Computer Programming Fundamentals

A. H. M. Saiful Islam
Associate Professor
Dept. of CSE
NDUB

Contents

- Program
- Program Development Process
- Algorithm
- Control Structures
- Flowchart: Symbols, Preparing a Flowchart, Control Structures in Flowchart
- Pseudo code: Preparing Pseudo code, Control Structures for Pseudo code
- Programming Paradigm: Structured Programming
- Characteristics of a good program

Program

- Instructions in a Program have three parts
  - Accept Input Data that needs to be processed
  - Act upon Input Data and process it
  - Provide Output to user
- Instructions are defined in a specific sequence
- Program writing is not a straightforward task
  - Follows Program Development Life Cycle

Program Development Lifecycle

Program Analysis

- Understand the problem
- Have multiple solutions
- Select a solution

Program Design

- Write Algorithm
- Write Flowchart
- Write Pseudo code

Program Development

- Choose a programming language
- Write the program, by converting the pseudo code, using the programming language.
- Compile the program and remove syntax errors
- Execute the program.
- Test the program. Check the output results with different inputs. If the output is incorrect, modify the program to get correct results.
- Install the tested program on the user's computer.

Program Development and Maintenance

- Document the program, for later use.
- Maintain the program for updating, removing errors, changing requirements etc.

Algorithm

- Ordered sequence of finite, well defined, unambiguous instructions for completing a task.
- English-like representation of logic to solve problem
- Step-by-step procedure for solving problem
- For a particular task, different algorithms can be written
  - Select an algorithm based on advantages and disadvantages
  - Different Algorithms would typically lead to trade off between memory requirements and execution speed

Algorithm: An Example

Find greatest among three numbers

Algorithm 1.

Step 1. Start
Step 2. Read the three numbers A, B, C
Step 3. Compare A and B. If A is greater perform step 4 else perform step 5.
Step 4. Compare A and C. If A is greater, output “A is greatest” else output “C is greatest”. Perform step 6.
Step 5. Compare B and C. If B is greater, output “B is greatest” else output “C is greatest”.
Step 6. Stop

Has more number of comparisons

Algorithm 2.

Step 1. Start
Step 2. Read the three numbers A, B, C
Step 3. Compare A and B. If A is greater, store A in MAX, else store B in MAX.
Step 4. Compare MAX and C. If MAX is greater, output “MAX is greatest” else output “C is greatest”.
Step 5. Stop

An additional variable MAX is required

Control Structures

- Specifies statements to be executed and order of execution of statements
  - Execution of a statement based on a decision
  - Repetitively execute statements unless condition met
- Used by Flowchart, Pseudo code
- Three kinds
  - Sequential: Instructions executed in linear order
  - Selection (branch or conditional): Asks true/false question and selects next instruction based on answer
  - Iterative (loop): Repeats execution of block of instructions

Flowchart

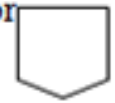
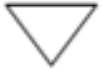
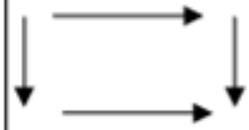
- Diagrammatic representation of logic for solving task
- Drawn using boxes of different shapes with lines connecting them to show the flow of control
- Make logic of program clearer in a visual form
- Diagrammatic representation forms a common medium of communication
- Drawn using different kinds of symbols.

The Flowchart

A Flowchart

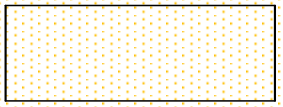
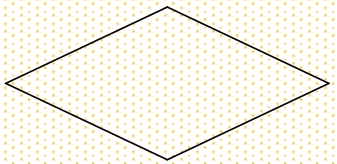
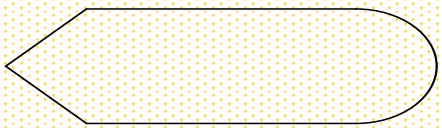
- shows logic of an algorithm
- emphasizes individual steps and their interconnections
- e.g. control flow from one action to the next

Flowchart symbols

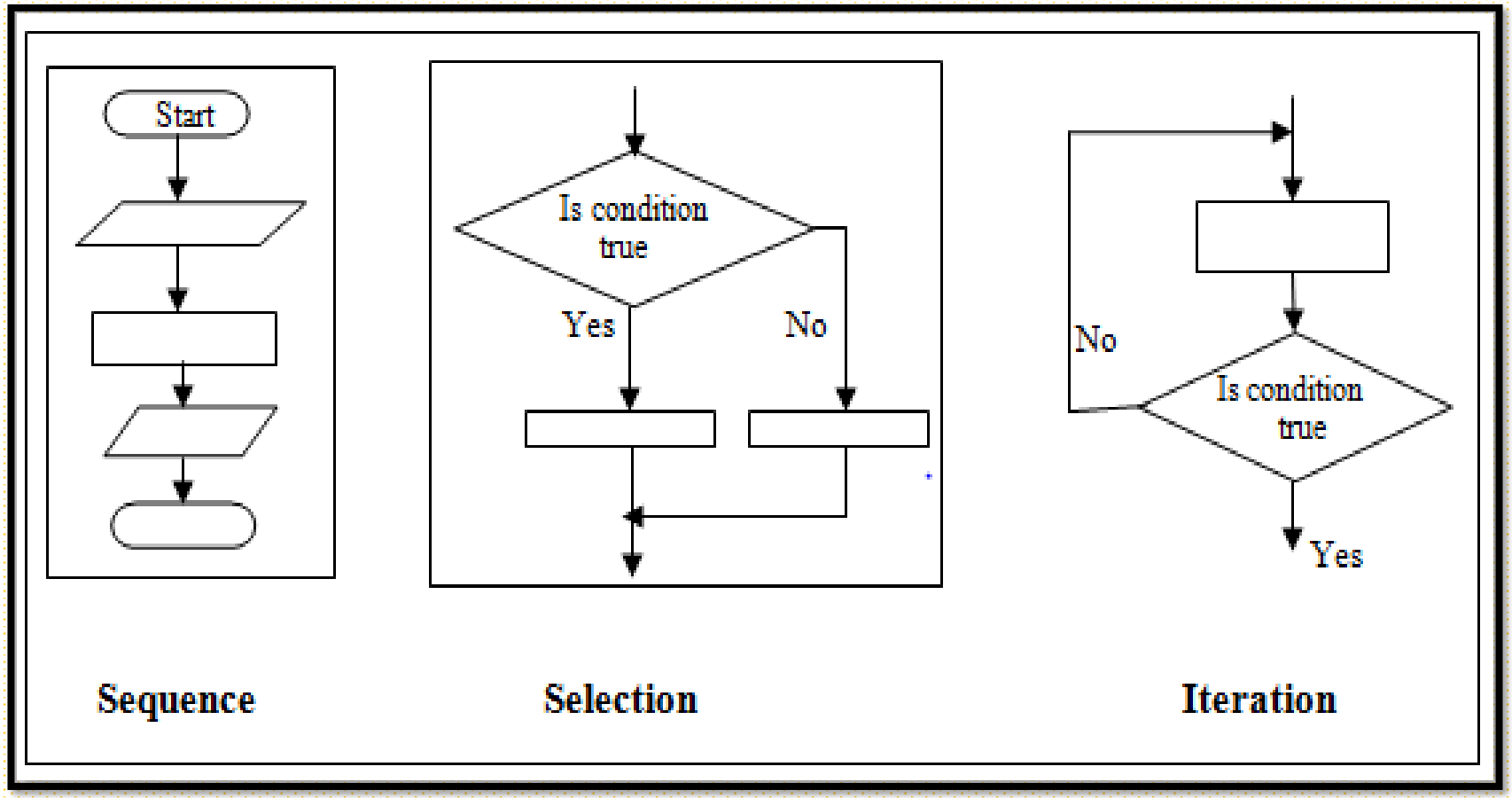
Process 	Alternate Process 	Decision 	Data 	Predefined process 
Internal Storage 	Document 	Multi document 	Terminator 	Preparation 
Manual Input 	Manual Operation 	Connector 	Off-page Connector 	Card 
Punched Tape 	Summing Junction 	OR 	Collate 	Sort 
Extract 	Merge 	Stored Data 	Delay 	Sequential Access Storage 
Magnetic Disk 	Direct Access Storage 	Display 	Flow Lines 	

Flowchart Symbols

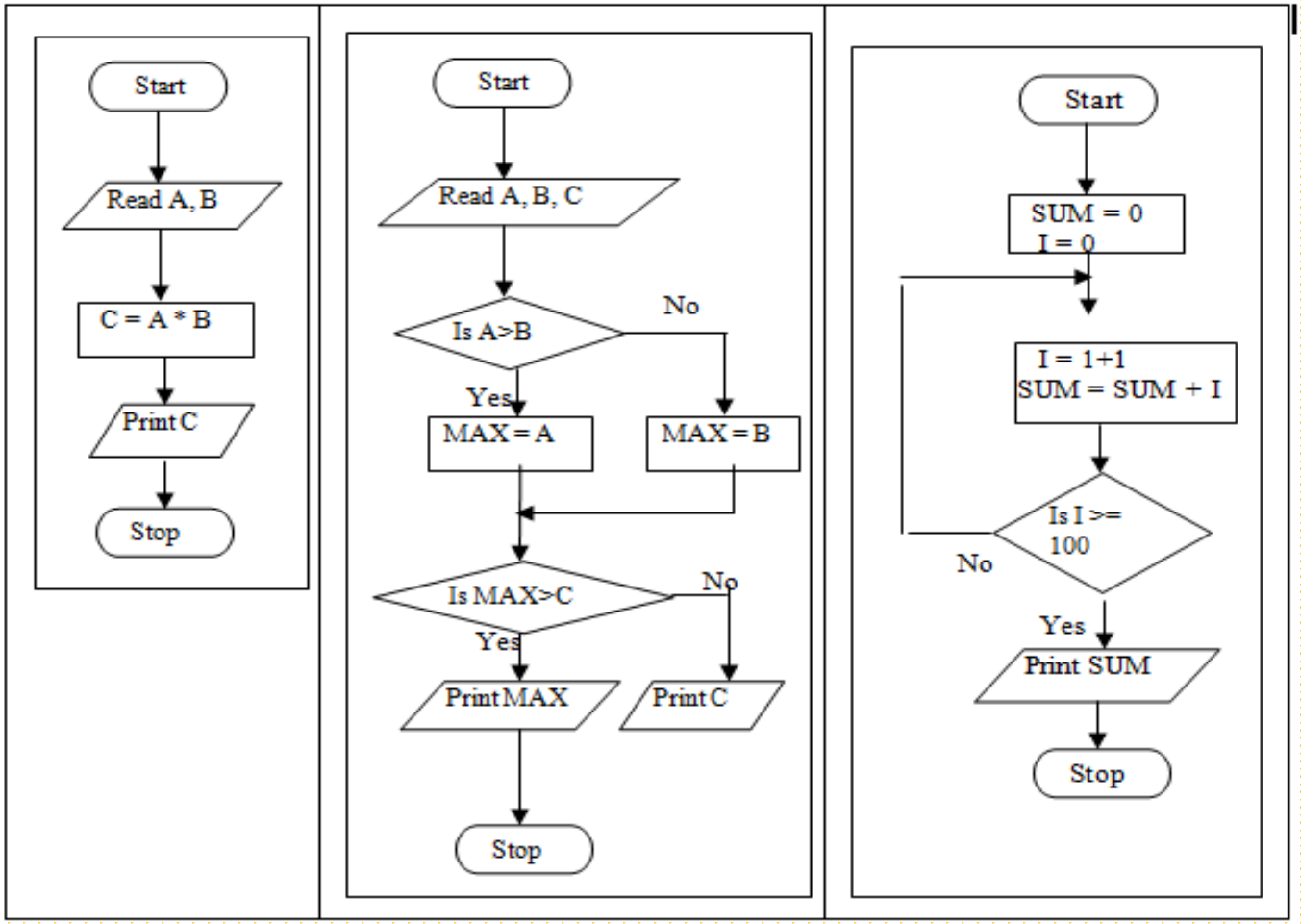
Basic

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

Control Structures in Flowchart



Examples of Flowchart



Product of two numbers

Maximum of three numbers

Sum of first 100 integers

Pseudo code

- Consists of short, readable and formally-styled English language used for explaining an algorithm.
- Does not include details like variable declarations, subroutines etc.
- Short-hand way of describing computer program
- Not based on any programming language
- Uses structured constructs of programming language but is not machine readable
- Cannot be compiled or executed
- No standard for syntax of pseudo code exists
- Easily translated into a programming language

Preparing Pseudo code

- Written using structured English
- Commonly used terms to represent actions
  - Inputting data: INPUT, GET, READ
  - Outputting data: OUTPUT, PRINT, DISPLAY
  - Calculations: COMPUTE, CALCULATE
  - Incrementing: INCREMENT
- Control structures
  - *Sequence structure:* sequence of steps executed in linear order
  - *Selection constructs:* if-statement, case statement
  - *Iterative statements:* WHILE, DO-WHILE

Control structures for Pseudo code

```
Step 1

Step 2

Step 3

:
:
:
```

Sequence

```
WHILE (condition)
Statement 1
 Statement 2
     :
     :
END
```

```
DO
    Statement 1
    Statement 2
     :
     :
  WHILE (condition)
```

Iteration

```
IF (condition) THEN
    Statement(s) 1
ELSE
     Statement(s) 2
ENDIF
```

```
IF (condition) THEN
    Statement(s) 1
ENDIF
```

```
CASE expression of
   Condition-1 : statement1
   Condition-2 : statement2
    :
   Condition-N : statement N
 OTHERS: default statement(s)
```

Selection

Algorithm, Flowchart, Pseudo code

- Algorithm: A sequence of instructions used to solve a particular problem
- Flowchart and Pseudo code: Tools to document and represent algorithm
  - Flowchart : Graphical representation of algorithm
  - Pseudo code: Readable, formally styled English like language representation of algorithm
- No knowledge of programming language required to write or understand flowchart or pseudo code

Programming task phases

- 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

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. Less than 50 will be considered as failing state.

Pseudocode & Algorithm

Algorithm:

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

  Print “FAIL”

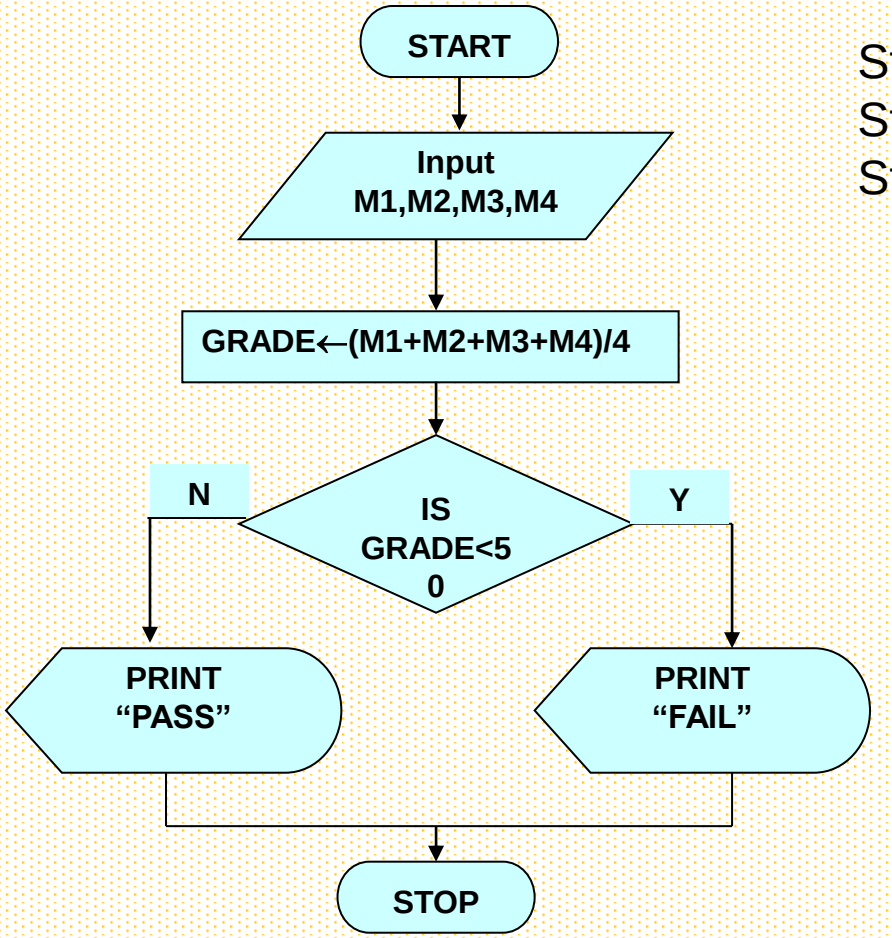
  else

  Print “PASS”

Pseudocode & Algorithm

- Detailed Algorithm/ Pseudocode
- 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



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.

Algorithm:

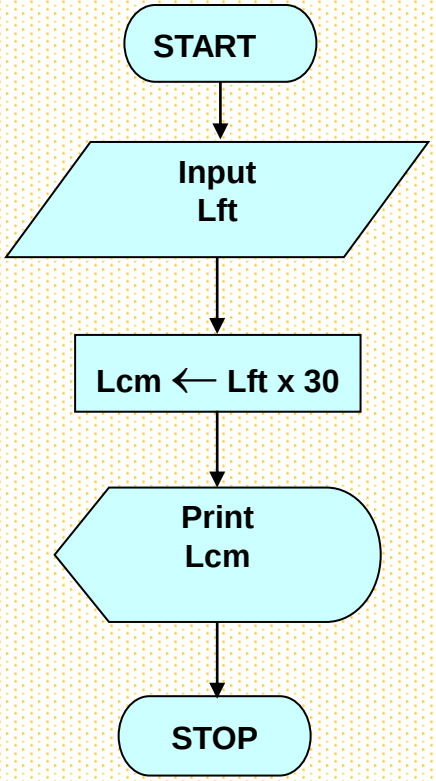
- *Input the length in feet (Lft)*
- *Calculate the length in cm (Lcm) by multiplying LFT with 30*
- *Print length in cm (LCM)*

Example 2

Pseudocode

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

Flowchart



Example 3

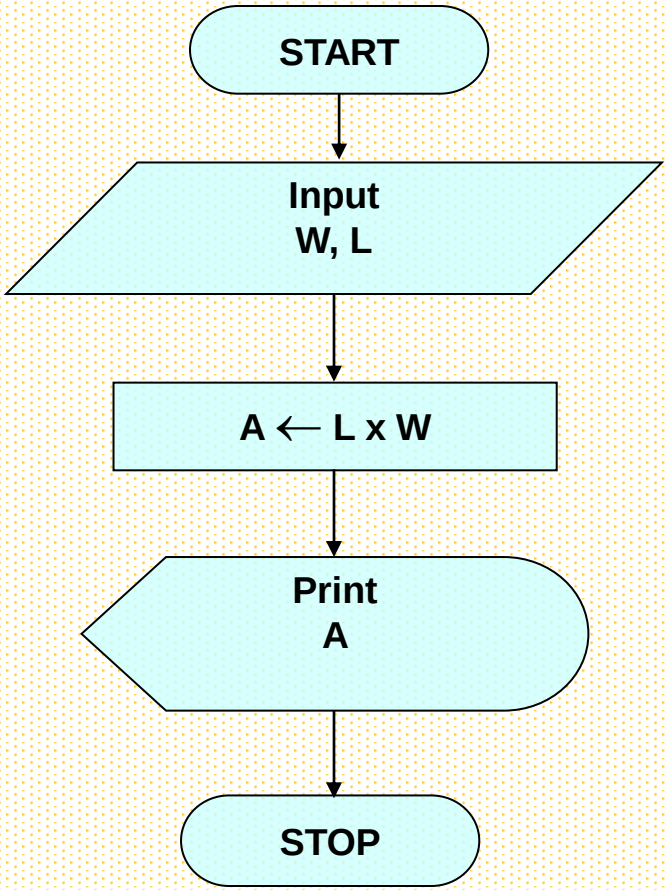
Write an algorithm and draw a flowchart that will read the two sides of a rectangle and calculate its area.

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

Example 3

Pseudocode

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



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** = sqrt ($b^2 - 4ac$) and the roots are: ***x*1** = (–*b* + *d*)/2*a* and ***x*2** = (–*b* – *d*)/2*a*

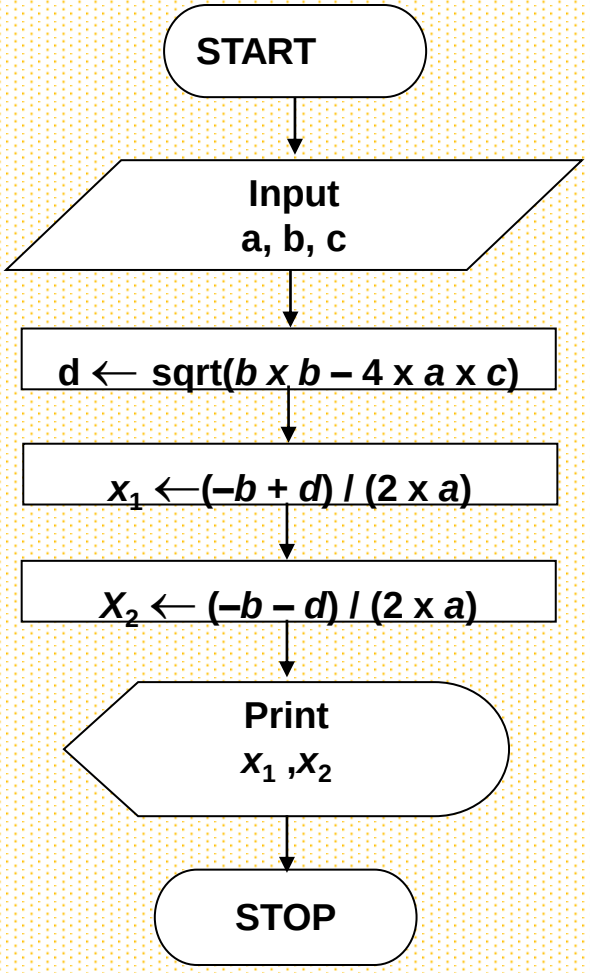
Example 4

Algorithm:

- *Input the coefficients (a, b, c) of the quadratic equation*
- *Calculate* **d**
- *Calculate* **x1**
- *Calculate* **x2**
- *Print* x1 *and* x2

Example 4

- **Pseudocode**
- Step 1: Input a, b, c
- Step 2: $d \leftarrow \text{sqrt}(\quad b \times b - 4 \times a) \times c$
- Step 3: $x1 \leftarrow (-b + d) / (2 \times a)$
- Step 4: $x2 \leftarrow (-b - d) / (2 \times a)$
- Step 5: Print $x1$, $x2$



Structured Programming

- Build programs using small modules
  - Modules: Easy to read & write
- Break problem into small tasks that can be written independently. Small tasks combined together to form complete task.
- Software divided into procedures or modules, based on overall functionality of software.
- *E.g. C, Pascal*

Characteristics of Good Program

- Easily readable and structured
- Should not have hard-coded input values
- Well-documented
- Portable: Minimum dependence on a particular OS

Thank You