

# PROGRAM CONTROL STRUCTURE

- Programming control structure controls how the each statement (step) of the algorithm flows (executes).
- Control structures in most programming languages typically include the following
  - Sequence
  - Selection
  - Iteration

# PROGRAM CONTROL STRUCTURE

## A. Sequence

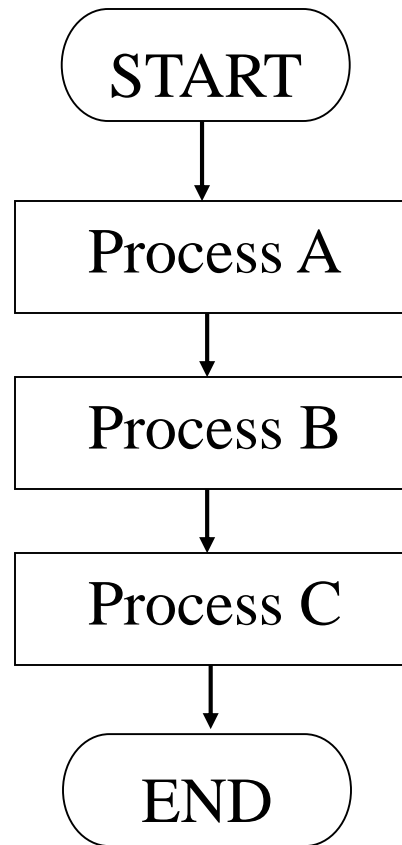
The sequence control structure is the straightforward execution of one processing step after another.

### *Sequence's pseudo code:*

1. Start
2. Process A
3. Process B
4. Process C
5. End

# PROGRAM CONTROL STRUCTURE

*Sequence's flowchart:*



# PROGRAM CONTROL STRUCTURE

## B. Selection

- ✓ The selection control structure is the presentation of a condition and two actions between two actions.
- ✓ The choice depending on whether the condition is true or false.
- ✓ This control structure represents the decision-making process.
- ✓ There are three main types of selection control structure: IF, IF-ELSE and Nested IF.

# PROGRAM CONTROL STRUCTURE

## B. Selection

- ✓ IF structure occurs when a choice is made between two alternative paths, based on the result of a condition being true or false.
- ✓ Program will execute specified tasks only if the result of condition is true.
- ✓ If the condition is false, then no processing will take place and the IF statement will be bypassed.

# PROGRAM CONTROL STRUCTURE

## B. Selection

✓ IF IF-ELSE structure occurs when a choice is made between two alternative paths, based the result of a condition being true or false.

✓ Nested IF structure occurs when there is (are) IF statement(s) within an IF statement. .

# PROGRAM CONTROL STRUCTURE

## B. Selection : IF structure

IF ( condition ) THEN

PROCESS 1

PROCESS 2

PROCESS 3



List of process  
when the  
condition is  
true

ENDIF

# PROGRAM CONTROL STRUCTURE

## B. Selection : IF – ELSE structure

IF ( condition ) THEN

PROCESS 1 } List of process  
PROCESS 2 } when the  
PROCESS 3 } condition is  
TRUE

ELSE

PROCESS 4 } List of process  
PROCESS 5 } when the  
PROCESS 6 } condition is  
FALSE

ENDIF



# PROGRAM CONTROL STRUCTURE

## *IF's pseudo code:*

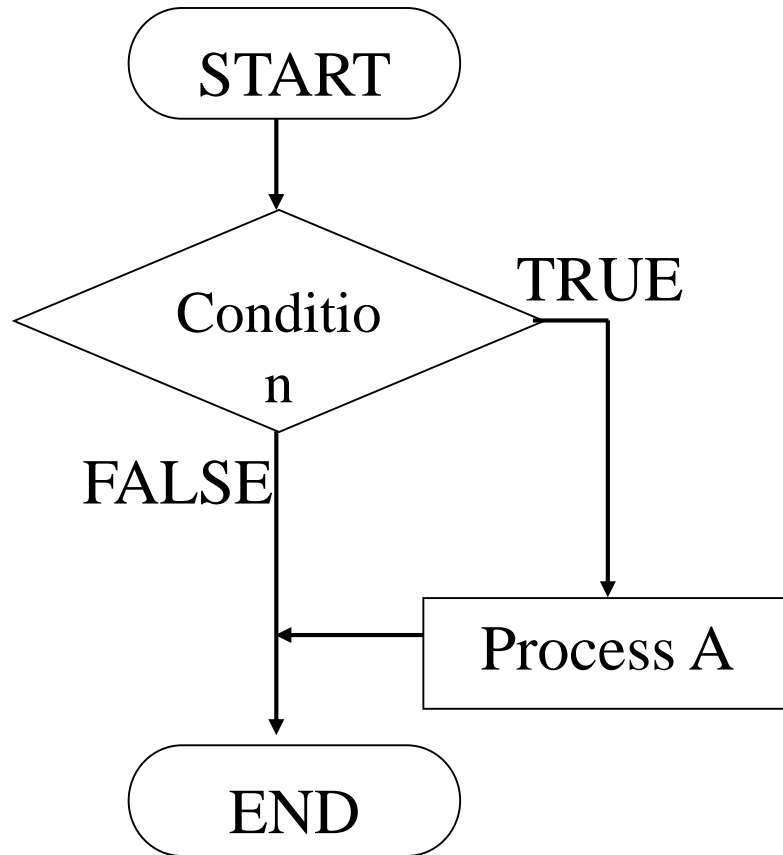
1. Start
2. IF (Condition) Then
3.     Process A  
    End IF
4. End

## *IF-ELSE's pseudo code:*

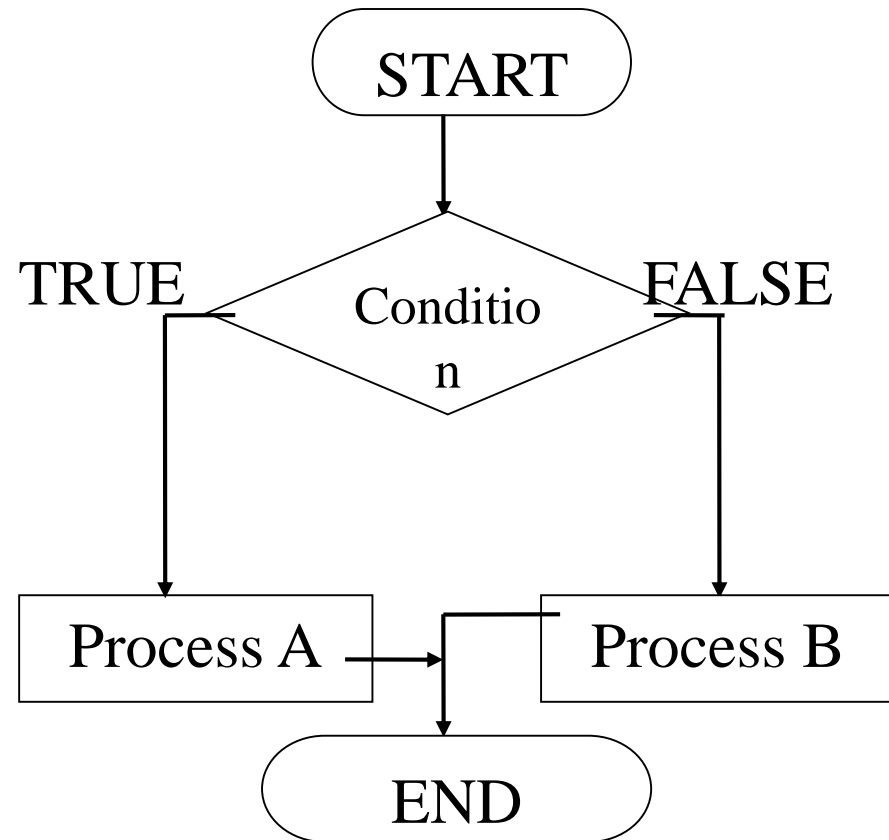
1. Start
2. IF (Condition) Then
3.     Process A  
    ELSE
4.     Process B  
    End IF
5. End

# PROGRAM CONTROL STRUCTURE

*IF's flowchart:*



*IF-ELSE's flowchart:*



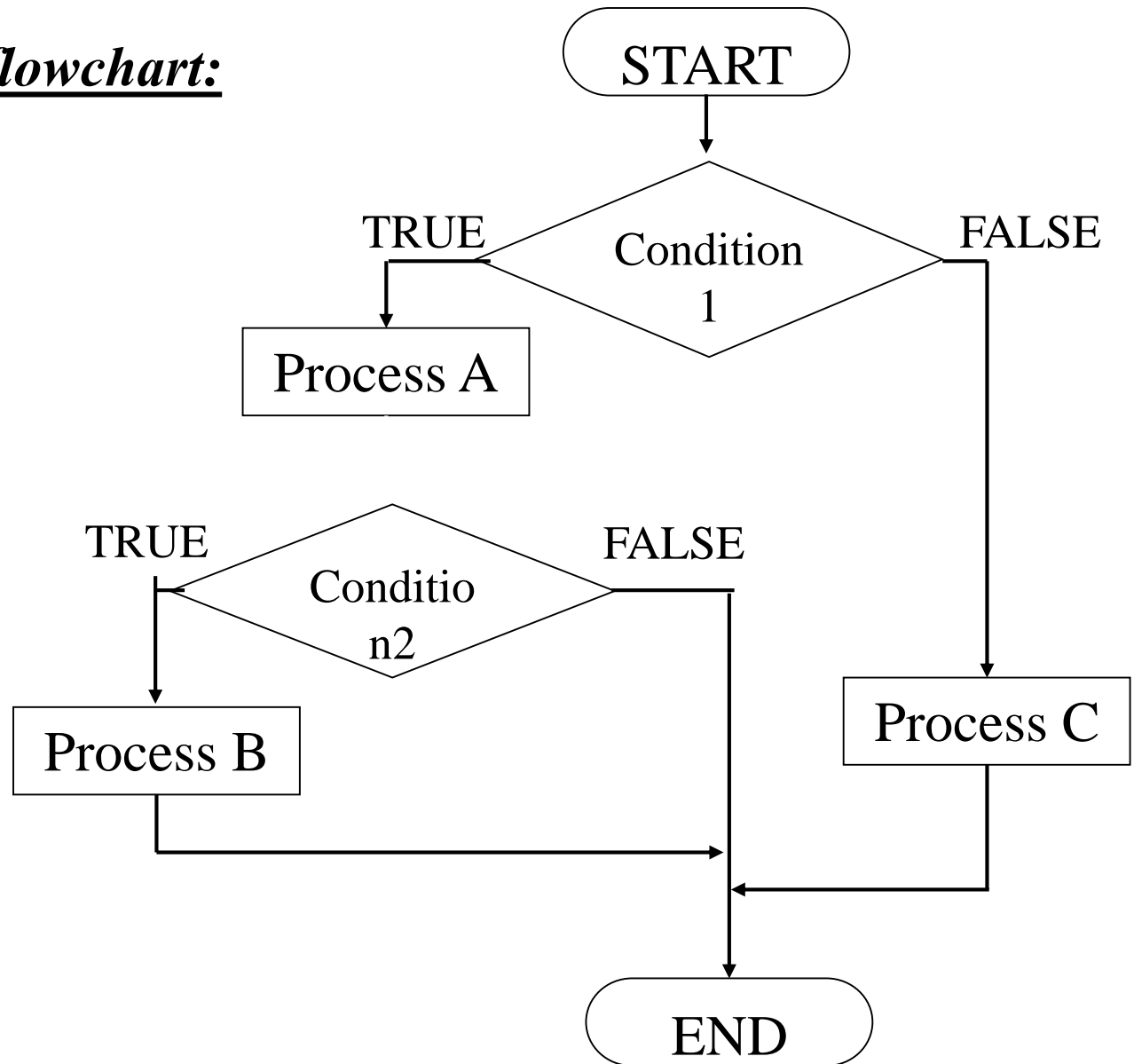
# PROGRAM CONTROL STRUCTURE

## *Nested IF's pseudo code:*

1. Start
2. IF (Condition1) Then
3.     Process A
4.     *IF (Condition2) Then*
5.         *Process B*
6.         *ENDIF*
7.     ELSE
8.     Process C
9.     End IF
10. End

# PROGRAM CONTROL STRUCTURE

*Nested IF' flowchart:*



# PROGRAM CONTROL STRUCTURE

## C. Iteration (Looping)

✓ The iteration control structure can be defined as the representation of a set of instructions to be performed repeatedly, as long as the condition is satisfied (TRUE or FALSE).

✓ There are two main types of iteration control structure: DOWHILE and REPEAT\_UNTIL.

# PROGRAM CONTROL STRUCTURE

- ✓ DOWHILE structure will test the condition at the *beginning of the loop*.
- ✓ A statement or a group of statements will be executed in top-down fashion from the beginning of the loop until the end of the loop, then go back to test the condition again.
- ✓ The loop will be terminated if the result of condition is *false*.

# PROGRAM CONTROL STRUCTURE

## C. Iteration : DOWHILE structure

DOWHILE ( condition )

PROCESS 1	}	<b>List of process when the condition is TRUE</b>
PROCESS 2		
PROCESS 3		

ENDDO

# PROGRAM CONTROL STRUCTURE

## *DOWHILE's pseudo code:*

1. Start

... ..

4. DOWHILE (Condition)

5. Process A

6. Process B

7. Process C

8. Process D

9. ....

.... ..

n. Process n

ENDDO

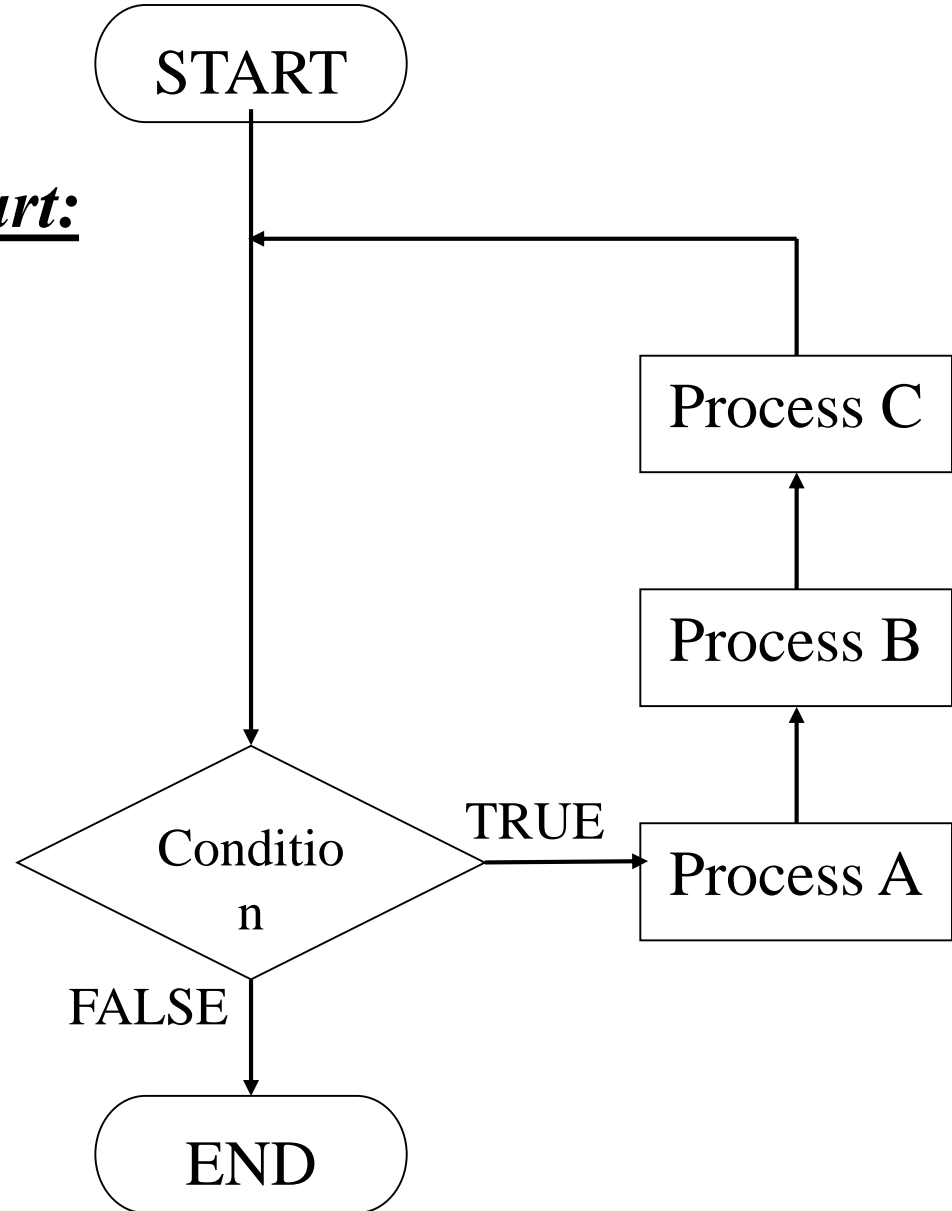
....

End



# PROGRAM CONTROL STRUCTURE

**DOWHILE's flowchart:**



# PROGRAM CONTROL STRUCTURE

- ✓ REPEAT\_UNTIL structure is similar to DOWHILE structure, in that a group of statements are repeated in accordance with a specified condition.
- ✓ But REPEAT\_UNTIL tests the condition at the end of the loop.
- ✓ This means that the statements within the loop will be execute once before the condition is tested.
- ✓ The loop will be terminated if the result of condition is *True*.

# PROGRAM CONTROL STRUCTURE

## C. Iteration : REPEAT-UNTIL structure

REPEAT

PROCESS 1 } List of process  
PROCESS 2 } when the  
PROCESS 3 } condition is  
FALSE

UNTIL ( condition )

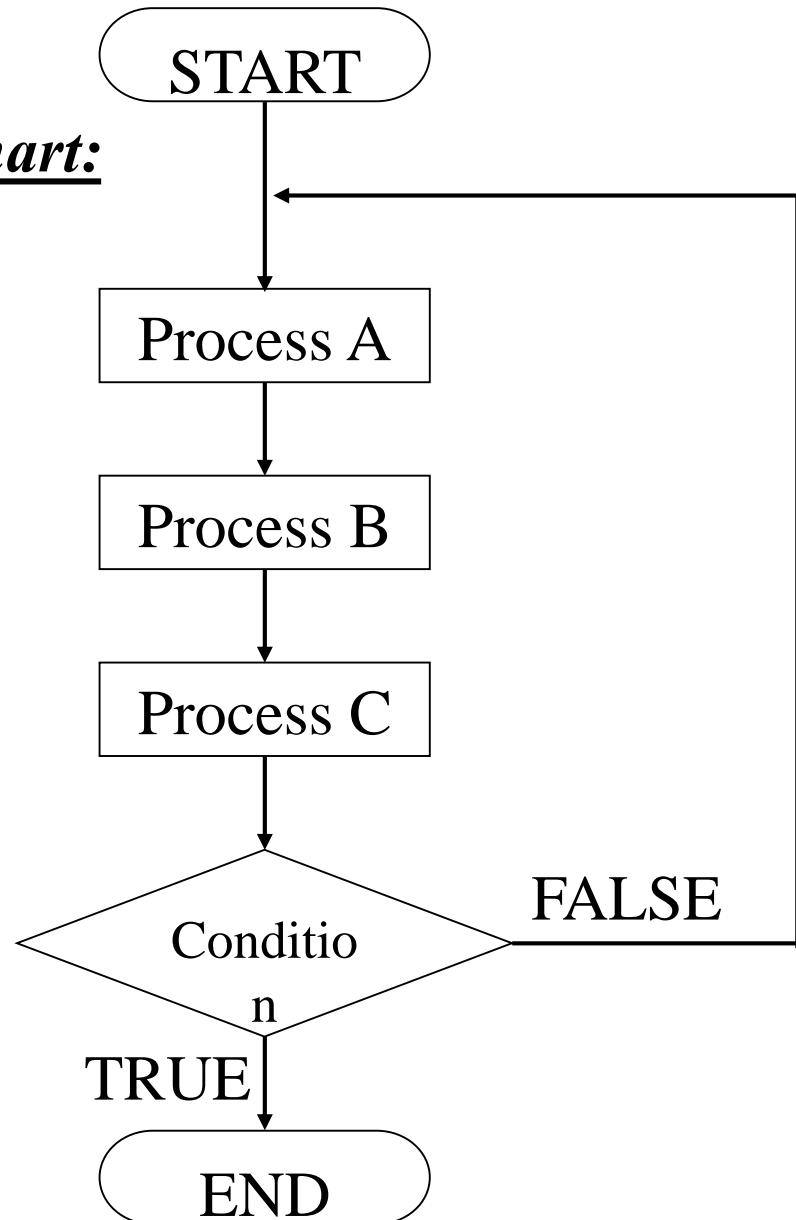
# PROGRAM CONTROL STRUCTURE

## *REPEAT UNTIL's pseudo code:*

1. Start
- .
3. REPEAT
4. Process A
5. Process B
6. Process C
7. Process D
8. Process E
- .
- n. Process n
- UNTIL (Condition)
- .
15. End

# PROGRAM CONTROL STRUCTURE

*REPEAT UNTIL's flowchart:*



# PROGRAM CONTROL STRUCTURE

Example 1:

- Problem:

- Create a program to read 10 numbers from keyboard. Program will calculate the average of that numbers. Print the number and the average.

# PROGRAM CONTROL STRUCTURE

Example 1:

- Problem Analysis:

1. INPUT: 10 number

2. PROCESS: Calculate the average by using the formula;  $\text{average} = \text{total} / 10$

3. OUTPUT: 10 number and average

# PROGRAM CONTROL STRUCTURE

Example 1:

- Program Modeling: Pseudo code

1. **START**

2. **Declare a name for every value to be input -a1, a2, a3, a4, a5, a6, a7, a8, a9, a10**

**Declare a name for the value of sum -TOTAL**

**Declare a name for a value of average -AVERAGE**

**Set all = 0**

3. **INPUT 10 numbers**

4. **Calculate the total by using**

**the formula;  $TOTAL = a1 + a2 + \dots + a10$**



# PROGRAM CONTROL STRUCTURE

Example 1:

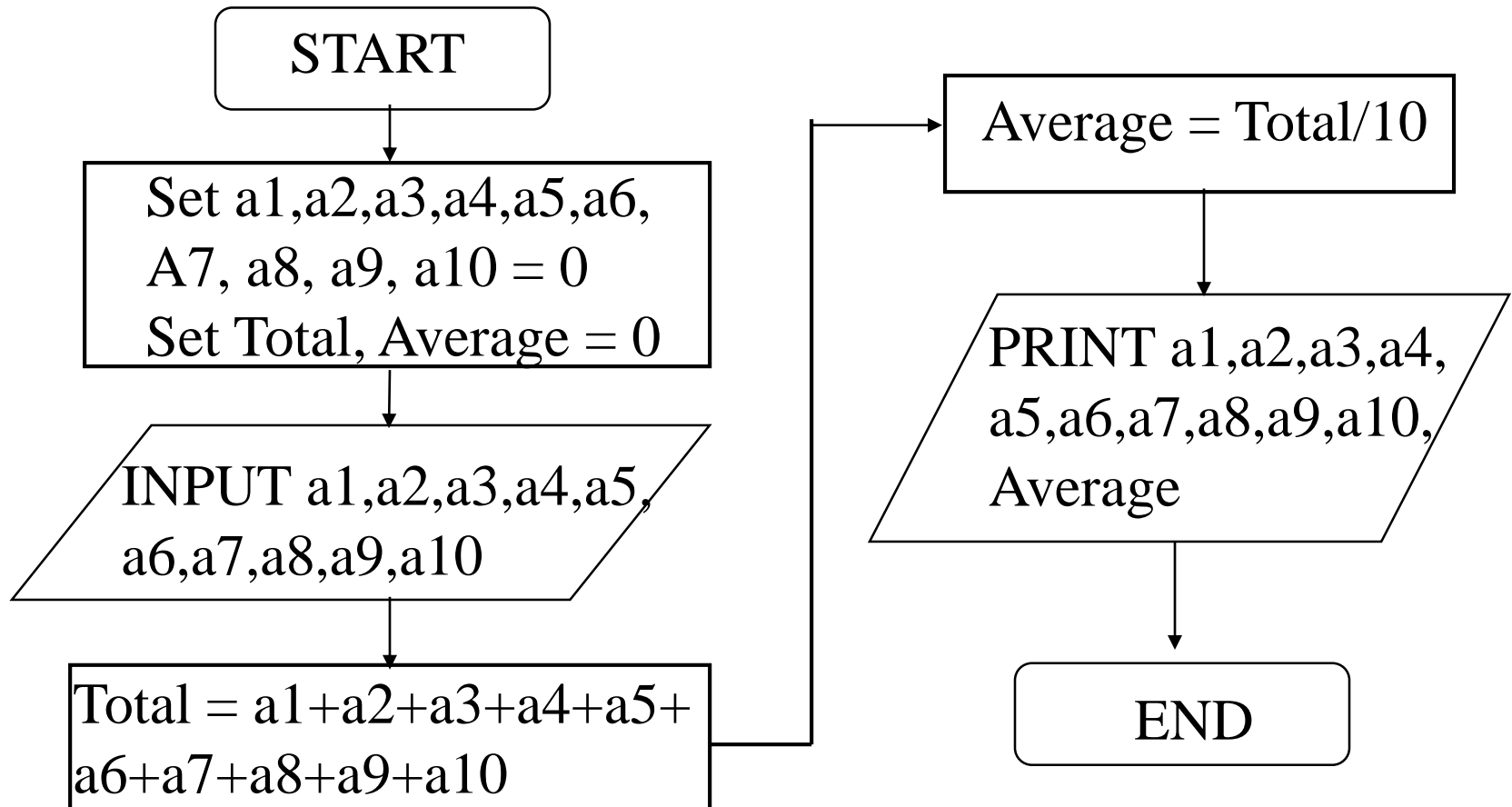
- Program Modeling: Pseudo code

**5. Calculate the average by using  
the formula;  $AVERAGE = TOTAL/10$**   
**6. PRINT 10 numbers and AVERAGE**  
**7. END**

# PROGRAM CONTROL STRUCTURE

Example 1:

## ■ Program Modeling: Flowchart



# PROGRAM CONTROL STRUCTURE

Example 2:

- **Problem: Add three numbers**

A program is required to read three numbers from a user, add them together, and print their total on the screen.

# PROGRAM CONTROL STRUCTURE

## Example 2:

- Problem Analysis:

1. INPUT: number1, number2, number3
2. PROCESS: read three numbers  
add numbers together  
print total to the screen
3. OUTPUT: total

# PROGRAM CONTROL STRUCTURE

## Example 2:

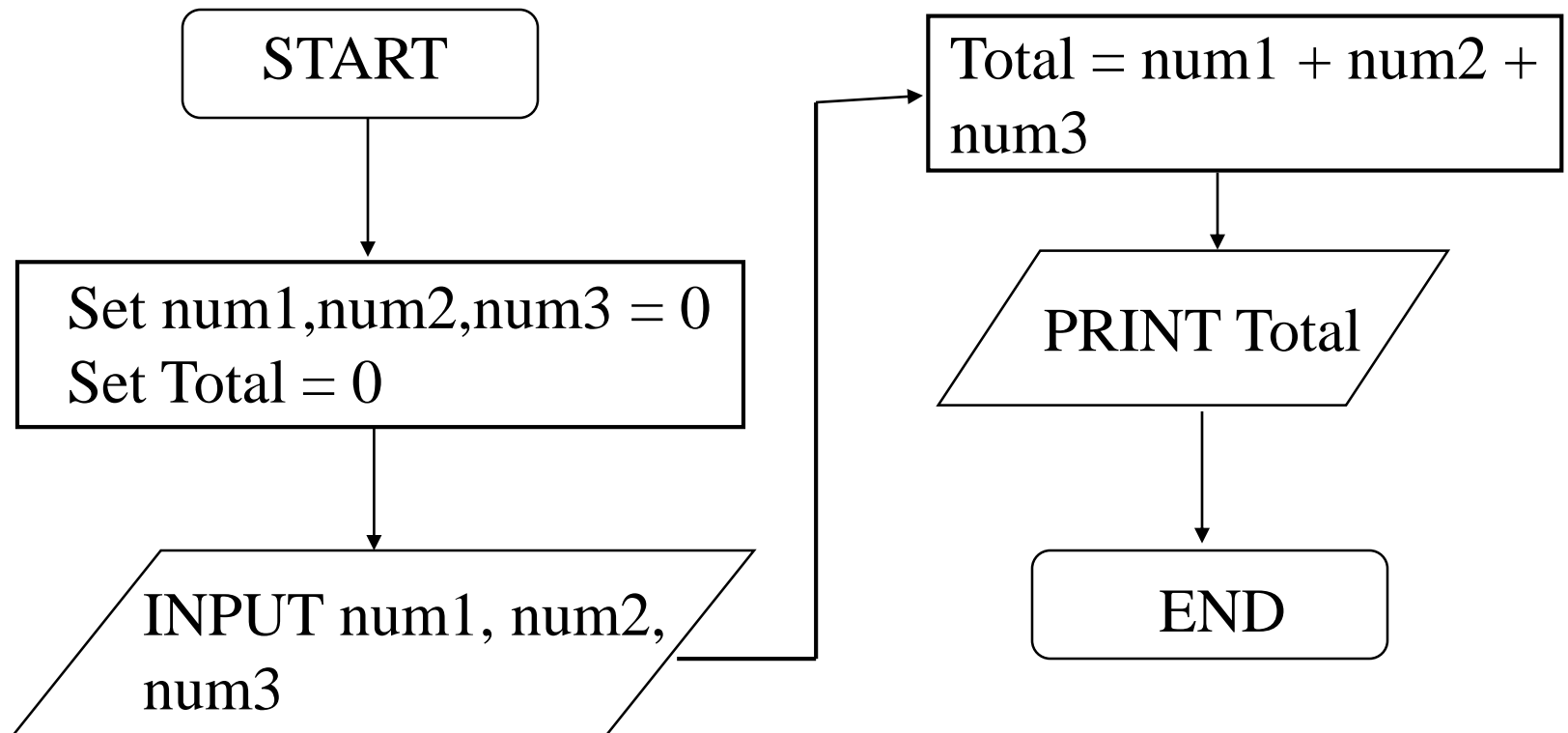
- Program Modeling: Pseudo Code

1. START
2. Declare num1, num2, num3 and Total  
Set num1, num2, num3 and Total = 0
3. INPUT num1, num2, num3
4. Add numbers together by using the  
formula;  $Total = num1 + num2 + num3$
5. PRINT Total
6. END

# PROGRAM CONTROL STRUCTURE

Example 1:

- Program Modeling: Flowchart



# PROGRAM CONTROL STRUCTURE

Exercise 1:

## **Probation?**

A program is required to read student id and GPA from a user. If an input GPA is less than 2.00, print “This student is on probation”. In all cases print “Finish” in the end.

# PROGRAM CONTROL STRUCTURE

Exercise 2:

## **Compare two numbers version1**

A program is required to read two numbers from a user, compare them to find out whether number1 is greater than number2 or not. Print the result of the comparison on the screen. i.e. number1 is greater than number2



# PROGRAM CONTROL STRUCTURE

## Exercise 3:

### **Compare two numbers version2**

At this time, the complexity of the program will be increased. A program is required to read two numbers from the user, compare them to find out whether number1 is greater than number2 or number2 is greater than number1 or they are equal. Print the result of comparing on the screen. i.e.  
number1 is equal to number2

# PROGRAM CONTROL STRUCTURE

Exercise 4:

## **Find average score**

A program will accept 10 students' scores then calculate the average score. After finishing the calculation, the program will print the average score on the screen.

# PROGRAM CONTROL STRUCTURE

Exercise 5:

## **Find maximum score**

A program will accept 10 students' scores then calculate the maximum score. After finishing the calculation, the program will print the maximum score on the screen.