# ALGORITHMS AND FLOWCHARTS

# ALGORITHMS AND FLOWCHARTS

- A typical programming task can be divided into two phases:
- ***Problem solving phase***
  - produce an ordered sequence of steps that describe solution of problem
  - this sequence of steps is called an ***algorithm***
- ***Implementation phase***
  - implement the program in some programming language

# Steps in Problem Solving

- First produce a general algorithm (one can use ***pseudocode***)
- Refine the algorithm successively to get step by step detailed ***algorithm*** that is very close to a computer language.
- ***Pseudocode*** is an artificial and informal language that helps programmers develop algorithms. Pseudocode is very similar to everyday English.

An **algorithm** must possess the following **properties**

**(i) Finiteness :** An algorithm must terminate in a finite number of steps.

**(ii) Definiteness :** Each step of the algorithm must be precisely and unambiguously stated.

**(iii) Effectiveness :** Each step must be effective, in the sense that it should be easily convertible into program statement.

- **(iv) Generality :** The algorithm must be complete in itself so that it can be used to solve all problems of a specific type for any input data.
- **(v) Input / Output :** Each algorithm must take zero, one or more inputs.

# Pseudocode & Algorithm

- **Example 1:** Write an algorithm to determine a student's final grade and indicate whether it is passing or failing. The final grade is calculated as the average of four marks.

# Pseudocode & Algorithm

**Pseudocode**:

- *Input a set of 4 marks*
- *Calculate their average by summing and dividing by 4*
- *if average is below 50*

  *Print “FAIL”*

  *else*

  *Print “PASS”*

# Pseudocode & Algorithm

- Detailed Algorithm
- Step 1: Input M1,M2,M3,M4
  Step 2: GRADE ← (M1+M2+M3+M4)/4
  Step 3: if (GRADE < 50) then
  Print “FAIL”
  else
  Print “PASS”
  endif

# The Flowchart

- A graphical representation of the sequence of operations in an information system or program. Information system flowcharts show how data flows from source documents through the computer to final distribution to users. Different symbols are used to draw each type of flowchart.

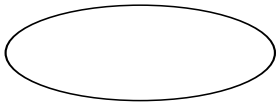
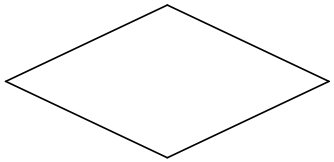
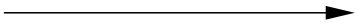
# The Flowchart

A Flowchart

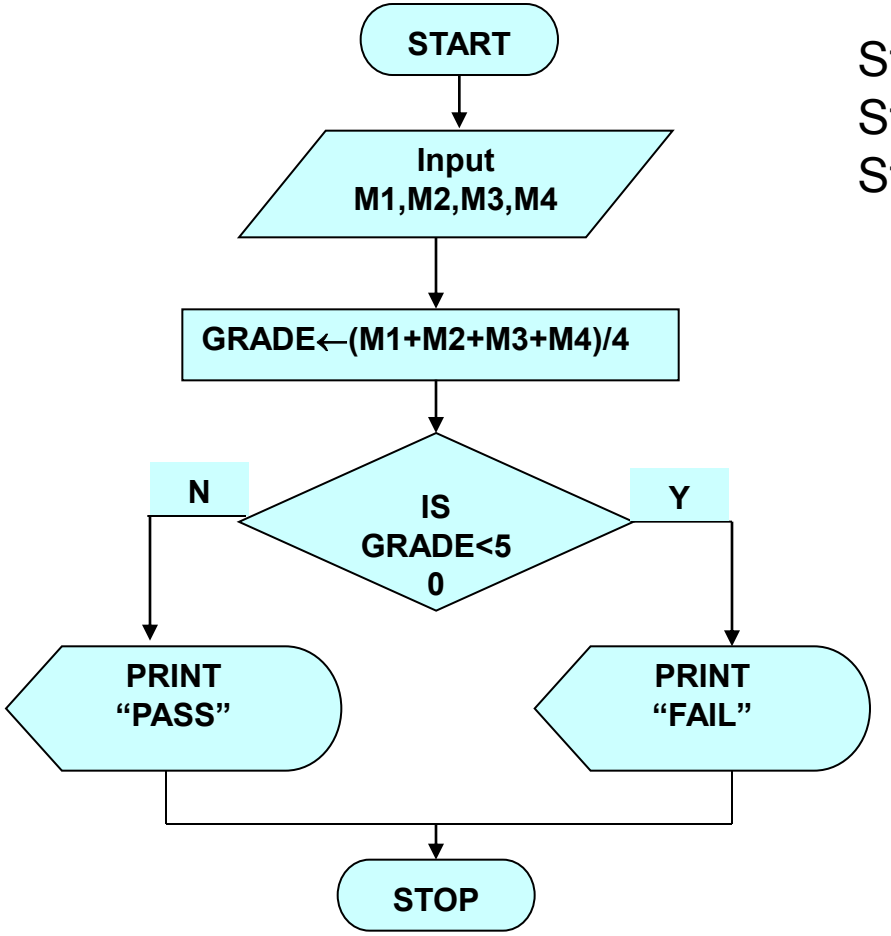
- shows logic of an algorithm
- emphasizes individual steps and their interconnections
- e.g. control flow from one action to the next

# Flowchart Symbols

## Basic

| Name | Symbol | Use in Flowchart |
| --- | --- | --- |
| Oval |  | Denotes the beginning or end of the program |
| Parallelogram |  | Denotes an input operation |
| Rectangle |  | Denotes a process to be carried out e.g. addition, subtraction, division etc. |
| Diamond |  | Denotes a decision (or branch) to be made. The program should continue along one of two routes. (e.g. IF/THEN/ELSE) |
| Hybrid |  | Denotes an output operation |
| Flow line |  | Denotes the direction of logic flow in the program |

# Example




Step 1: Input M1,M2,M3,M4
Step 2: GRADE ← (M1+M2+M3+M4)/4
Step 3: if (GRADE <50) then
Print “FAIL”
else
Print “PASS”
endif

# Example 2

- Write an algorithm and draw a flowchart to convert the length in feet to centimeter.

**Pseudocode**:

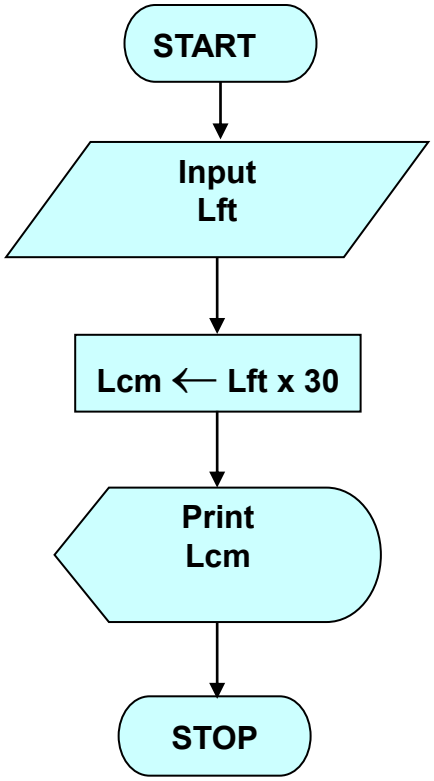
- *Input the length in feet (Lft)*
- *Calculate the length in cm (Lcm) by multiplying LFT with 30*
- *Print length in cm (LCM)*

# Example 2

**Algorithm**

- Step 1: Input Lft
- Step 2: Lcm ← Lft x 30
- Step 3: Print Lcm

**Flowchart**



# Example 3

**Write an algorithm and draw a flowchart that will read the two sides of a rectangle and calculate its area.**

**Pseudocode**

- *Input the width (W) and Length (L) of a rectangle*
- *Calculate the area (A) by multiplying L with W*
- *Print A*

# Example 3

**Algorithm**

- Step 1: Input W,L
- Step 2: A ← L x W
- Step 3: Print A