinels -

It is the value in a list of values that indicates the end of data.

Special value that cannot be confused with a valid value (Eg, -999 for test score).

Used to terminate input when user may not know how many values will be entered.

Eg.

```

Start
  sum = 0
  read value
  while(value != 0)
    sum = sum + value
    read value
  print sum
End
  
```