SINAI UNIVERSITY

CSW 232
Computer Programming (1)

SPRING 2024

Lecture 01 - Introduction

Instructor: Dr. Tarek Abdul Hamid

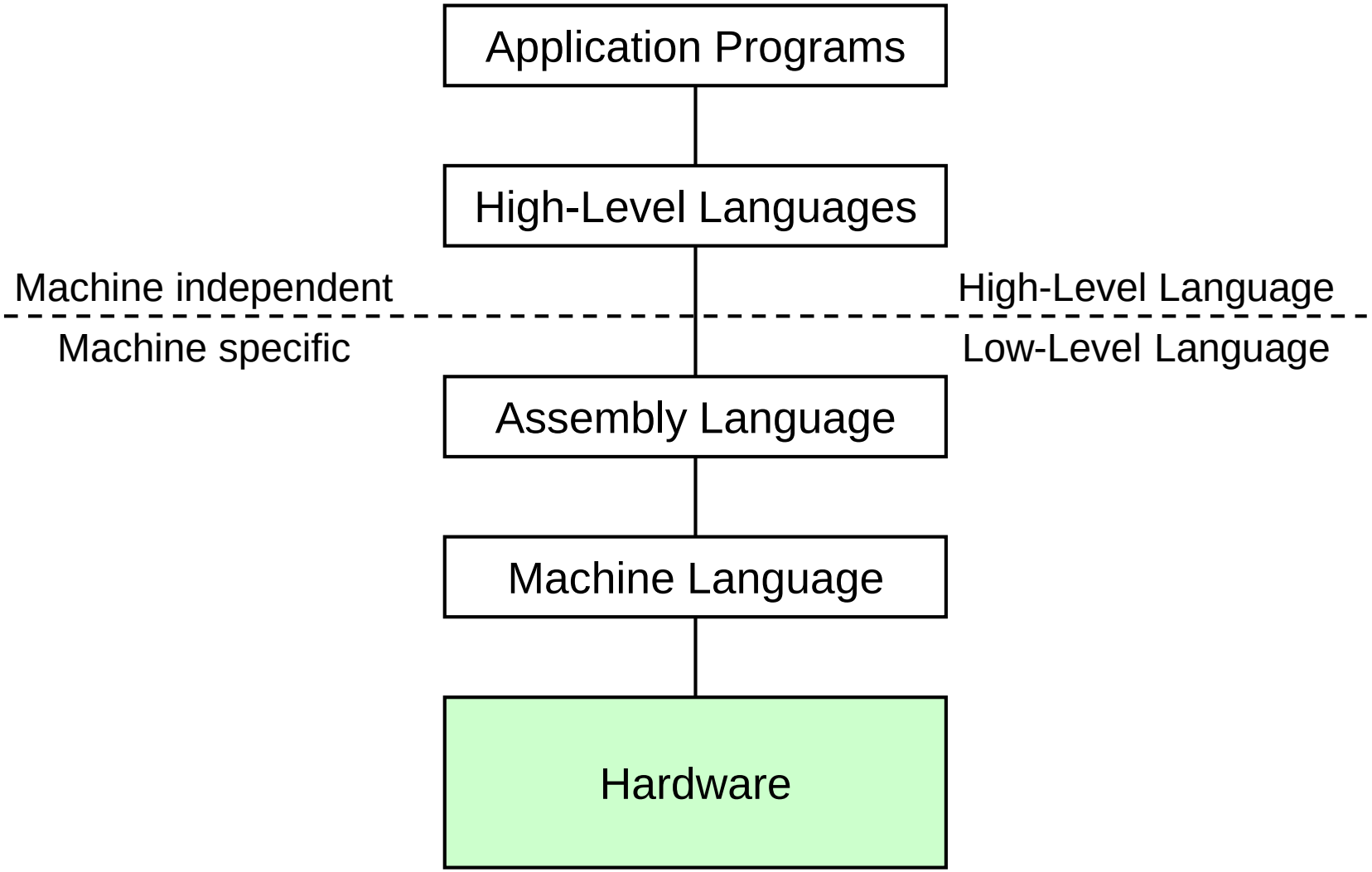
Computer Language

- Digital devices have two stable states, which are referred to as Zero and One by convention

- The binary number system has two digits, 0 and 1. A single digit (0 or 1) is called a *bit*, short for *binary digit*. A byte is made up of 8 bits.

- Binary Language: Data and instructions (numbers, characters, strings, etc.) are encoded as binary numbers - a series of bits (one or more bytes made up of zeros and ones)

A Hierarchy of Languages



Programming

- Programming – the creation of an ordered set of instructions to solve a problem with a computer.
- Only about 100 instructions that the computer understands - Different programs will just use these instructions in different orders and combinations.
- The most valuable part of learning to program is learning how to think about arranging the sequence of instructions to solve the problem or carry out the task

Programming Fundamentals:

- Sequential Processing
  - A List of Instructions
- Conditional Execution
  - Ifs
- Repetition
  - Looping / Repeating
- Stepwise Refinement / Top-Down Design
  - Breaking Things into Smaller Pieces
- Calling Methods / Functions / Procedures / Subroutines
  - Calling a segment of code located elsewhere
  - Reuse of previously coded code segment

Problem Solving

- The process of defining a problem, searching for relevant information and resources about the problem, and of discovering, designing, and evaluating the solutions for further opportunities. Includes:
  - Finding an Answer to a Question
  - Figuring out how to Perform a Task
  - Figure out how to Make Things Work

- Not enough to know a particular programming language… Must be able to problem solve…

- Very desirable to be a good Problem Solver in any CIS discipline.

Polya's 4 Steps of Problem Solving

- U – Understand the Problem
- D – Devise a Good Plan to Solve
- I – Implement the Plan
- E – Evaluate the Solution

Polya's 4 Steps of Problem Solving

- U - Read the Problem Statement
  - Identify the inputs, outputs, and processes
- D - Decide how to Solve the Problem
  - Create an Algorithm / Flowchart / Pseudocode
- I - Program the Code
  - Implement in Programming Language
- E - Test the Solution
  - Run the Code using numerous, varied test cases

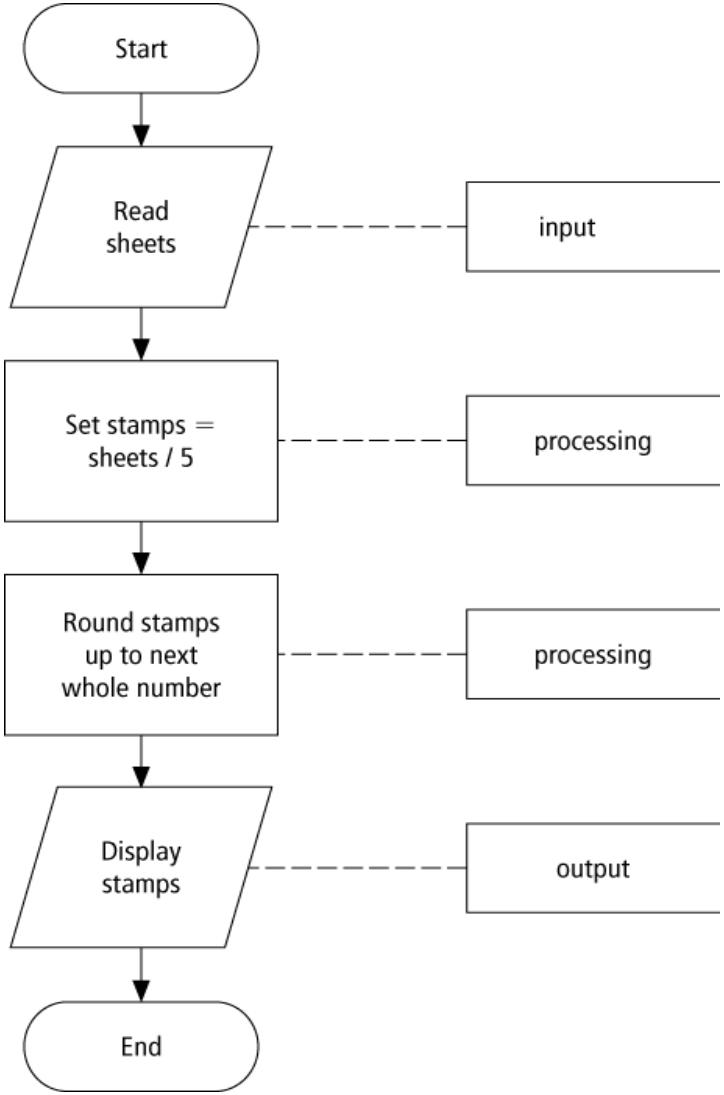
Flowcharts

- Graphically depict the logical steps to carry out a task and show how the steps relate to each other.

Flowchart symbols

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

Flowchart example



Pseudocode

- Uses English-like phrases to outline the task.

Pseudocode example

Determine the proper number of stamps for a letter

Read Sheets *(input)*

Set the number of stamps to Sheets / 5 *(processing)*

Round the number of stamps up to the next whole number *(processing)*

Display the number of stamps *(output)*

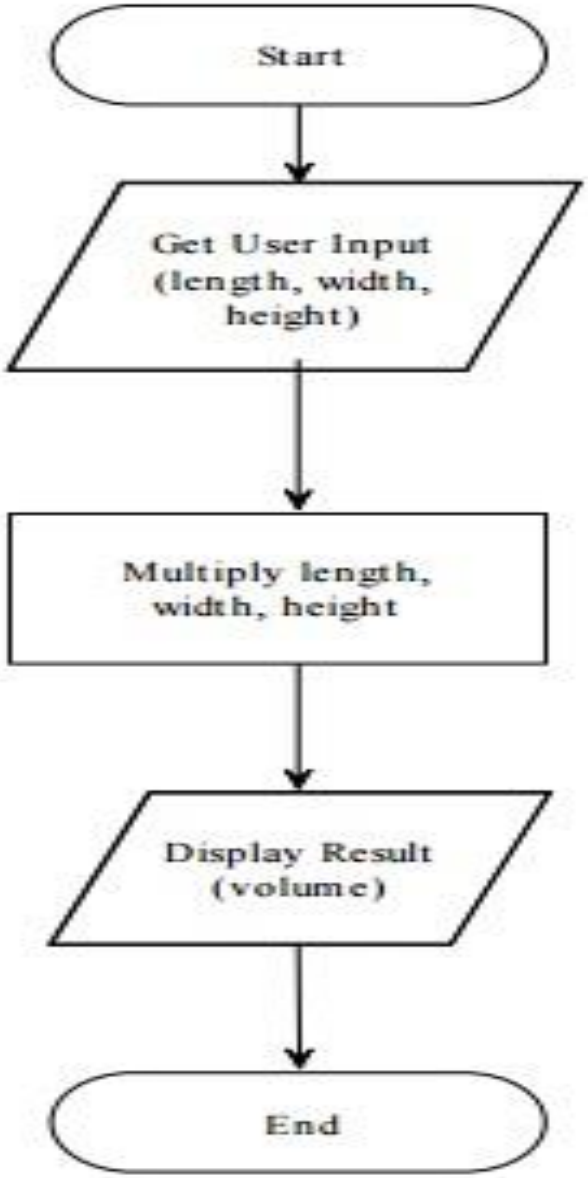
Statement structures

- Sequence – follow instructions from one line to the next without skipping over any lines

- Decision - if the answer to a question is “Yes” then one group of instructions is executed. If the answer is “No,” then another is executed

- Looping – a series of instructions are executed over and over

Sequence



Get length, width, height

Compute volume

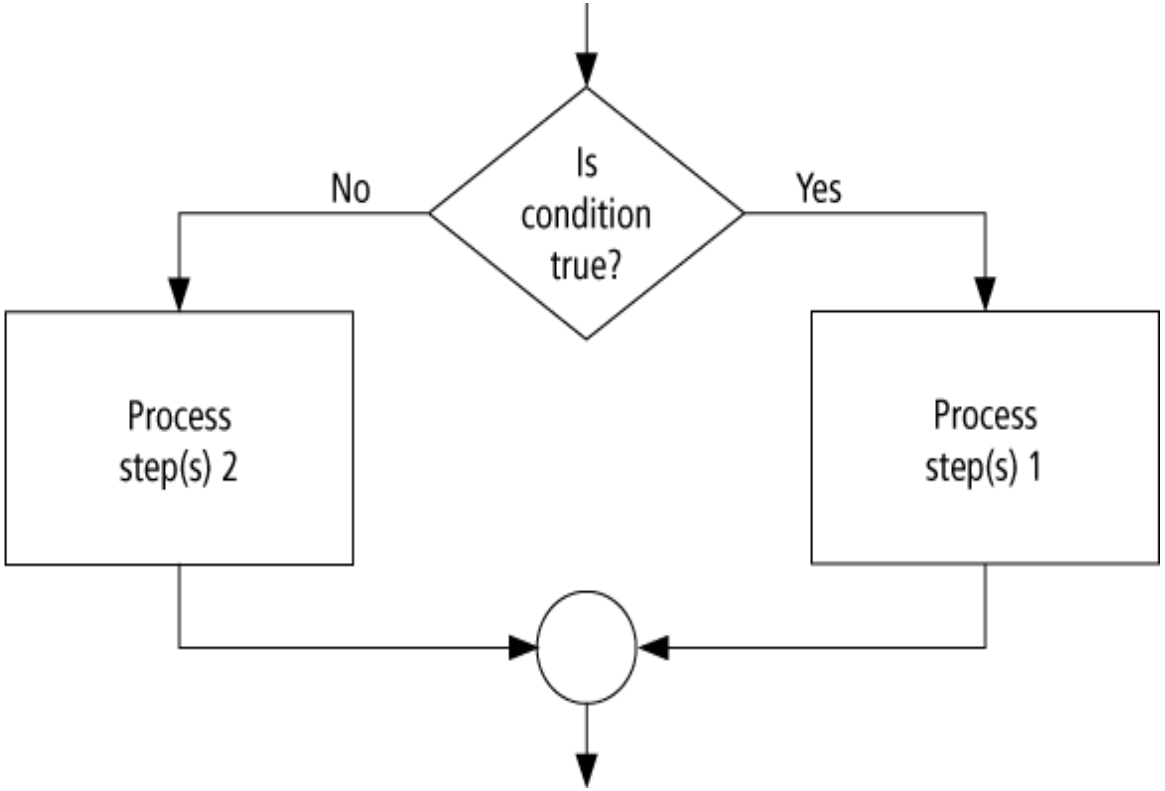
volume = length * width * height

Store volume

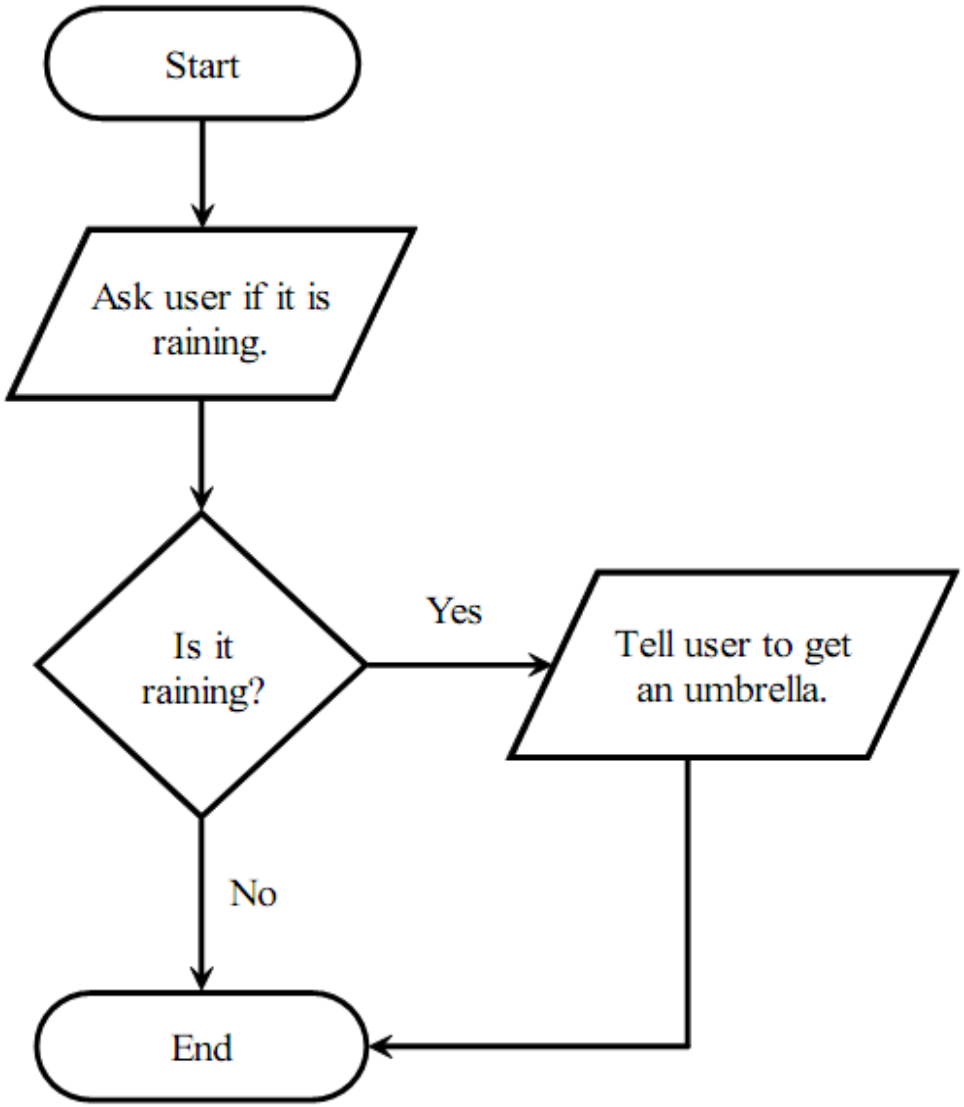
Display volume

Decision

```
If condition is true Then
      Process step(s) 1
  Else
      Process step(s) 2
End If
```

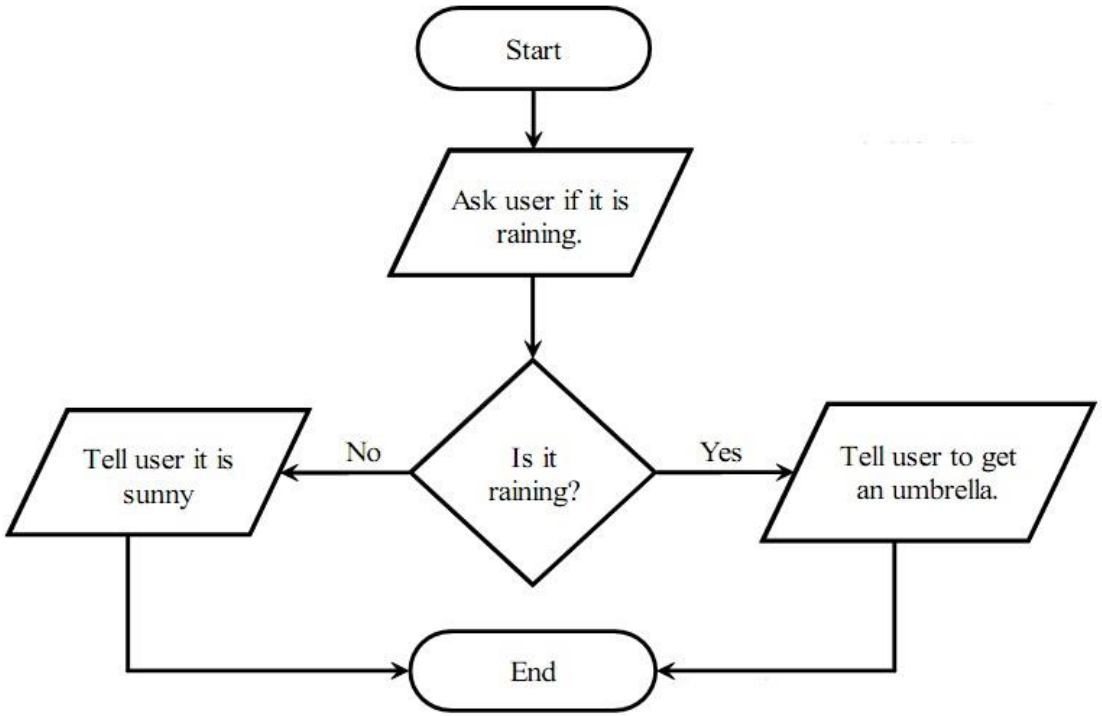


Decision



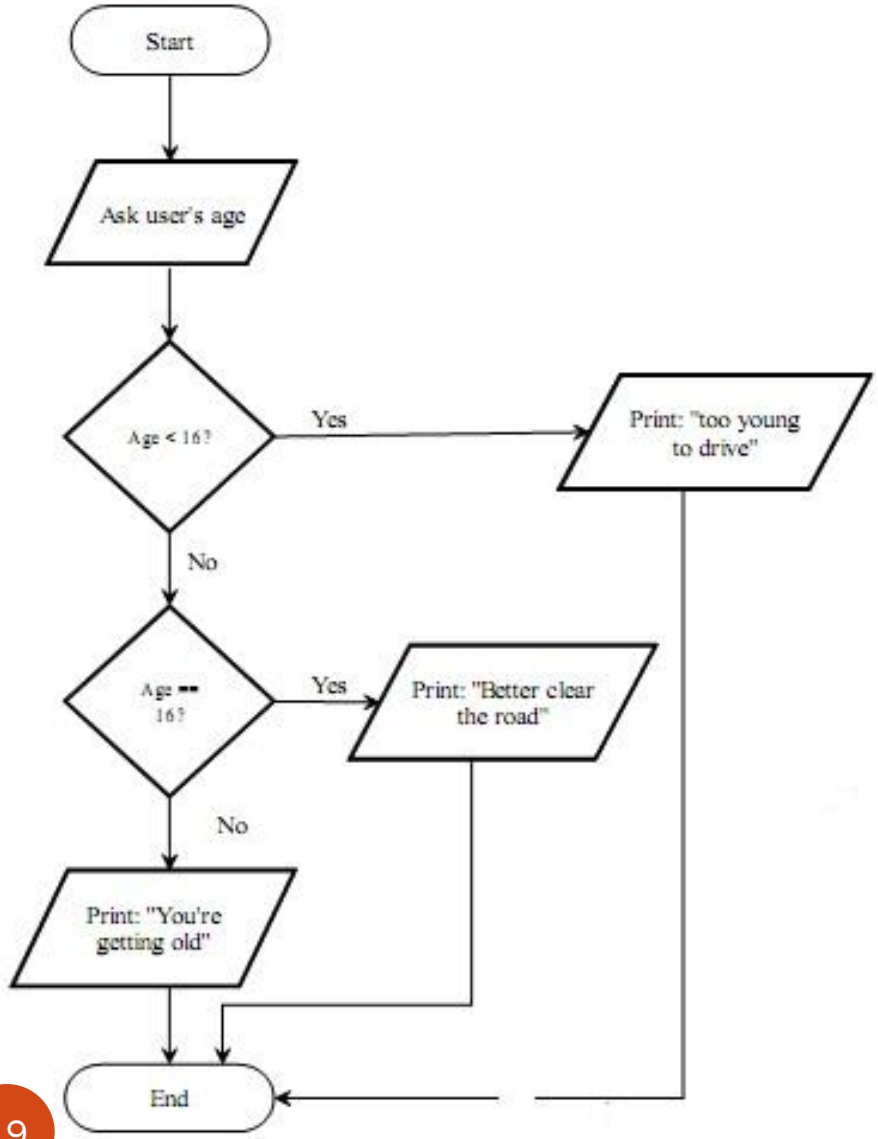
Ask user if it is raining
Get answer
If answer is yes
Display "Get an umbrella"

Decision



Ask user if it is raining
Get answer
If answer is yes
Display “Get an umbrella”
Else
Display “It is sunny”

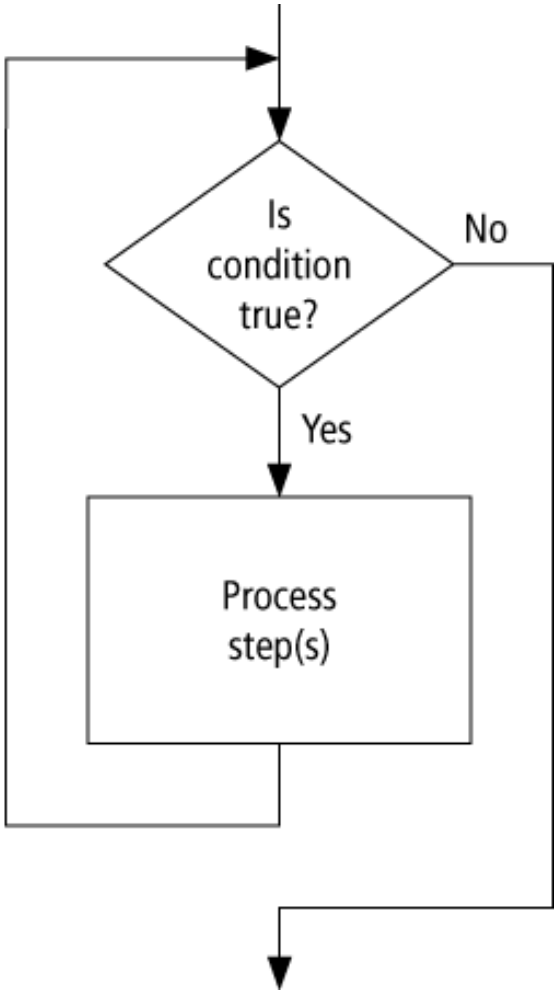
Decision



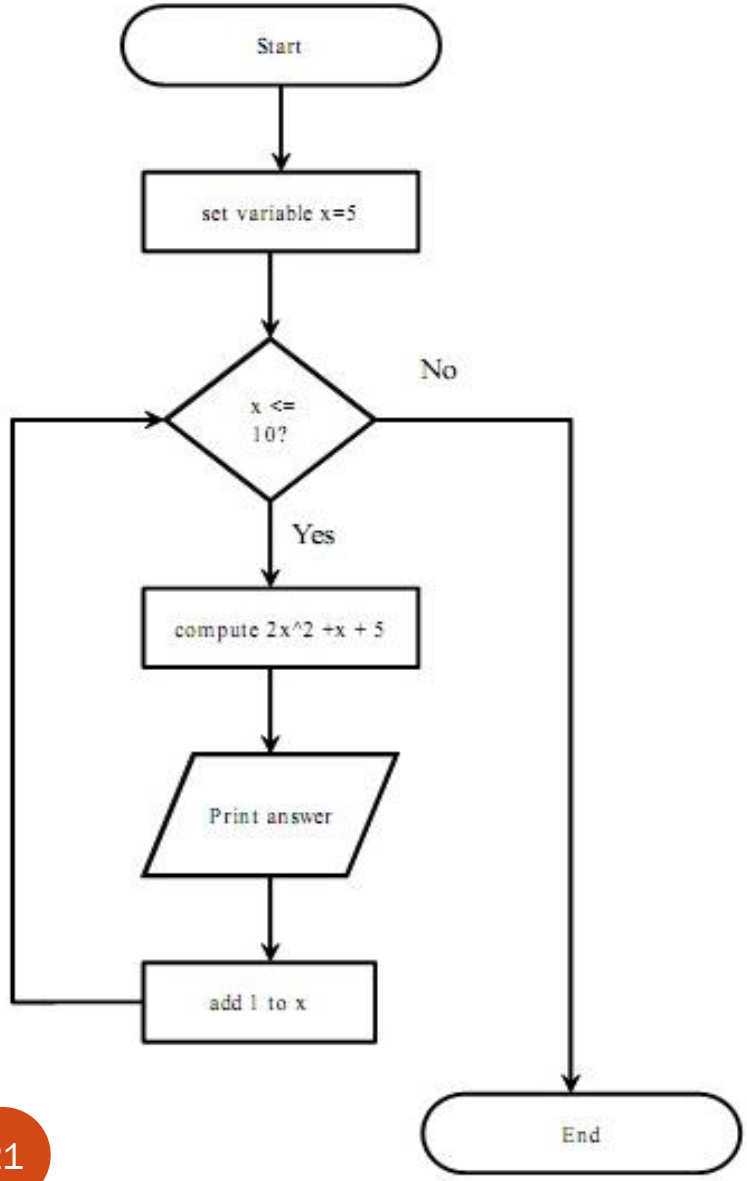
Ask user for age
Get age
If age < 16
    Display "Too young to drive"
Else if age = 16
    Display "Better clear the road"
Else
    Display "You're getting old"

Looping

```
Do While condition is true
    Process step(s)
Loop
```



Requiring Iteration



Set x to 5
While x <= 10
    Compute $2x^2 + x + 5$
    Store answer
    Quit
    Print answer
    Increment x
    Return to While Statement
Quit

Example 1

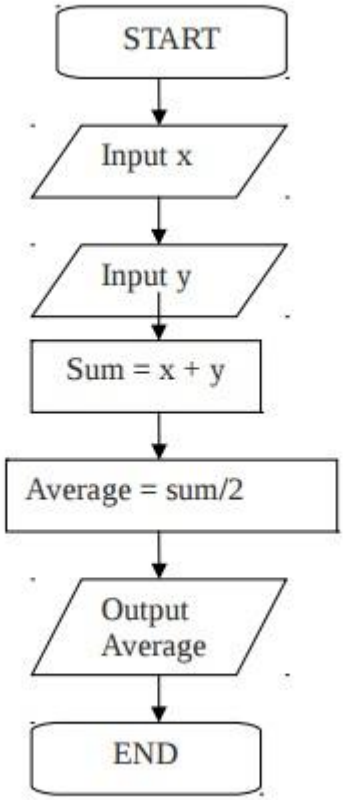
Write an algorithm in pseudocode and draw a flowchart to find the average of two numbers.

Input: Two numbers

1. Add the two numbers
2. Divide the result by 2

Output: Return the result by step 2

End



Example 2

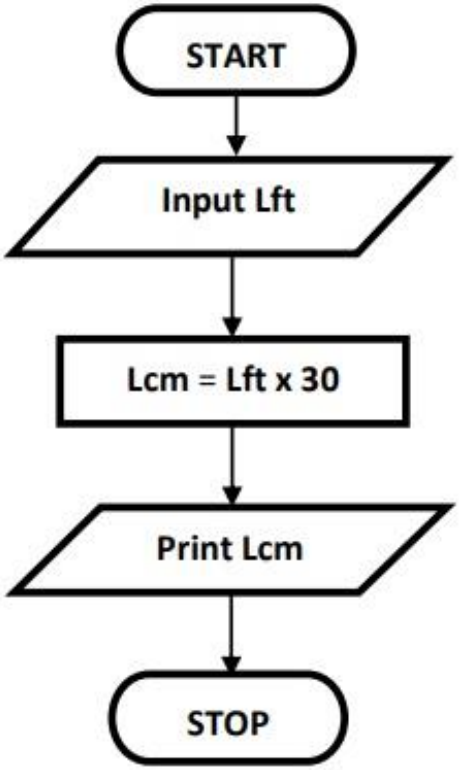
Write an algorithm and draw a flowchart to convert the length in feet to centimeter.

Input: the length (Lft)

Calculate the length (Lcm) in cm

by multiplying LFT with 30

Output: Print length in cm (Lcm)



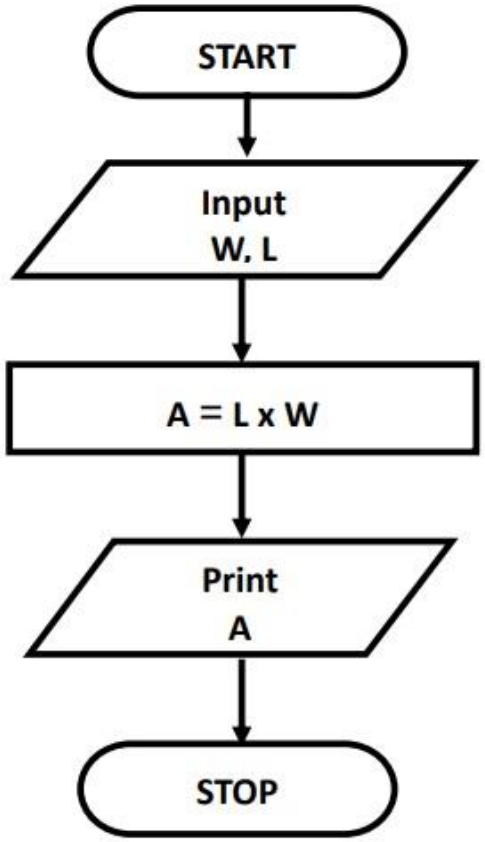
Example 3

Write an algorithm and draw a flowchart to 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

Output: Print The Area (A)



Example 5

Write an algorithm in pseudocode and draw a flowchart to designate a grade as either passing or failing.

Input: One number

1.if (the number is greater than or equal 60) then

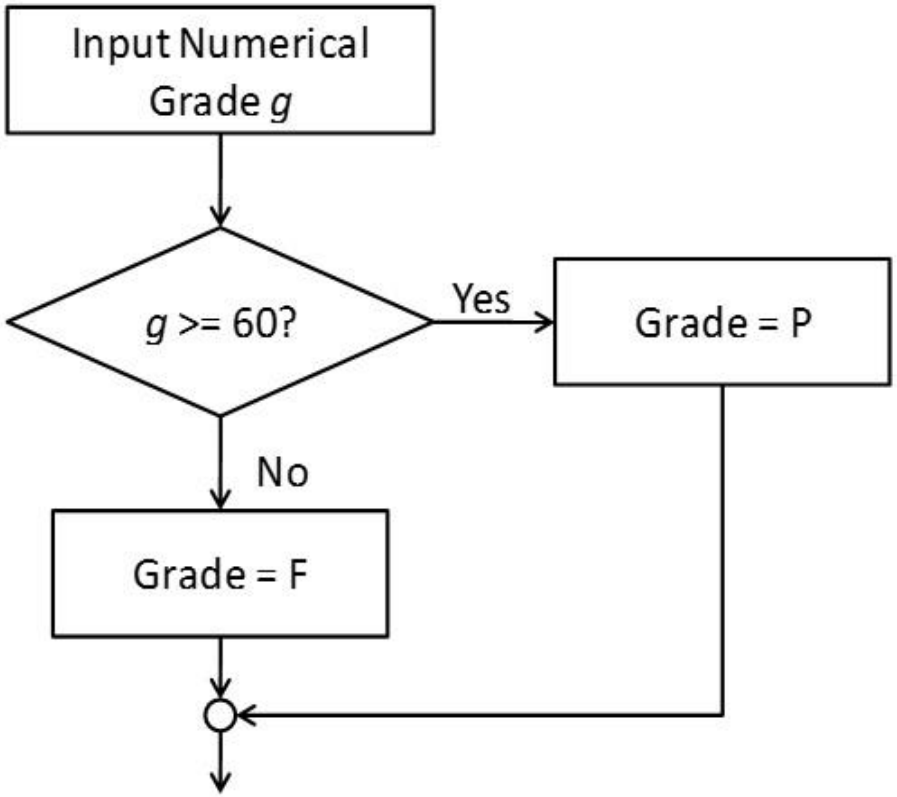
1.1 Set the grade to "P"

End if

2. if (the number is less than 60) then

2.1 Set the grade to "B"

End if



Example 6

Write an algorithm and draw a flowchart to read Two values, determines the largest value and prints the largest value with an identifying message.

Input: *Input* VALUE1, VALUE2

if (VALUE1 > VALUE2) *then*

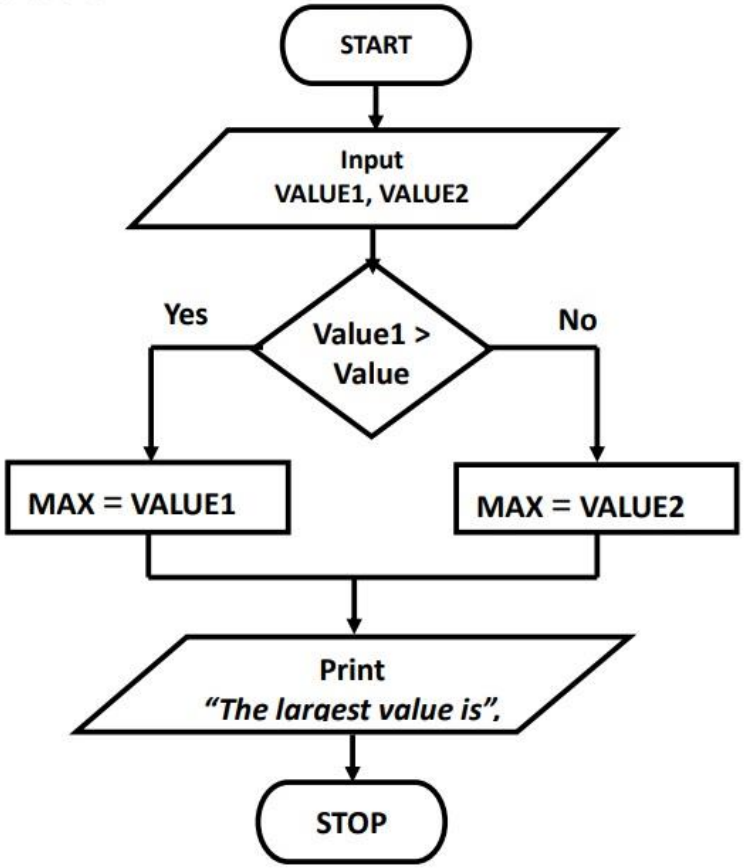
MAX ← VALUE1

else

MAX ← VALUE2

endif

Output: Print "The largest value is", MAX



Example 7

Write an algorithm and draw a flowchart to read THREE values, determines the largest value and prints the largest value with an identifying message.

Input *Input Three numbers A, B, and C.*

IF (A > B) then

if (A > C) then

Print A

else

Print C

endif

Else

if (B > C) then

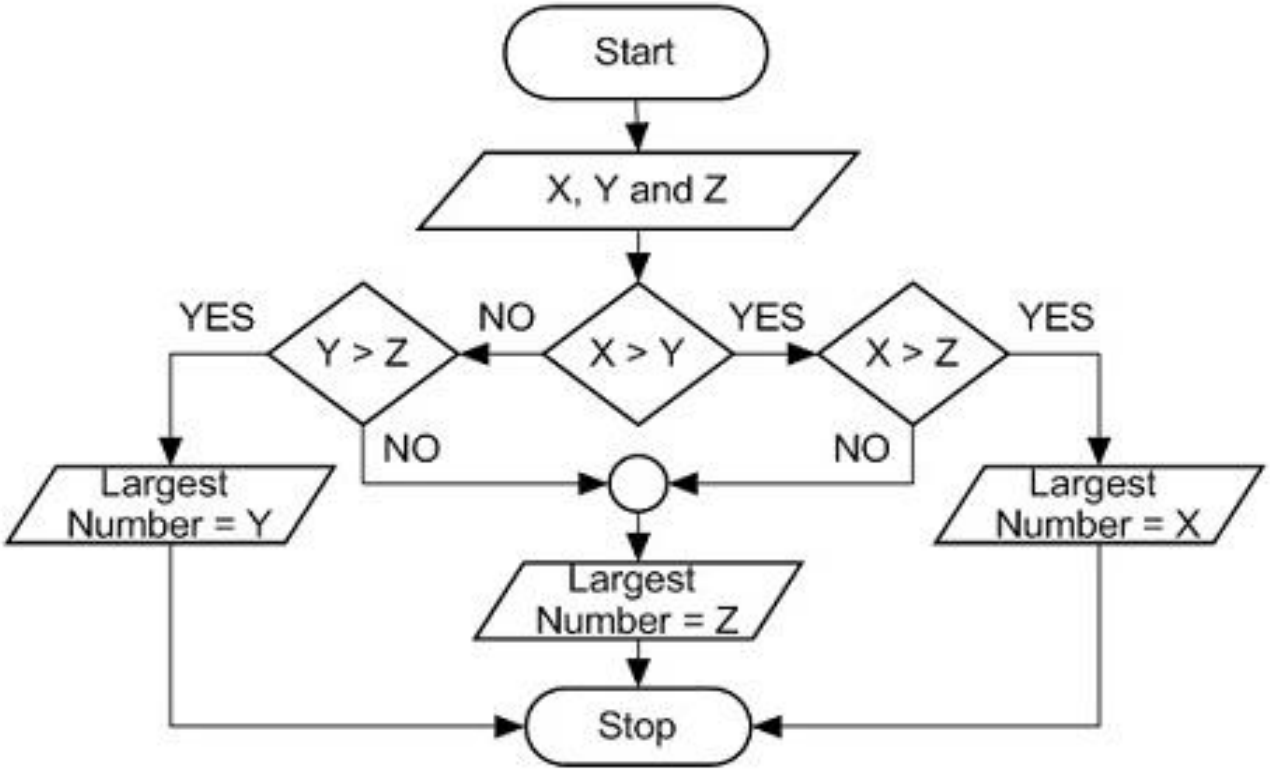
Print B

else

Print C

endif

Endif

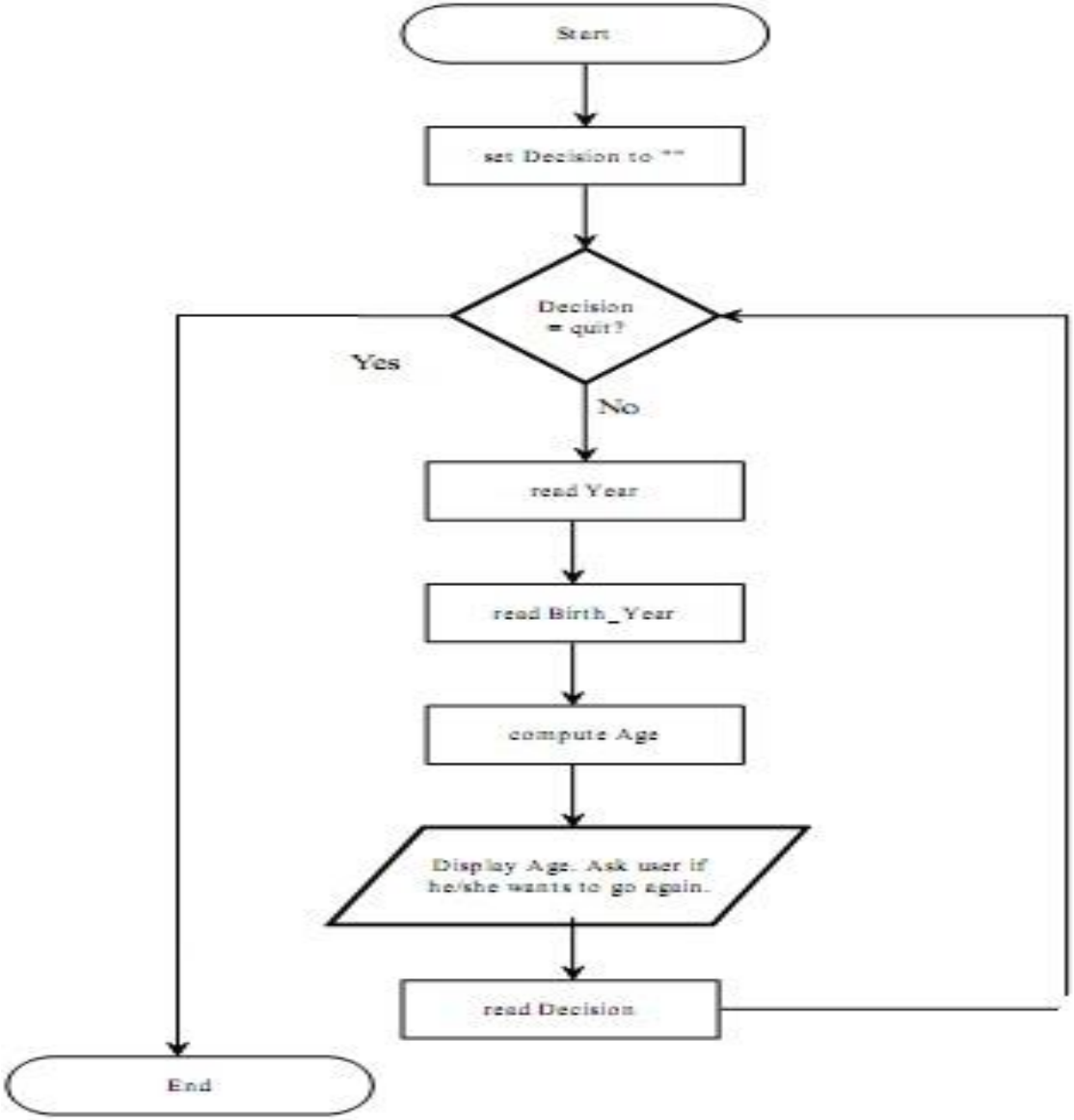


Example 8

Write an algorithm in pseudocode and draw a flowchart to Get the current year from the user, get the user's birth year from the user, compute and display the users age. Ask if the user wishes to continue or to quit. If "continue", repeat the program. If "quit", exit the program

```
Set Decision to ""
While Decision != "quit"
        Read Year
        Read Birth_year
        Age = Year – Birth_Year
        Display Age Display "Do you wish to quit?"
        Read Decision
        Return to While Statement
Quit
```

Example 8


Start
set Decision to ""
Decision
= quit?
Yes
No
read Year
read Birth_Year
compute Age
Display Age. Ask user if
he/she wants to go again.
read Decision
End

Example 9

Write an algorithm and draw a flowchart to print odd numbers between 1 and 100

Count = 1

Loop:

Print Count

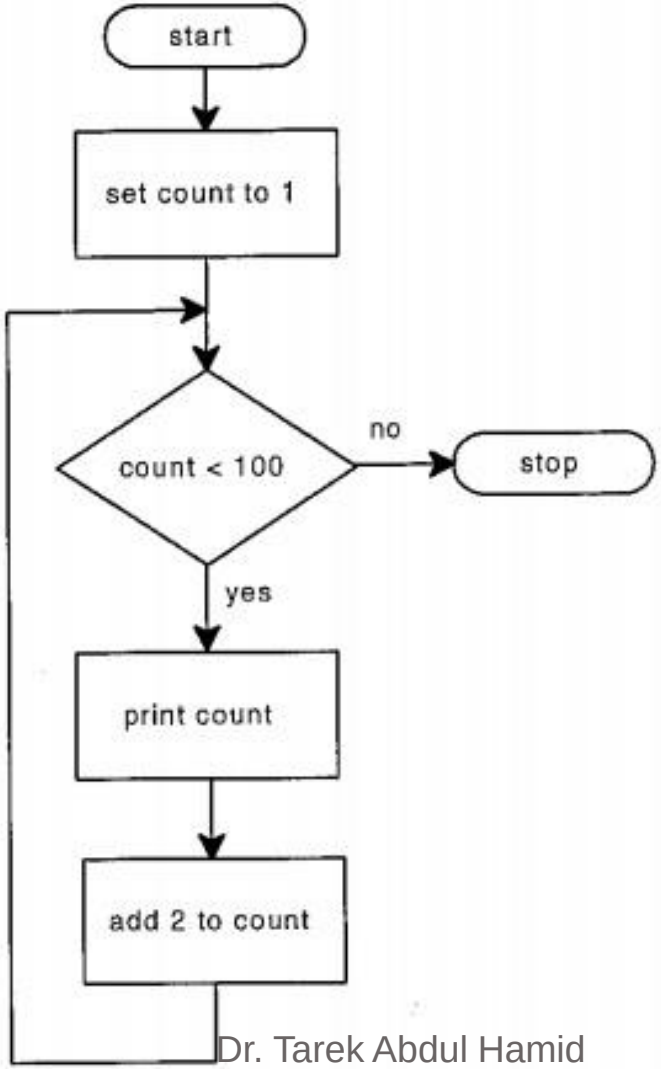
Count = Count + 2

If Count <100

Go to Loop

Else

Stop



Example 10

Write an algorithm and draw a flowchart to calculate N factorial

Input: n

Repeat : F = F * M

If M !=N

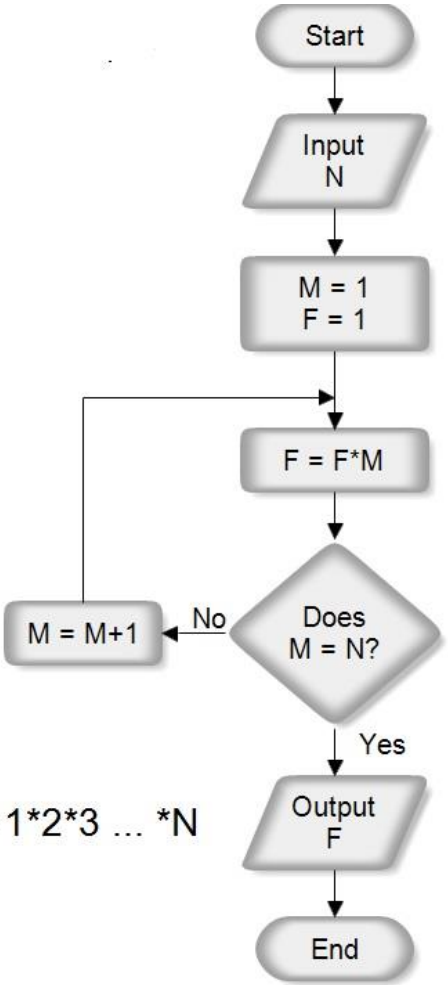
M= M + 1

Go to Repeat

Else

Print F

Output: F



F = 1*2*3 ... *N

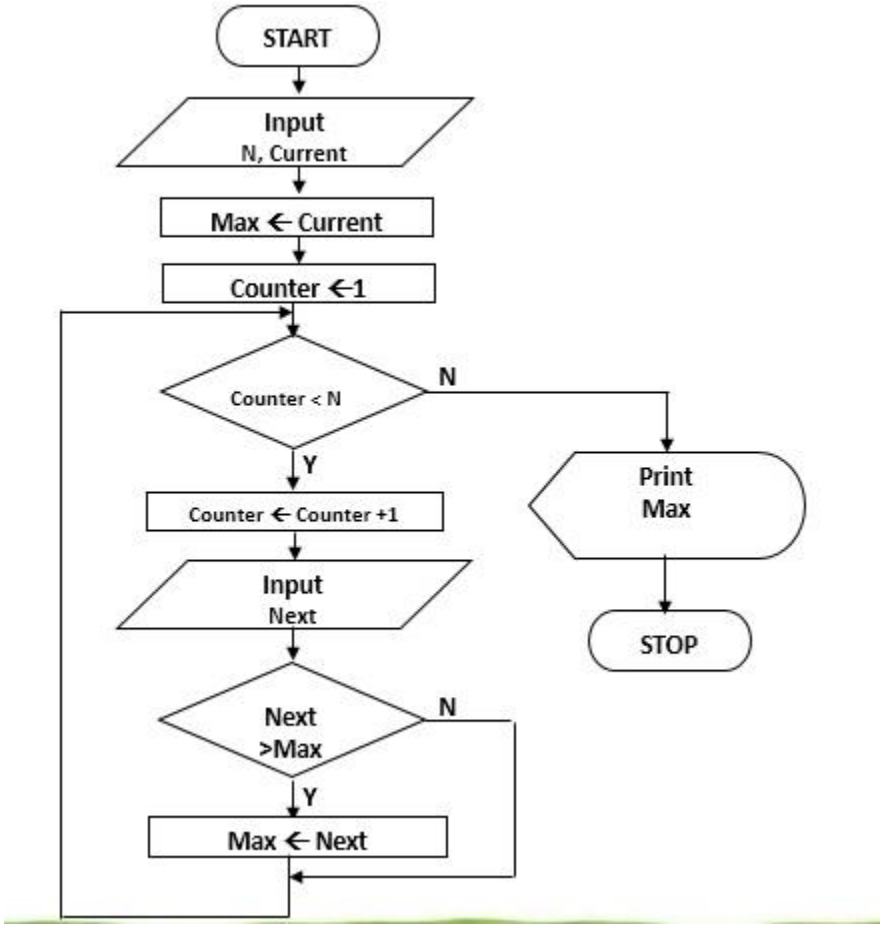
Example 12

Write an algorithm and draw a flowchart to find and print the largest of N (N can be any number) numbers.

Step 1: Input N
Step 2: Input Current
Step 3: Max = Current
Step 4: Counter = 1
Step 5: While (Counter < N)

Repeat steps 5 through 8
Step 6: Counter = Counter + 1
Step 7: Input Next
Step 8: If (Next > Max) then

Max = Next
endif
Step 9: Print Max




Thanks!