



# ALGORITHMS AND FLOWCHARTS

**Presented by: Mr. V. B. Pujari**

# 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.

# 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.

# Algorithm

- Detailed Algorithm

- Step 1: Input M1,M2,M3,M4
- Step 2:  $GRADE \leftarrow (M1+M2+M3+M4)/4$
- Step 3: if (GRADE < 50) then  
    Print "FAIL"  
else  
    Print "PASS"  
endif

# The Flowchart

- (Dictionary) A schematic representation of a sequence of operations, as in a manufacturing process or computer program.
- (Technical) 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.
  - Program flowcharts show the sequence of instructions in a single program or subroutine. Different symbols are used to draw each type of flowchart.



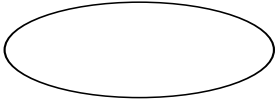


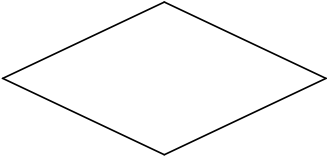
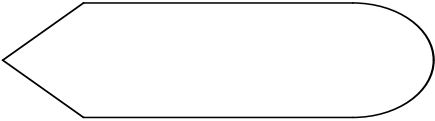

# The Flowchart

## A Flowchart

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

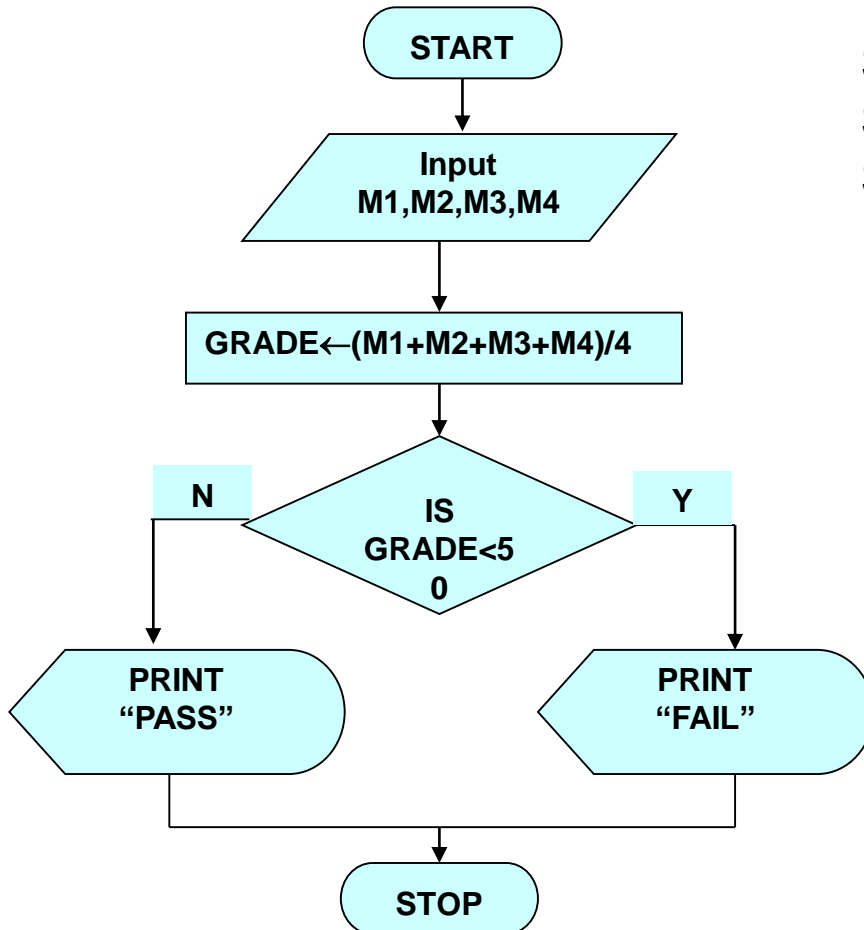
# Flowchart Symbols

## 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:  $GRADE \leftarrow (M1 + M2 + M3 + M4) / 4$   
Step 3: if (GRADE < 50) then  
    Print "FAIL"  
    else  
        Print "PASS"  
    endif



# Example 2

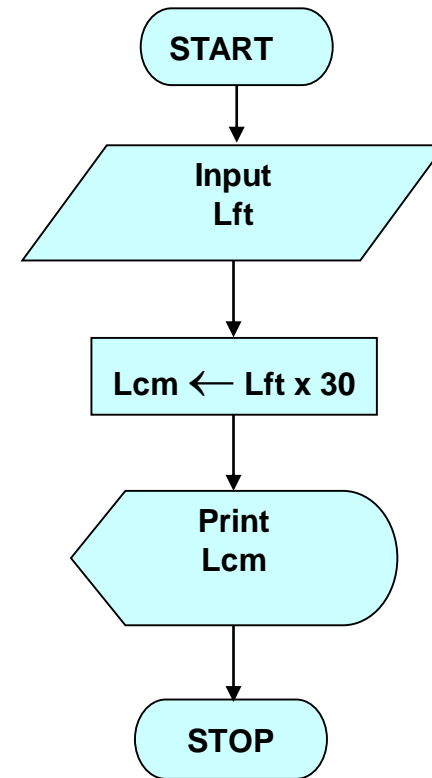
- Write an algorithm and draw a flowchart to convert the length in feet to centimeter.

# Example 2

## Algorithm

- Step 1: Input Lft
- Step 2:  $Lcm \leftarrow Lft \times 30$
- Step 3: Print Lcm

## Flowchart





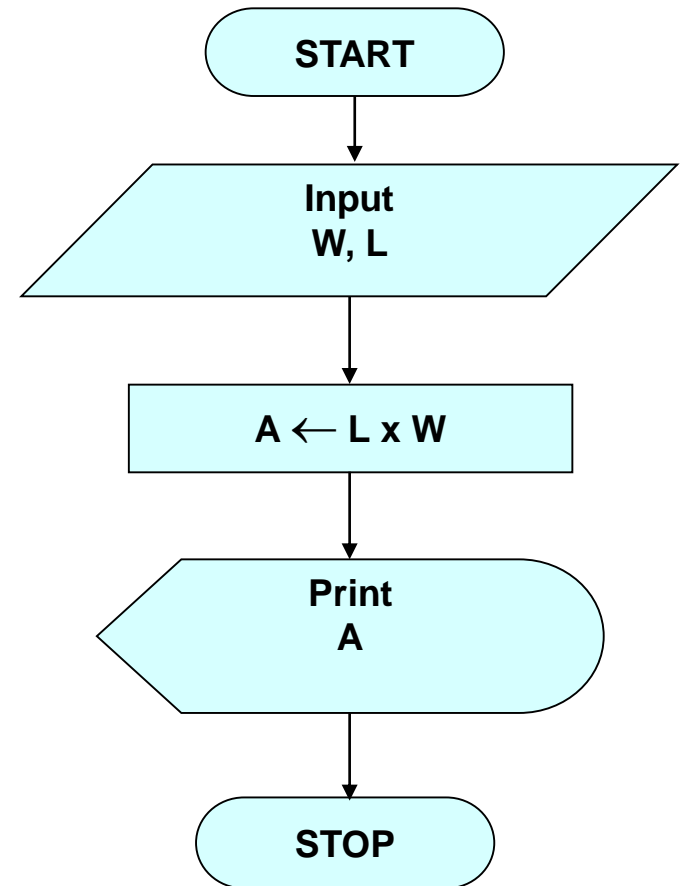
# Example 3

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

# Example 3

## Algorithm

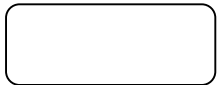
- Step 1: Input  $W, L$
- Step 2:  $A \leftarrow L \times W$
- Step 3: Print  $A$



# Flowcharts

- Flowcharts is a graph used to depict or show a step by step solution using **symbols** which represent a task.
- The symbols used consist of geometrical shapes that are connected by **flow lines**.
- It is an alternative to pseudocoding; whereas a pseudocode description is verbal, a flowchart is graphical in nature.

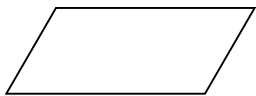
# Flowchart Symbols



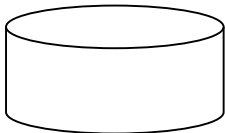
**Terminal symbol** - indicates the beginning and end points of an algorithm.



**Process symbol** - shows an instruction other than input, output or selection.



**Input-output symbol** - shows an input or an output operation.

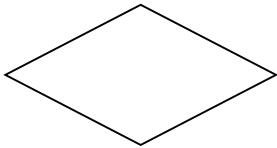


**Disk storage I/O symbol** - indicates input from or output to disk storage.

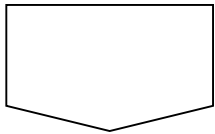


**Printer output symbol** - shows hardcopy printer output.

# Flowchart Symbols cont...



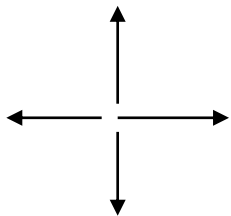
**Selection symbol** - shows a selection process for two-way selection.



**Off-page connector** - provides continuation of a logical path on another page.



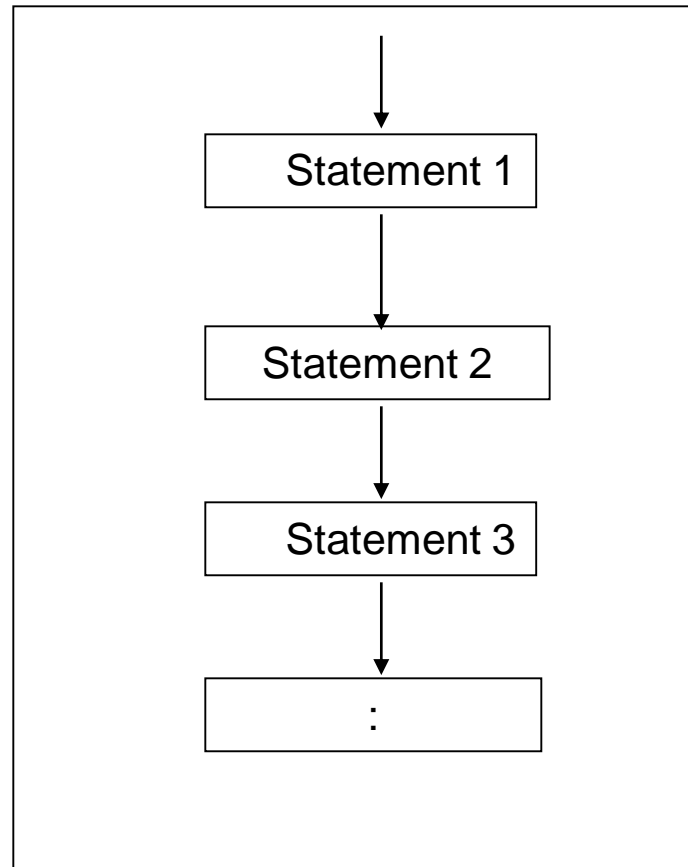
**On-page connector** - provides continuation of logical path at another point in the same page.



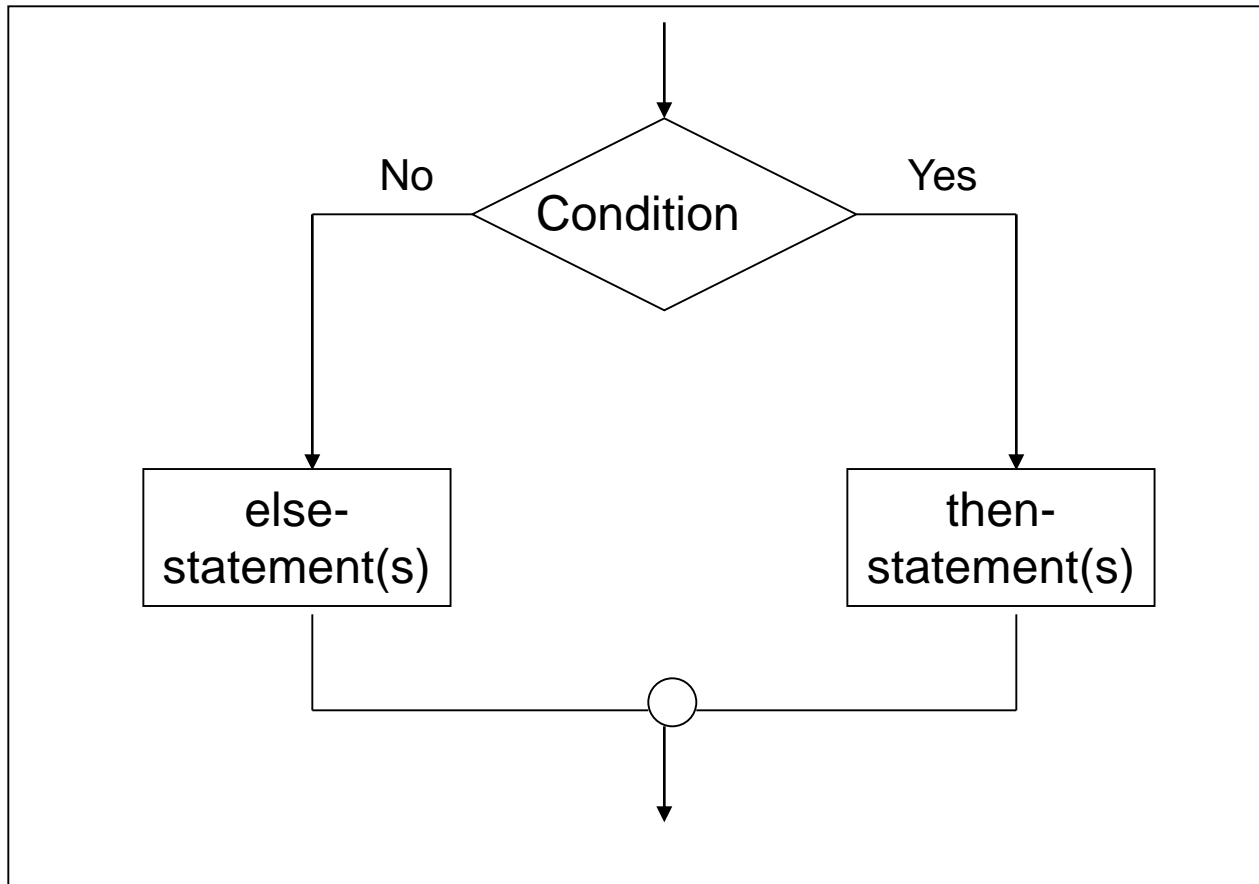
**Flow lines** - indicate the logical sequence of execution steps in the algorithm.



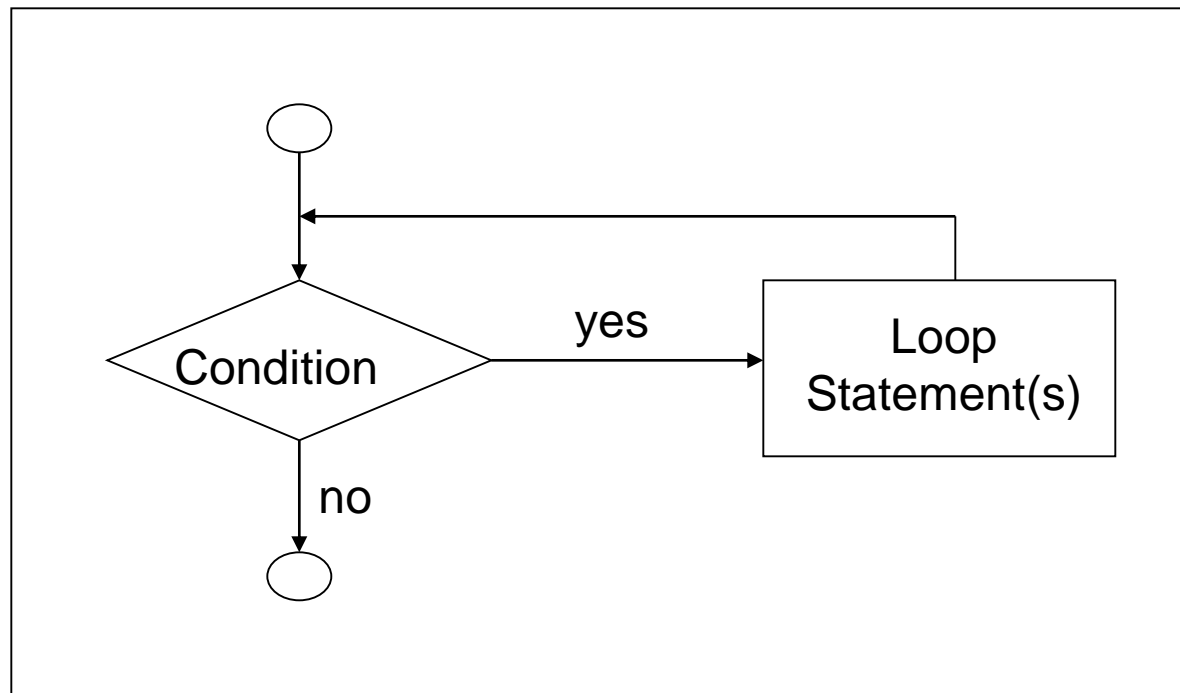
# Flowchart – sequence control structure



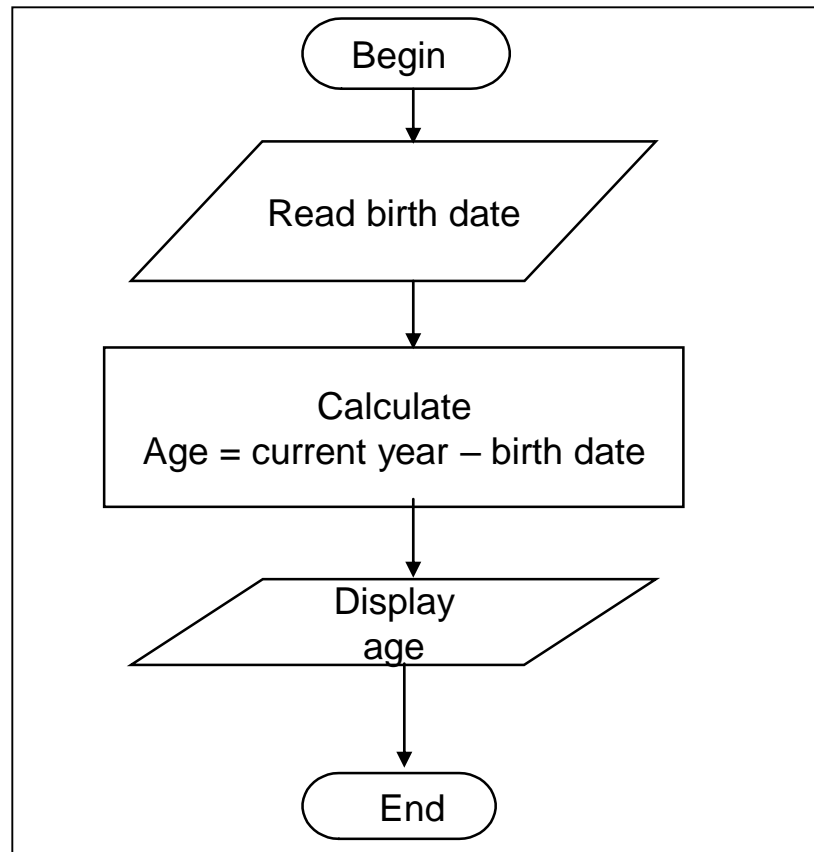
# Flowchart – selection control structure



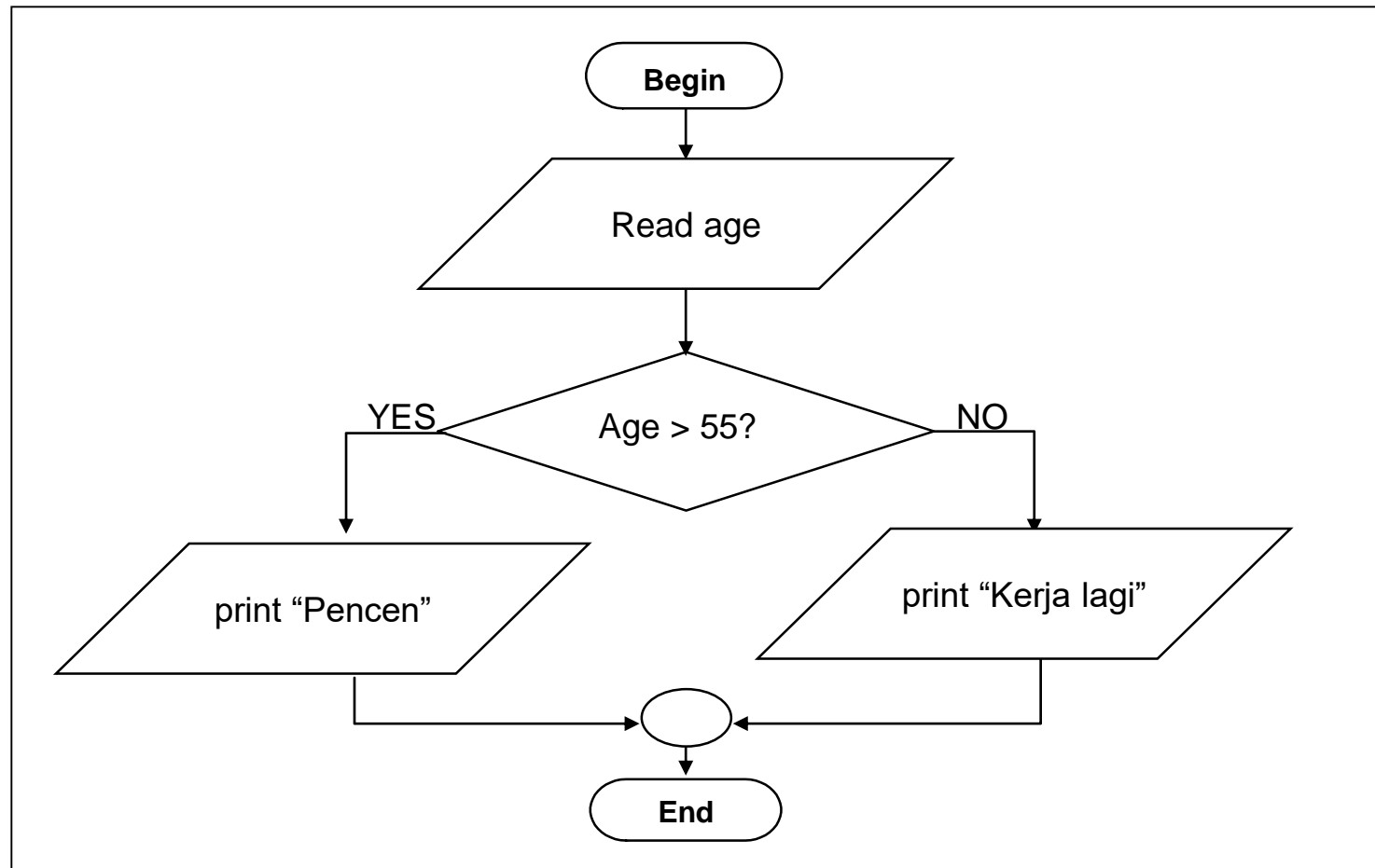
# Flowchart – repetition control structure



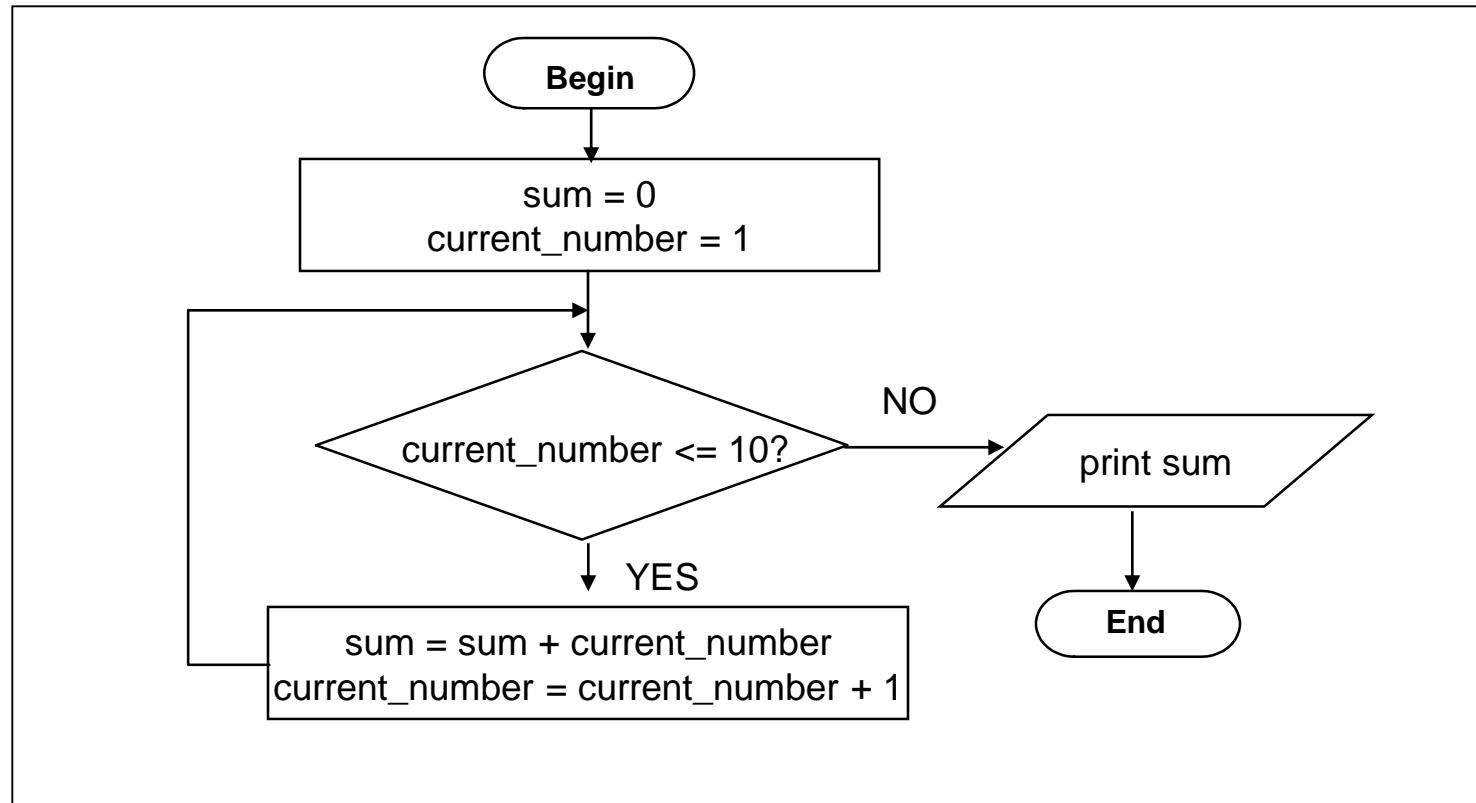
# Flowchart – example 1



# Flowchart – example 2



# Flowchart – example 5



# Example 4

- Write an algorithm and draw a flowchart that will calculate the roots of a quadratic equation  $ax^2 + bx + c = 0$
- Hint:  $d = \text{sqrt} ( b^2 - 4ac )$ , and the roots are:  
 $x1 = (-b + d)/2a$  and  $x2 = (-b - d)/2a$



# Exercises: Algorithm & Flowchart

- 1.) Create an algorithm and a flowchart that will accept/read two numbers and then display the bigger number.





# Exercises: Algorithm & Flowchart

2.) Create an algorithm and a flowchart that will compute the area of a circle.



# Exercises: Algorithm & Flowchart

3.) Create an algorithm and a flowchart that will compute the sum of two numbers. If the sum is below or equal to twenty, two numbers will be entered again. If the sum is above 20, it will display the sum.



# Lab Activity: Algorithm & Flowchart

- 4) Create an algorithm and a flowchart that will output the largest number among the three numbers.



# Assignment 1

1. Create an algorithm and a flowchart that will output for g.c.d.
2. Create an algorithm and a flowchart that will output the factorial of a given number.
3. Create an algorithm and a flowchart that will output the Fibonacci series up to a given number.
4. Create an algorithm and a flowchart that will output all the prime numbers between 2 numbers.