UTM

Control Structure of Algorithm: Repetition

UTM

Repetition Structure

- Specifies a block of one or more statements that are repeatedly executed until a condition is satisfied.
- Usually the loop has two important parts:
  1. An expression that is tested for a true/false,
  2. A statement or block that is repeated as long as the expression is true
- 2 styles of repetition or loop
  1. Pre-test loop
  2. Post test loop

Repetition Structure - Counters

• Counter: Can be used to control execution of the loop (loop control variable)

• It will increment or decrement each time a loop repeats
• Must be initialized before entering loop

Repetition Structure: Pre-Test Loop

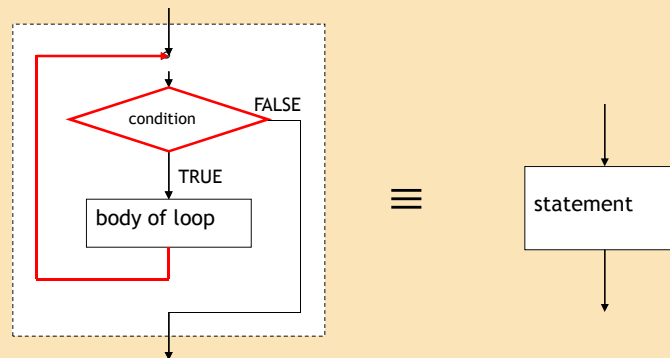• Pseudo code – requires the use of the keywords `while` for pre-test loop.

```
Algorithm: one choice selection
:
n.   While condition
         n.1 statement
              :
n+1. end_while
:
```

Repetition Structure (*cont..*)

while Loop
(pre-test loop)
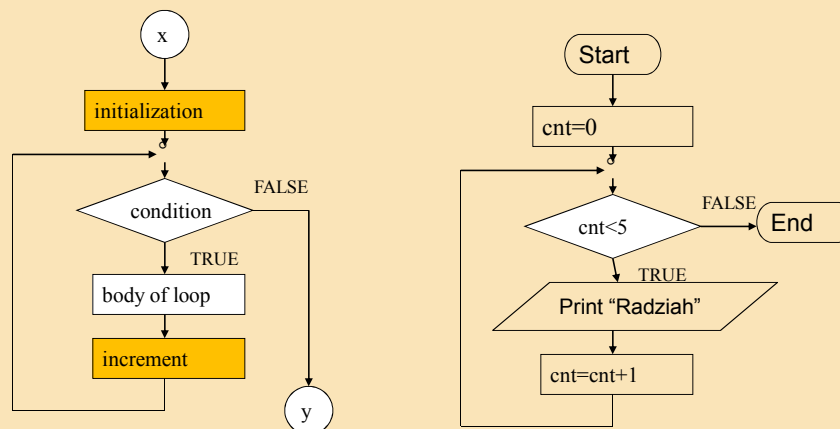

FALSE
condition
TRUE
body of loop

≡

statement

While a set condition is true, repeat statement (body of loop)

Repetition Structure (*cont..*)

x
initialization
FALSE
condition
TRUE
body of loop
increment
y

Start
cnt=0
FALSE
cnt<5 → End
TRUE
Print "Radziah"
cnt=cnt+1

Pre-test loop steps summary

- Counter-controlled loop
  - Initialization of counter: counter = 0
  - Testing of counter value: counter > n
  - Updating of counter value (increase by 1) during each iteration: counter=counter+1

Example

- Suppose we want to write a program to compute a sum of the first 10 positive integers.
- Steps:
  - How many repetition?
    - Initialization
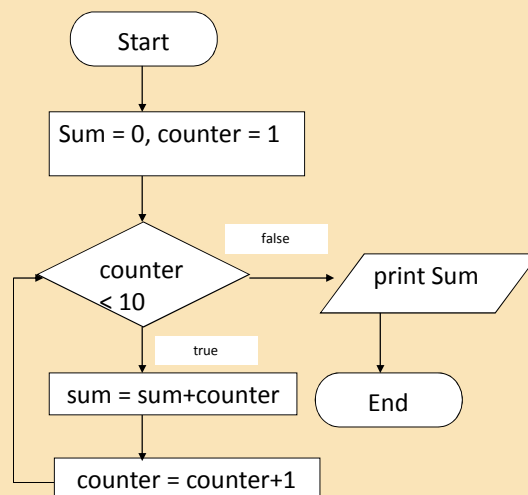   - Condition to check for the counter?
    - Update of counter

Pseudo code

1. Start
2. Set sum=0, counter = 0
3. While (counter <10)
   1. Sum=sum+counter
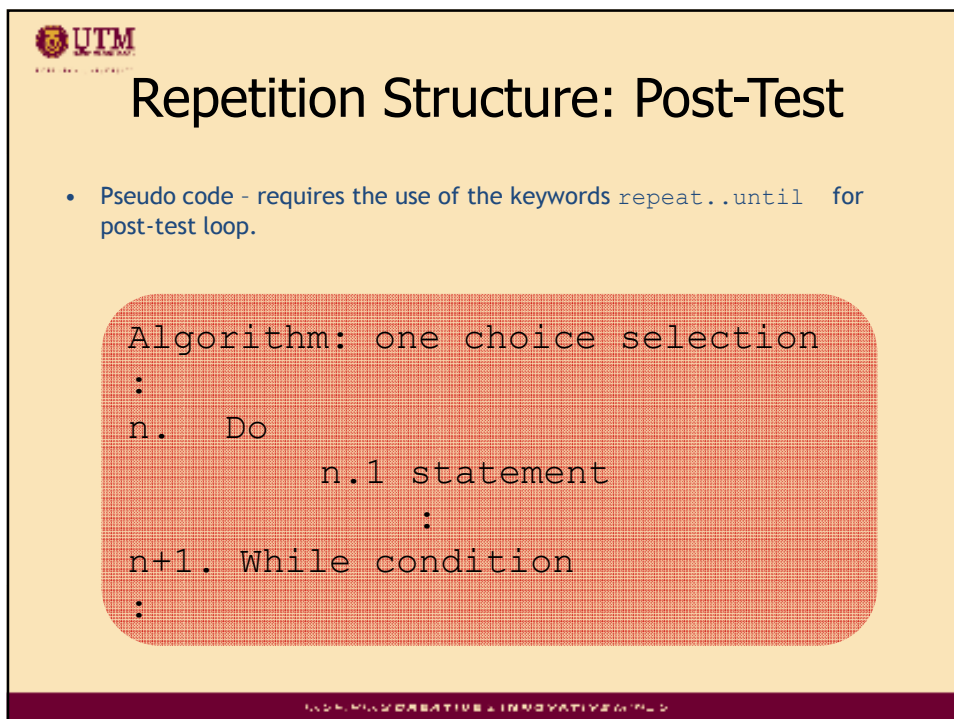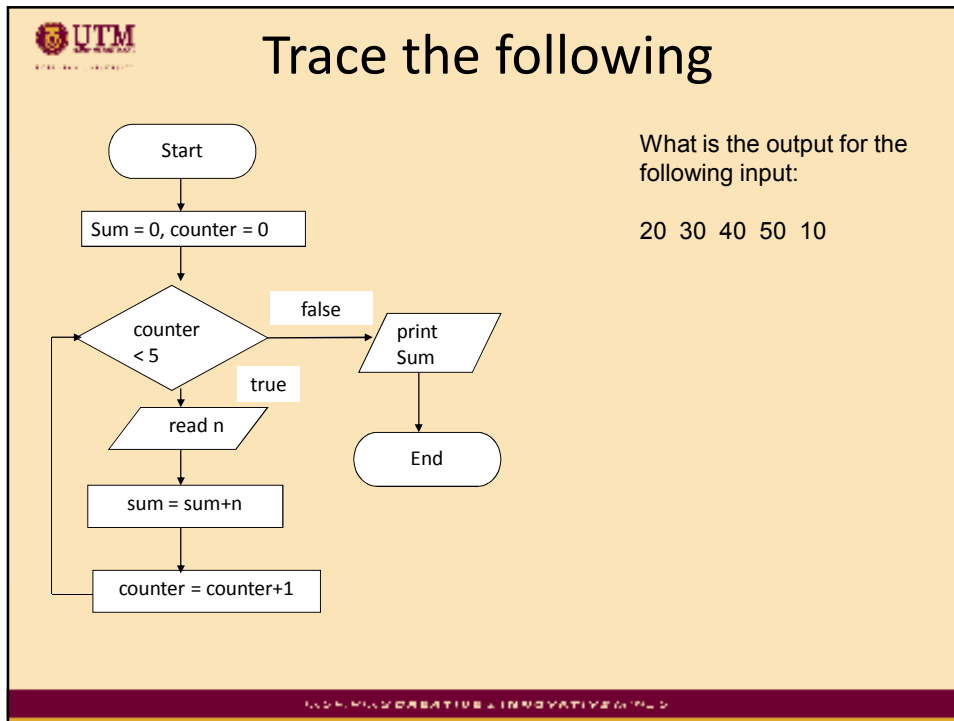   2. Counter=counter+1
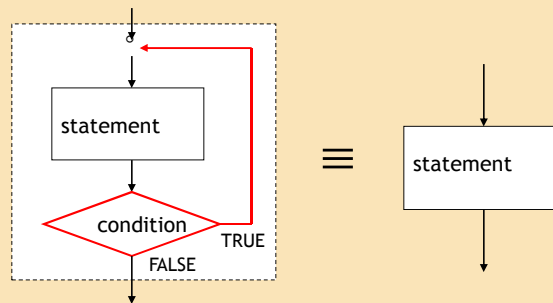4. End_While
5. Display sum
6. End

Flow Chart

Trace the following

Start

Sum = 0, counter = 0

counter < 5 — false → print Sum

true

read n

sum = sum+n

counter = counter+1

End

What is the output for the following input:

20 30 40 50 10

Repetition Structure: Post-Test

- Pseudo code – requires the use of the keywords `repeat..until` for post-test loop.

```
Algorithm: one choice selection
:
n.   Do
          n.1 statement
              :
n+1. While condition
:
```

Repetition Structure (*cont..*)

do-while Loop
(post-test loop)

Do the statement (body of loop) while a condition is true

Do-While loop

- The loop body is executed at least once.

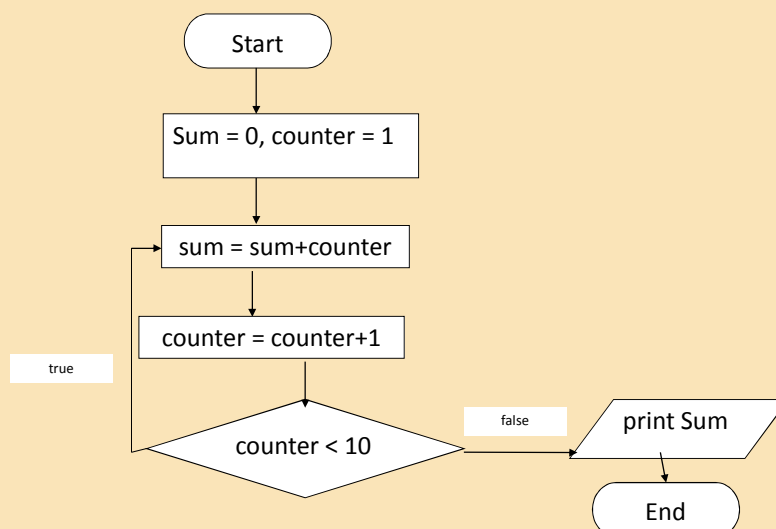# Example

1. Start
2. Set sum=0, counter = 1
3. Do
    1. Sum= Sum+counter
    2. Counter=counter+1
4. While (counter <10)
5. Display sum
6. End

Flow Chart

In-Class Exercise

- Develop an algorithm (pseudo code) and flow chart for a program to calculate an average of 15 numbers input by the user. Use pre-test loop

- Modify your solution above by using the post-test loop.

In-Class Exercise

- Develop an algorithm and flow chart to print even numbers between 1 to 50. Use pre-test loop.

- Modify your solution by using post-test loop.

In-Class Exercise

- Lab 3, Exercise 2, No. 1 (pg. 37)
- Lab 3, Exercise 2, No. 2 (pg. 38)