Lecture 7

PROBLEM SOLVING

Outline

This lecture covers:

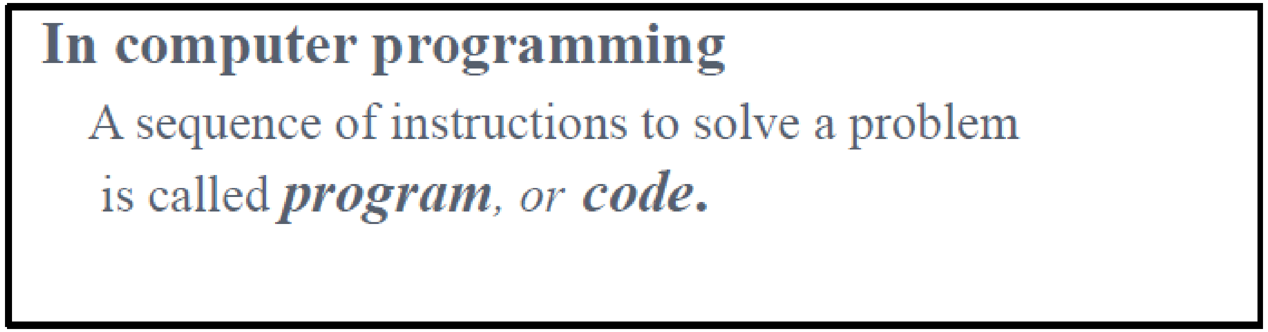
- Part A
  - Problem solving concepts & steps
  - Algorithms, Pseudocode, & Flow Charts
  - Flowcharts’ Symbols
- Part B
  - Control Structures
  - Selection control structure
  - Equality & Relational Operators
  - If … Else & Multiple Cases

PROBLEM SOLVING STEPS

- Define the problem.
- Analyze the problem.
- Develop an algorithm *(a method)*for solving a problem.
- Write a computer program corresponding to the algorithm.
- Test and debug the program.
- Document the program *(write an explanation of how the program works and how to use it)*.

ALGORITHMS

- A set of steps for carrying out a specific task.



In computer programming

A sequence of instructions to solve a problem is called ***program***, *or* ***code*.**

The **four** essential properties of an algorithm:

1.Each step of an algorithm must be **exact**.

2.An algorithm must **terminate**.

3.An algorithm must be **effective**.

4.An algorithm must be **general**.

ALGORITHMS

The **four** essential properties of an algorithm:

❑***Exact:*** Precisely and unambiguously described, so that there remains no uncertainty.

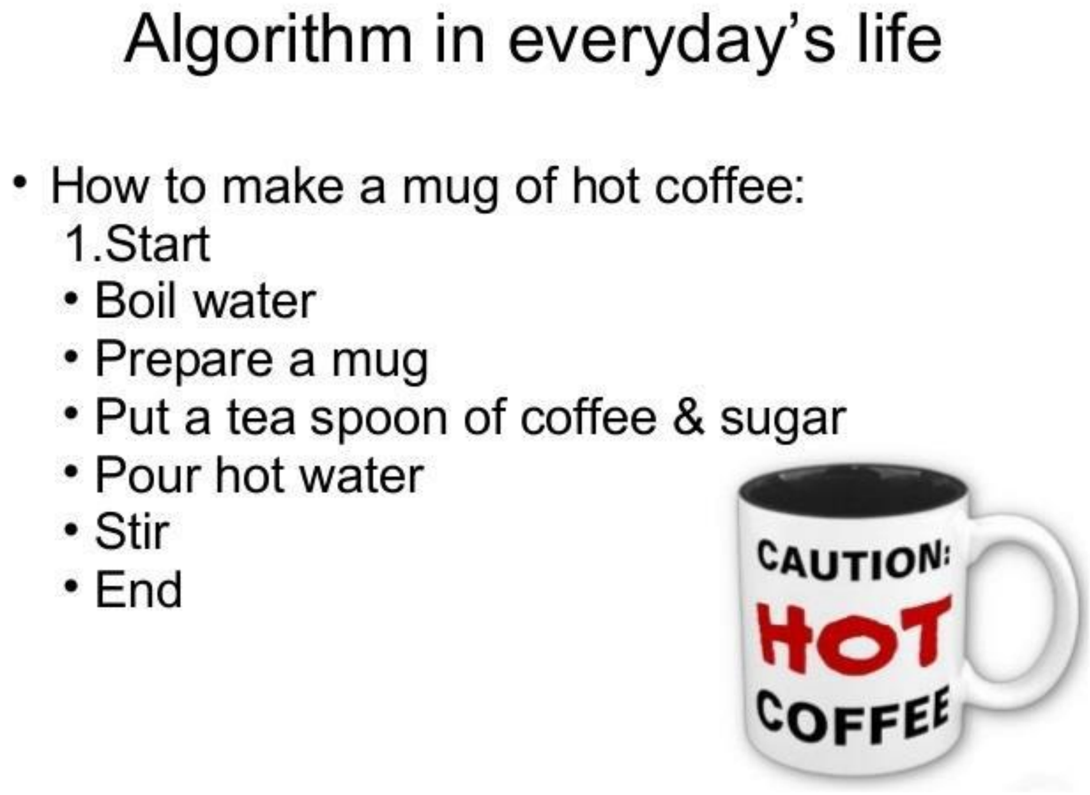
❑***Terminate:*** The ultimate purpose of an algorithm is to solve a problem.

❑***Effective:*** must give the correct answer.

❑***General:*** Must solve every instance of the problem.

E.g, a program that computes the area of a rectangle, within the limits of the programming language and machine.

ALGORITHMS



Algorithm in everyday's life

- How to make a mug of hot coffee:
  1.Start
  - Boil water
  - Prepare a mug
  - Put a tea spoon of coffee & sugar
  - Pour hot water
  - Stir
  - End

CAUTION:
HOT
COFFEE

PROGRAMMING TOOLS

HOW TO SOLVE A PROBLEM?

❑ Two commonly used tools:

1. **Pseudo-code**
2. **Flowcharts**

WHAT IS A PSEUDO-CODE

- ❑Pseudocode is an artificial and informal language that helps programmers developalgorithms.
- ❑It is the very first version(s) of an algorithm, usually very general and unrefined. It's then refined successively to get a step-by-step detailed algorithm.

- ❑Pseudocode is very similar to everyday English.
- ❑… a mixture of:
- ✓English statements.
- ✓Some mathematical notations.
- ✓Selected keywords from programming language.
- ❑There is no standard convention for writing pseudo

Pseudocode:

Example: 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:

```
1.Input a set of 4 marks
2.Calculate their average by summing and dividing by 4
3.If average is below 50
  1.Print"FAIL"
  else
  2.Print "PASS"
```

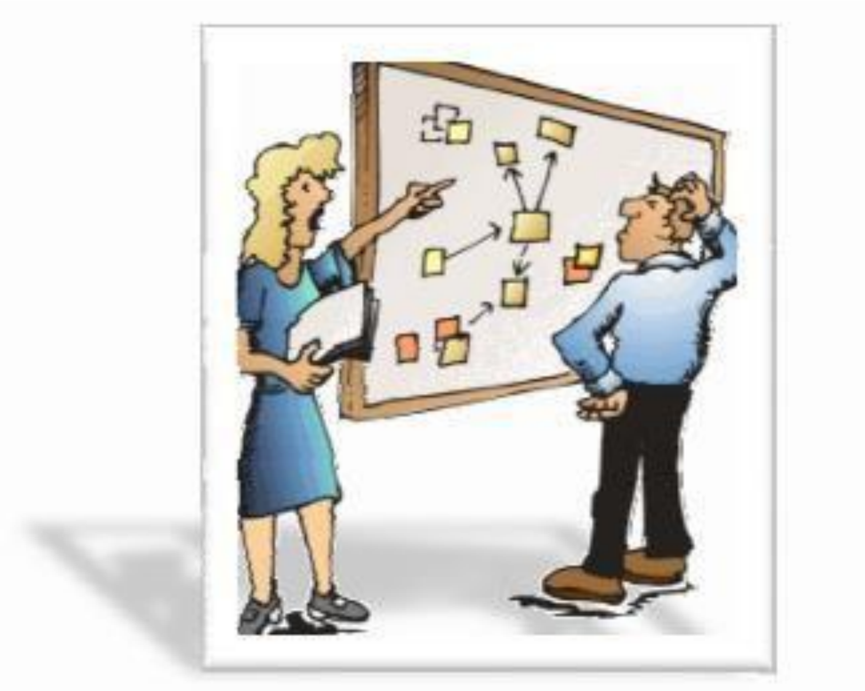
Pseudocode:

Detailed Algorithm

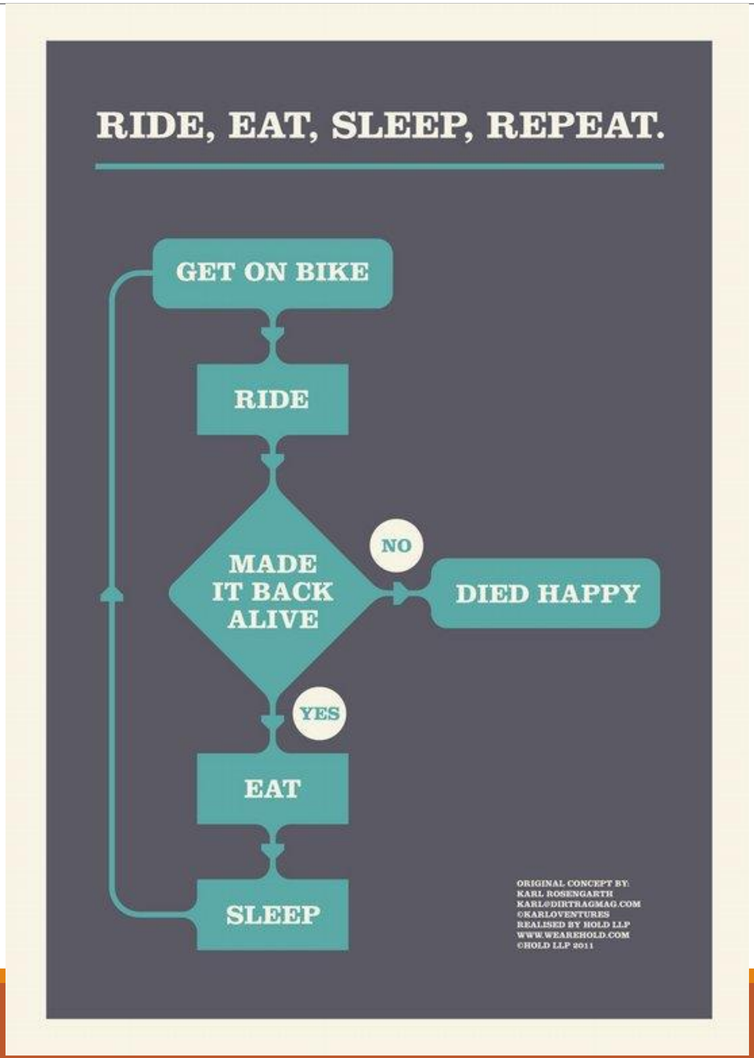
```
Step 1:Input M1,M2,M3,M4
Step 2:    GRADE       (M1+M2+M3+M4)/4
Step 3:    If (GRADE < 50) then
Print "FAIL"
else
Print "PASS"
endif
```

WHAT IS A FLOWCHART?

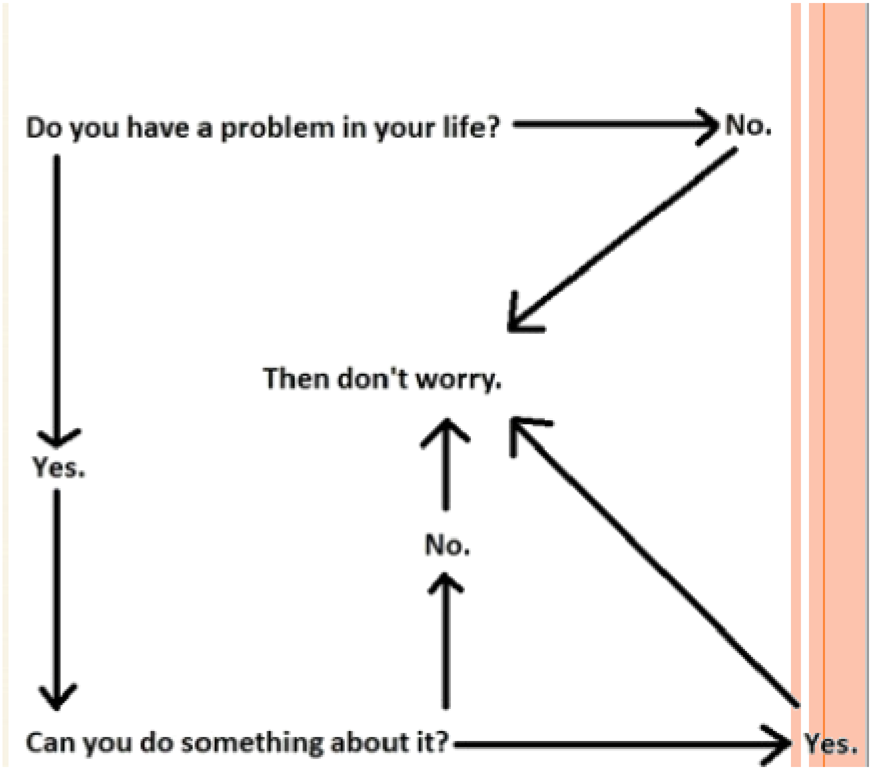
A diagram that uses **standard symbols** to solve the problem.



WHAT IS A FLOWCHART?



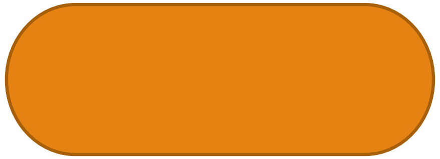
ORIGINAL CONCEPT BY:
KARL ROSENGARTH
KARL@DIRTRAGMAG.COM
©KARLOVENTURES
REALISED BY HOLD LLP
WWW.WEAREHOLD.COM
©HOLD LLP 2011


Do you have a problem in your life?
No.
Then don't worry.
Yes.
No.
Can you do something about it?
Yes.

FLOWCHART Symbols

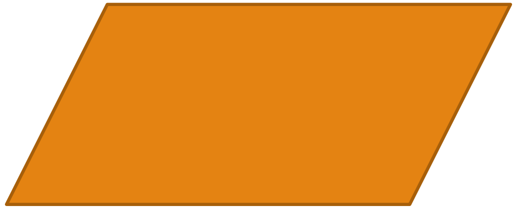
- **Terminal symbol**

It is used to represent the start, end of the program logic.



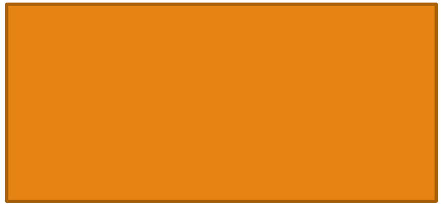
- **Input /Output**

It is used for input or output.



- **Process Symbol**

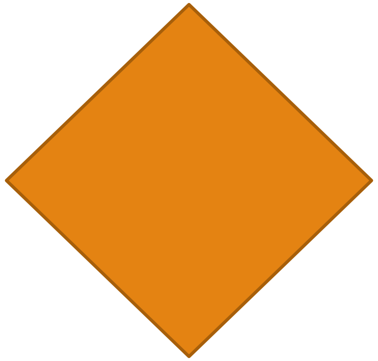
It is used to represent the calculations, data movements,



FLOWCHART SYMBOLS

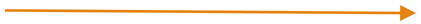
- **Decision Symbol**

It is used to denote a decision to be made at that point



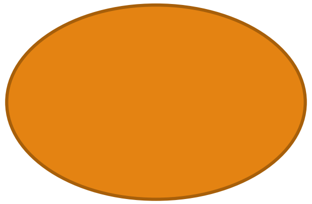
- **Flow lines**

It is used to connect the symbols



- **Connectors**

It is used to connect the flow lines.



Example1: FLOWCHART

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

Pseudocode:

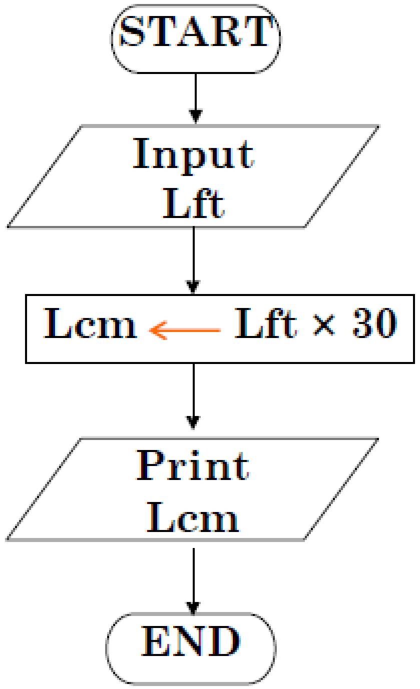
```
1.Input the length in feet (Lft).
2.Calculate the length in cm (Lcm) by multiplying Lftwith 30.
3.Print length in cm (LCM).
```

Example1 *(continued)* : FLOWCHART

Algorithm

```
Step 1:    Input Lft
Step 2:    Lcm      Lft×30
Step 3:    Print Lcm
```

Flowchart



Example 2:FLOWCHART

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

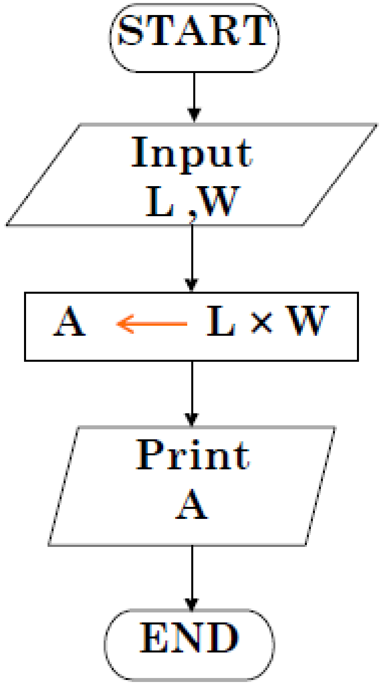
Pseudocode:

```
1.Input the width(W)and Length(L)of a rectangle.
2.Calculatethearea(A)by multiplying L with W.
3.Print (A).
```

Algorithm

```
Step 1:      Input W,L
Step 2:        A  =  L ×W
Step 3:      Print A
```

Flowchart



Example 3

Write an algorithm and draw a flowchart that will

$$ax^2 + bx + c = 0$$

- ❑ Hint: **d** = sqrt (b^2 - 4ac), and the roots are:
  x1 = (*-b + d*) / *2a* and **x2** = (*-b - d*) / *2a*

$$\boxed{x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}}$$

Example 3: (CONT.)

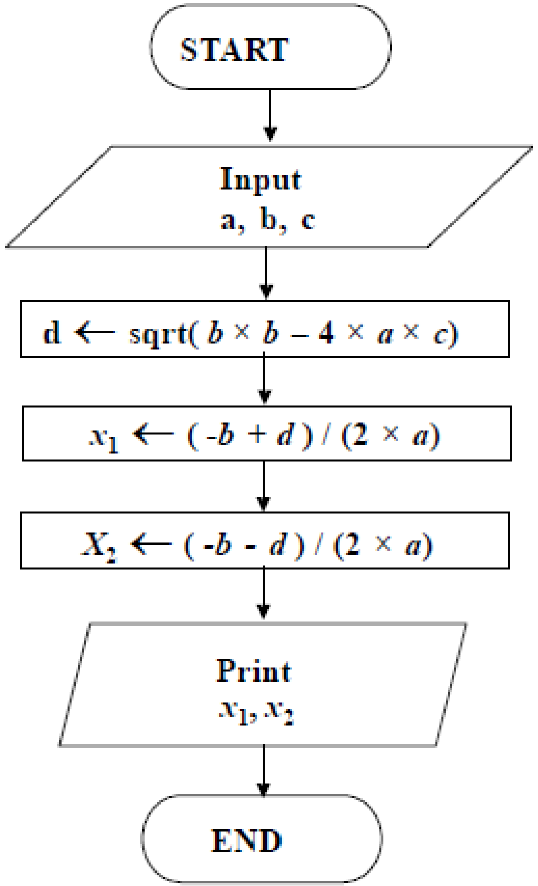
Pseudocode:

```
1.Input the coefficients (a, b, c) of the quadratic equation.
2.Calculate d.
3.Calculate x1.
4.Calculate x2.
5.Print x1 and x2.
```

Example 3: *(continued)*

Algorithm:

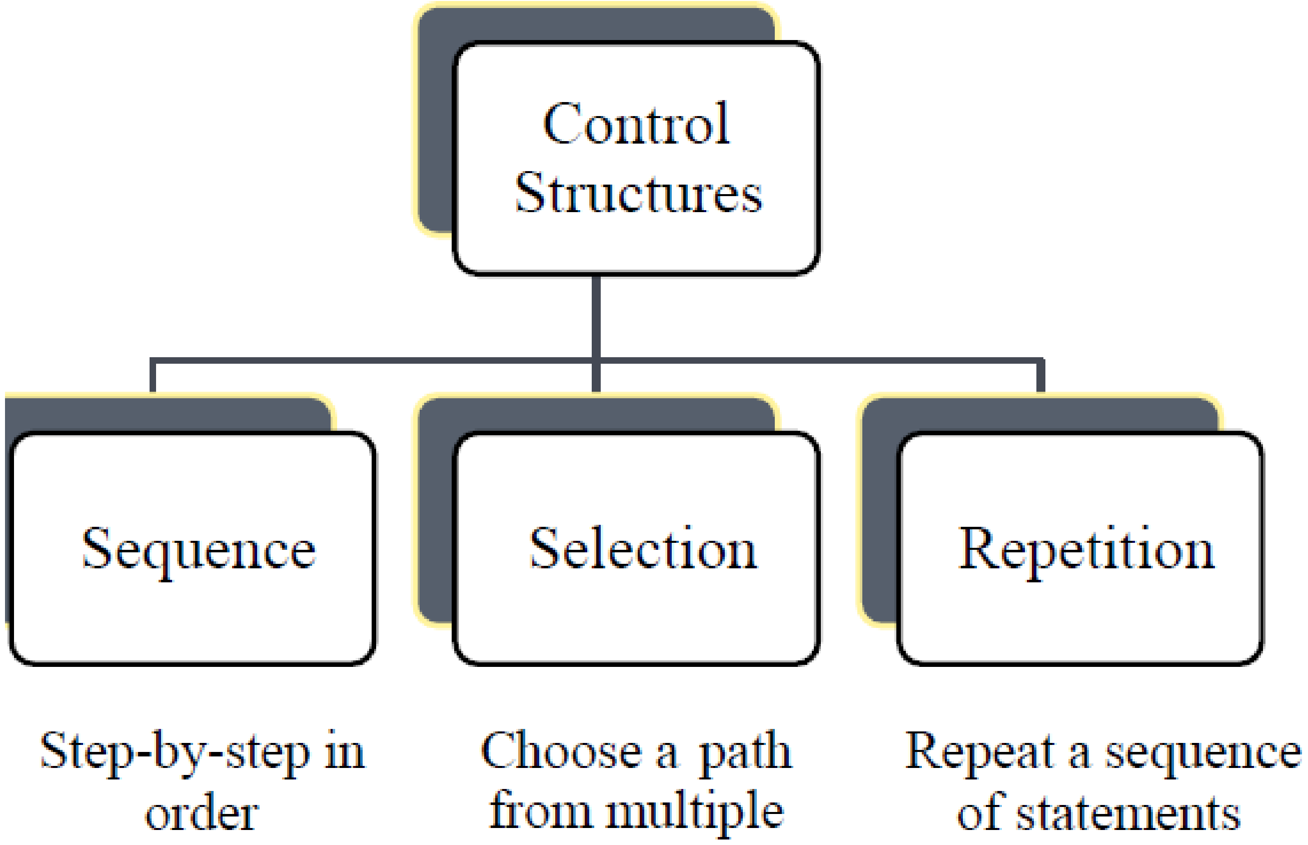
```
Step 1:   Input a, b, c
Step 2:   d ← sqrt ( b × b - 4 × a × c)
Step 3:   x1 ← (-b + d) / (2 × a)
Step 4:   x2 ← (-b - d) / (2 × a)
Step 5:   Print x1, x2
```


START
Input
a, b, c
d ← sqrt($b \times b - 4 \times a \times c$)
$x_1 \leftarrow (-b + d) / (2 \times a)$
$X_2 \leftarrow (-b - d) / (2 \times a)$
Print
x_1, x_2
END

PART B

SELECTION(DECISION) CONTROL STRUCTURE

Control Structures



Control Structures

Sequence

Selection

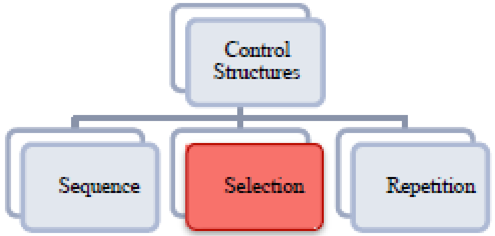
Repetition

Step-by-step in order

Choose a path from multiple

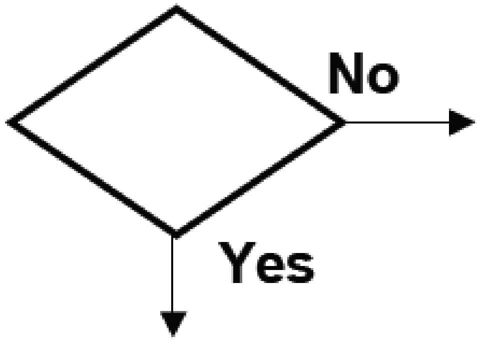
Repeat a sequence of statements

SELECTION(DECISION)



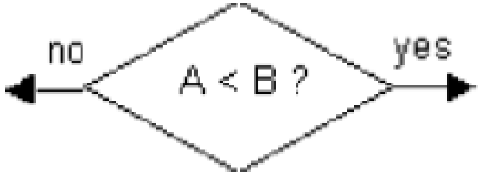
❑A step in an algorithm that leads to more than one possible

Decision



For example:

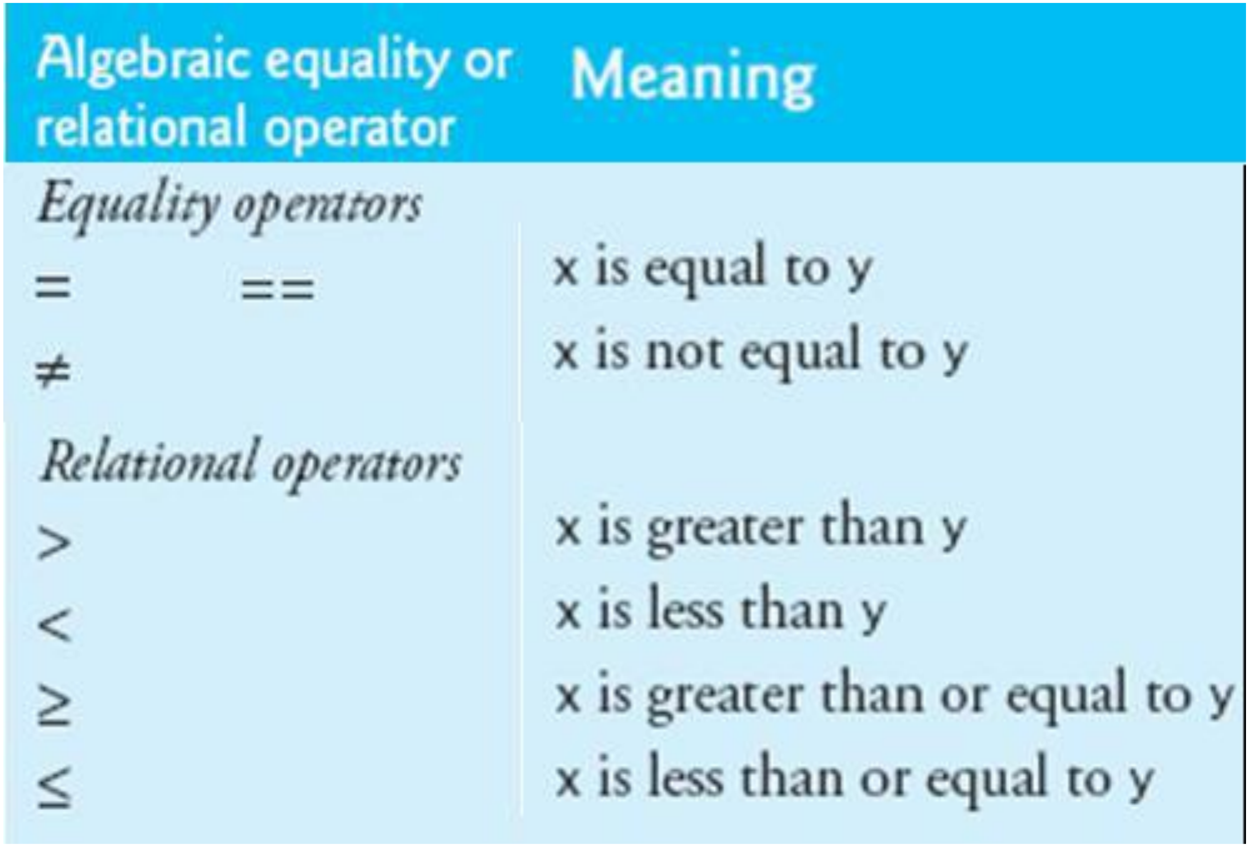
if A = 10 and B = 20



The answer is? ***Yes***

EQUALITY& RELATIONAL OPERATORS

- The idea of true/false (Yes/No) answer.



Algebraic equality or relational operator	Meaning
Equality operators	
= ＝＝	x is equal to y
≠	x is not equal to y
Relational operators	
>	x is greater than y
<	x is less than y
≥	x is greater than or equal to y
≤	x is less than or equal to y

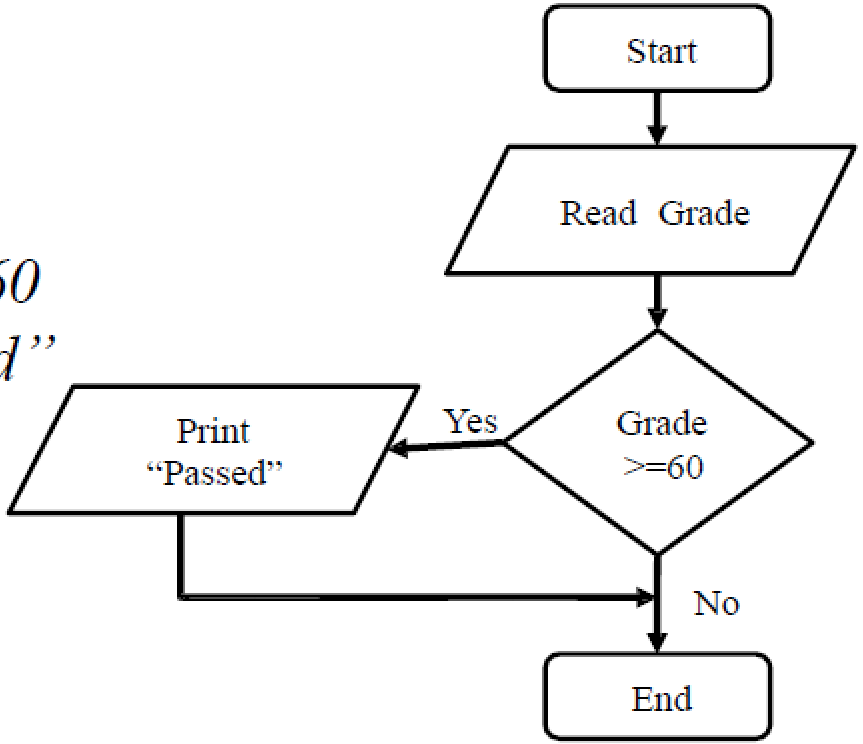
Example 4: SELECTION(DECISION)

(if STATEMENT)

Reads a student grade and write “passed” if it is greater than or equal 60.

Pseudocode

```
1. Input Grade
2. If Grade>= 60
3. Print “Passed”
```

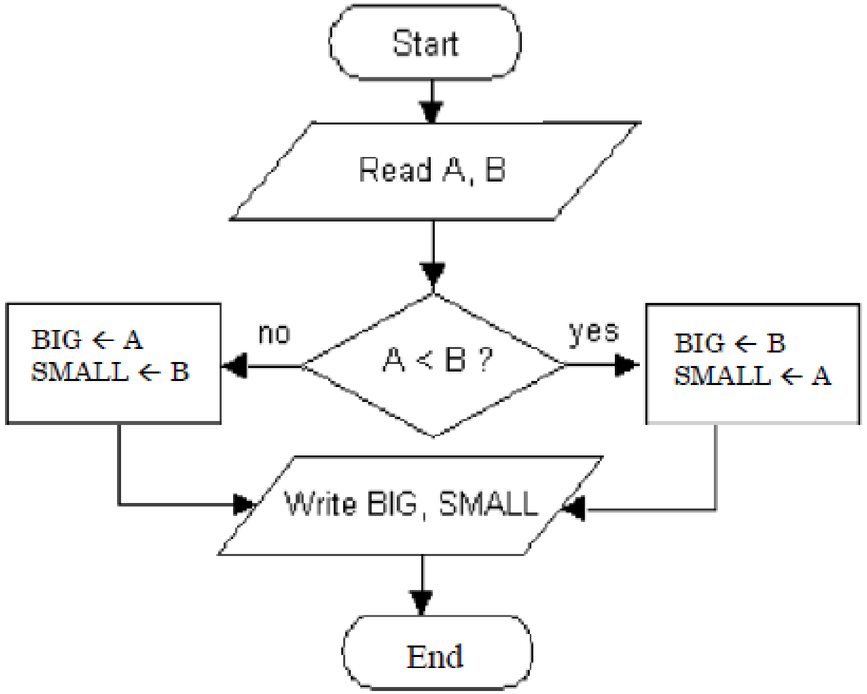


Example 5: SELECTION(DECISION)

Example 5:(if … ELSE) Reads two numbers and displays the numbers read in descending (decreasing) order

Algorithm:

- Input A,B
- *if A < B*
- *BIG* ← B
- *SMALL* ← A
- ELSE
- *BIG* ← A
- *SMALL* ← B
- Print *BIG*, SMALL



Start

Read A, B

A < B ?

no

BIG ← A
SMALL ← B

yes

BIG ← B
SMALL ← A

Write BIG, SMALL

End

Example 6: SELECTION(DECISION)

Draw a flowchart 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.

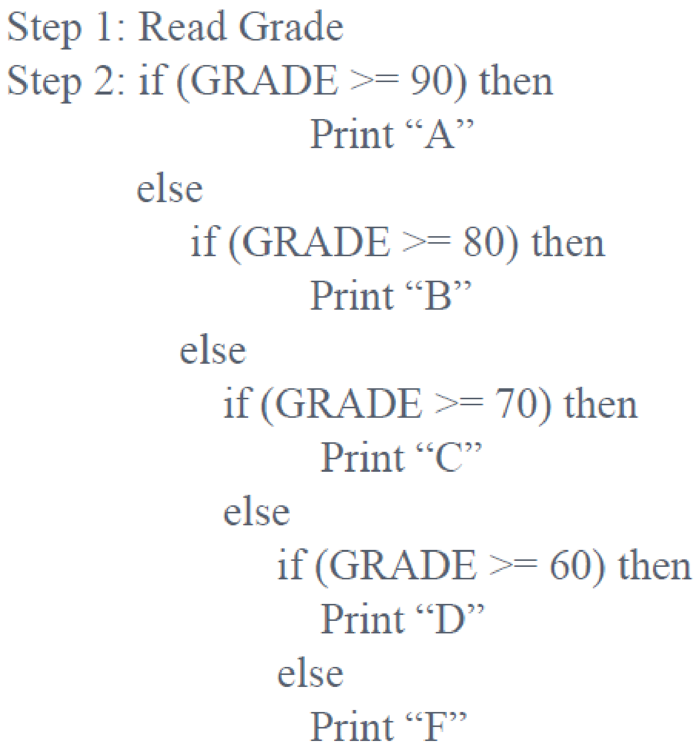
SELECTION(DECISION)

Example 7: (Multiple Cases)

- Reads a student grade and write the following:
  - ✓A for exam grades greater than or equal to 90,
  - ✓B for grades greater than or equal to 80,
  - ✓C for grades greater than or equal to 70,
  - ✓D for grades greater than or equal to 60, and
  - ✓F for all other grades.

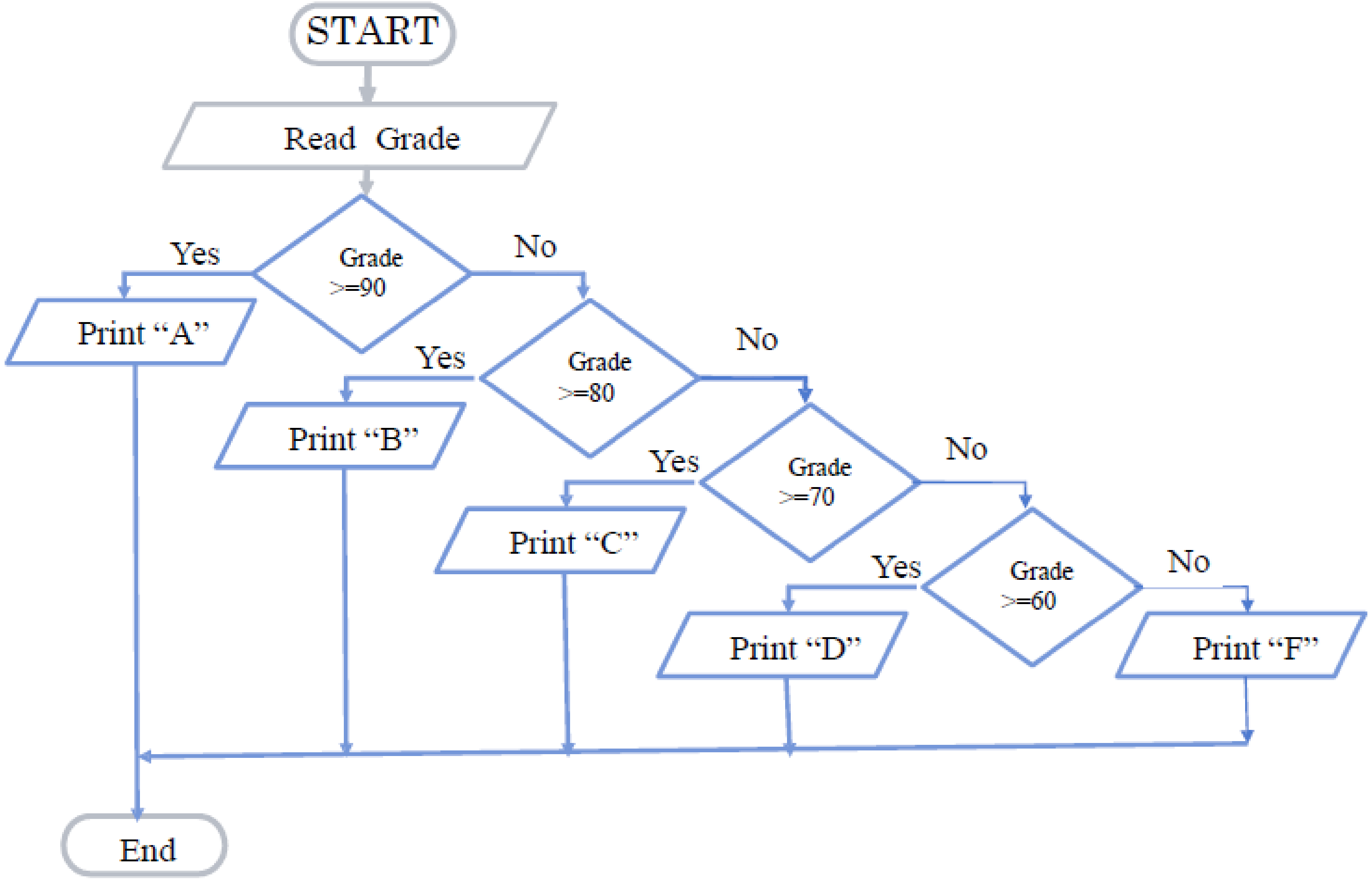
Example 7:(continued)

Algorithm



```
Step 1: Read Grade
Step 2: if (GRADE >= 90) then
              Print "A"
      else
        if (GRADE >= 80) then
              Print "B"
        else
          if (GRADE >= 70) then
              Print "C"
          else
            if (GRADE >= 60) then
              Print "D"
            else
              Print "F"
```

Example 7:(continued)


START
Read Grade
Grade >=90
Yes
No
Print "A"
Grade >=80
Yes
No
Print "B"
Grade >=70
Yes
No
Print "C"
Grade >=60
Yes
No
Print "D"
Print "F"
End