IF 130 Programming Fundamentals

04 – Pseudocode of Loop Control Structure

Dr. Maria Irmina Prasetiyowati, S.Kom,. M.T.
Alethea Suryadibrata, S.Kom., M.Eng.
Putri Sanggabuana Setiawan, S.Kom, M.T.I.
Januar Wahjudi, S.Kom., M.Sc.
Drs Slamet Aji Pamungkas, M.Eng
Kursehi Falgenti S.Kom., M.Kom.

Course Learning Outcome:

Students are able to compile pseudocode with selection control structures, repetition control structures, and modularization control structures (C3).

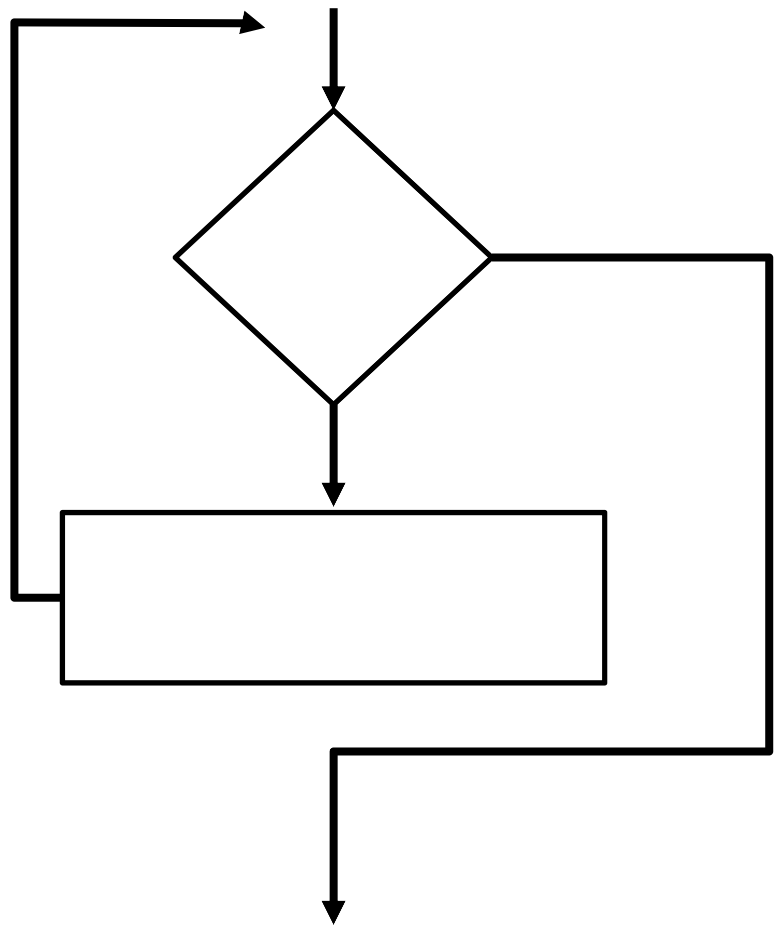
Review

- A loop is a **group of instructions** the computer executes repeatedly while some loop repetition condition remains true.
- 2 kinds of repetition
  1. Sentinel-controlled repetition
  2. Counter-controlled repetition

Outline

1. Pseudocode of repetition control structure
2. Desk checking
3. Exercises

DEFINITION & KIND OF REPETITION

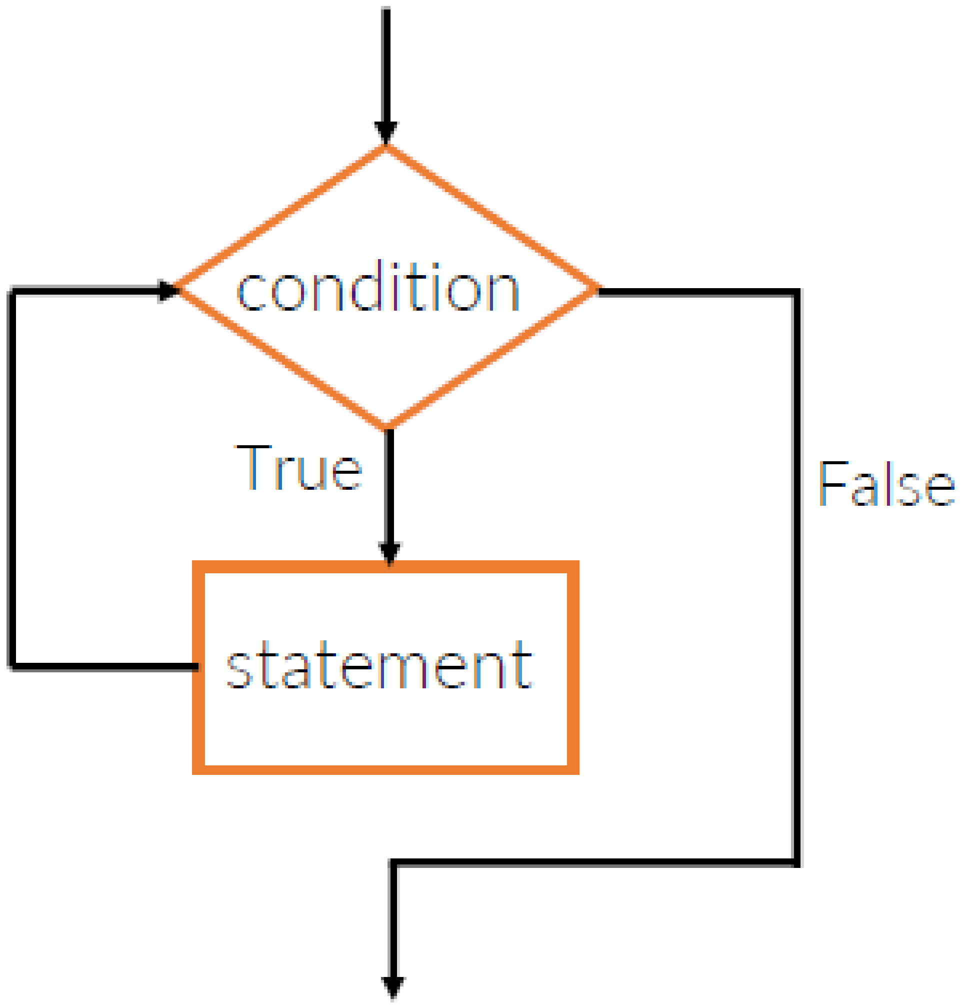


Sentinel-Controlled Repetition

- **While**
- **Do-While**
- **Do-Until**
- Both the **While** and **Do-While** loops cause a statement or set of statements to repeat **as long as** a condition is **true**.
- The **Do-Until** loop causes a statement or set of statements to repeat **until** a condition is **true**.

WHILE Loop

- “While a condition is true, do some task.”
- The loop is repeated when the condition is true (when its value is not 0).
- The loop is exited when the condition is false.

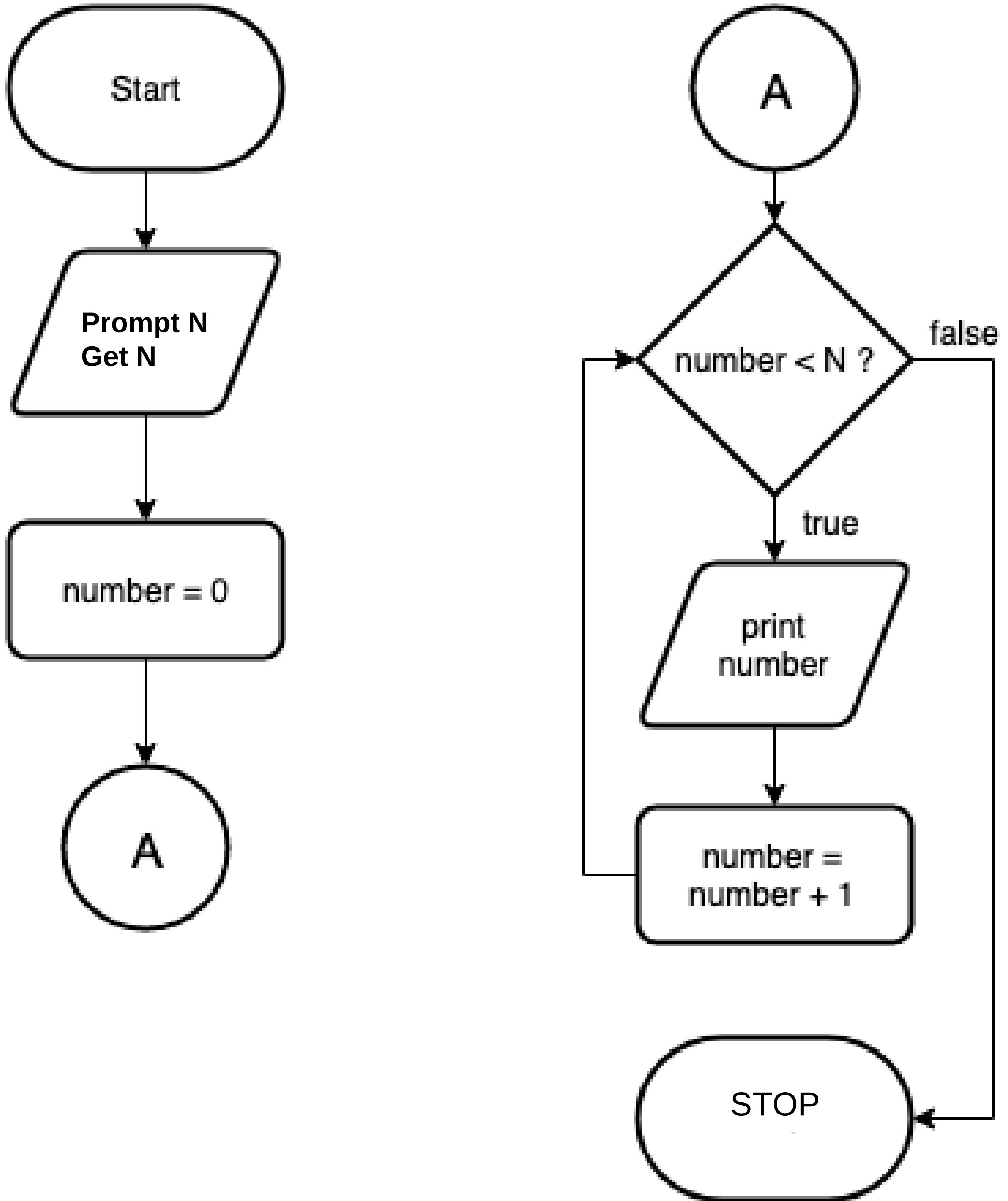


WHILE Loop

Program output
(with Input Shown in Bold)

```
10 [enter]
0  1  2  3  4  5  6  7  8  9
```

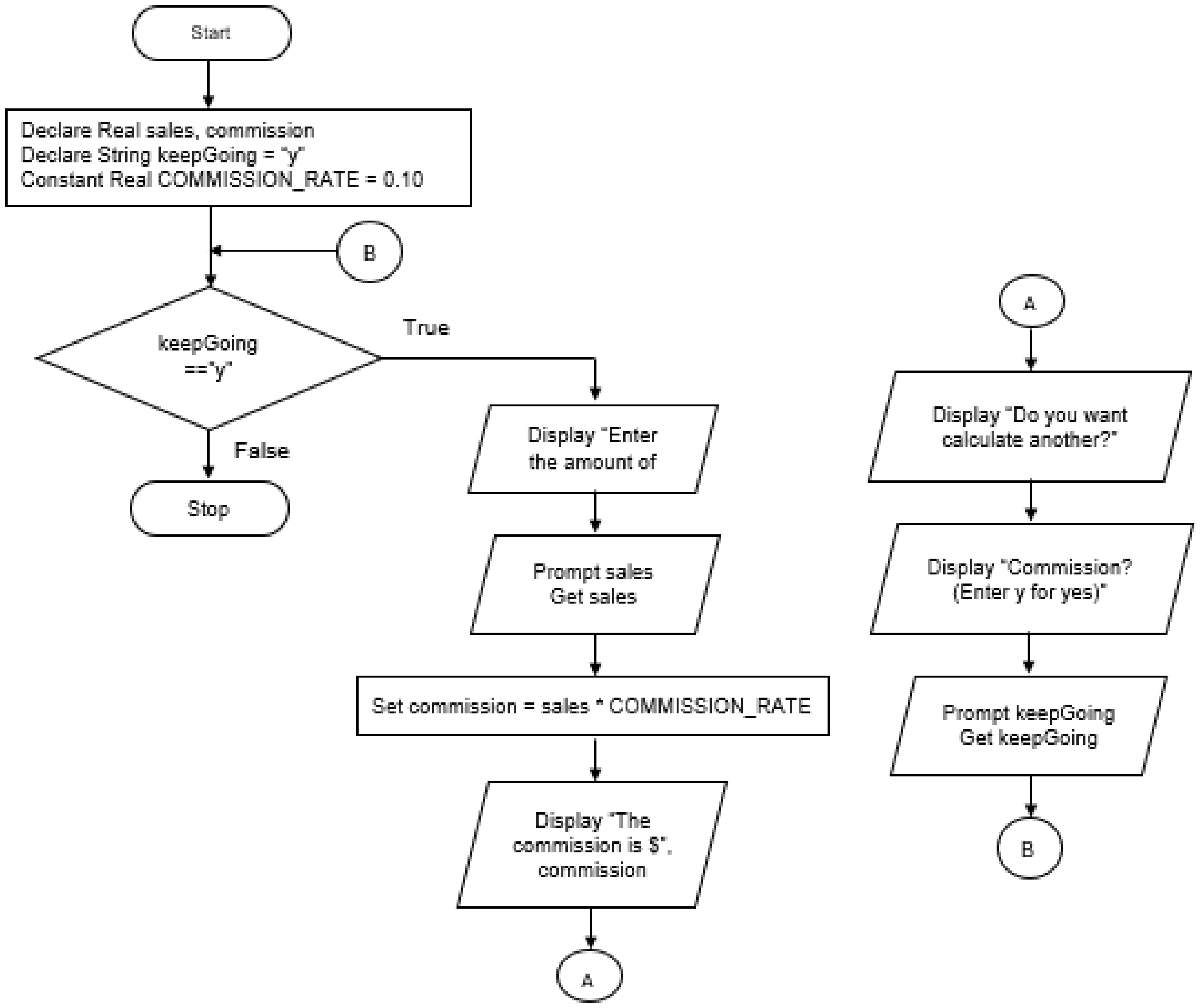
```
Display_N_numbers
  Prompt for N
  Get N
  Set number to 0
  WHILE number < N
    Print number
    number = number + 1
  ENDWHILE
END
```



WHILE Loop

Program Output (with Input Shown in Bold)

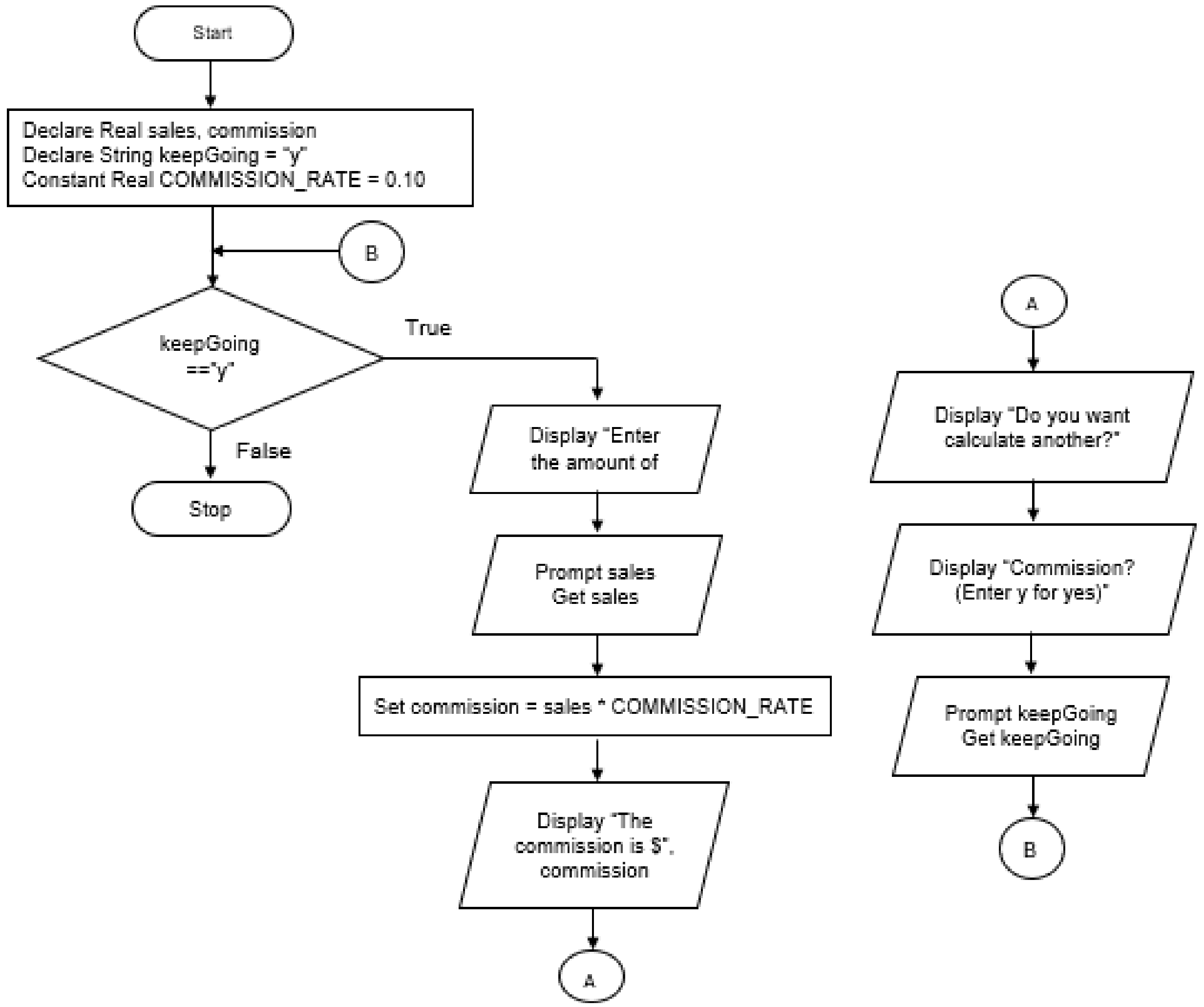
```
Enter the amount of sales.
10000.00 [Enter]
The commission is $1000
Do you want to calculate another
commission? (Enter y for yes.)
y [Enter]
Enter the amount of sales.
5000.00 [Enter]
The commission is $500
Do you want to calculate another
commission? (Enter y for yes.)
y [Enter]
Enter the amount of sales.
12000.00 [Enter]
The commission is $1200
Do you want to calculate another
commission? (Enter y for yes.)
n [Enter]
```



Start
Declare Real sales, commission
Declare String keepGoing = "y"
Constant Real COMMISSION_RATE = 0.10
B
keepGoing =="y"
True
False
Stop
Display "Enter the amount of
Prompt sales Get sales
Set commission = sales * COMMISSION_RATE
Display "The commission is $", commission
A
A
Display "Do you want calculate another?"
Display "Commission? (Enter y for yes)"
Prompt keepGoing Get keepGoing
B

WHILE Loop

```
Commission_calculation
  Declare Real sales, commission
  Declare String keepGoing = “y”
  Declare Constant Real COMMISSION_RATE = 0.10
  WHILE keepGoing == “y”
    Display “Enter the amount of sales.”
    Prompt sales
    Get sales
    Commission = sales * COMMISSION_RATE
    Display “The commission is $”, commission
    Display “Do you want to calculate another”
    Display “commission? (Enter y for yes)”
    Prompt keepGoing
    Get keepGoing
  ENDWHILE
END
```


Start
Declare Real sales, commission
Declare String keepGoing = "y"
Constant Real COMMISSION_RATE = 0.10
B
keepGoing =="y"
True
False
Stop
Display "Enter the amount of
Prompt sales
Get sales
Set commission = sales * COMMISSION_RATE
Display "The commission is $", commission
A
A
Display "Do you want calculate another?"
Display "Commission? (Enter y for yes)"
Prompt keepGoing
Get keepGoing
B

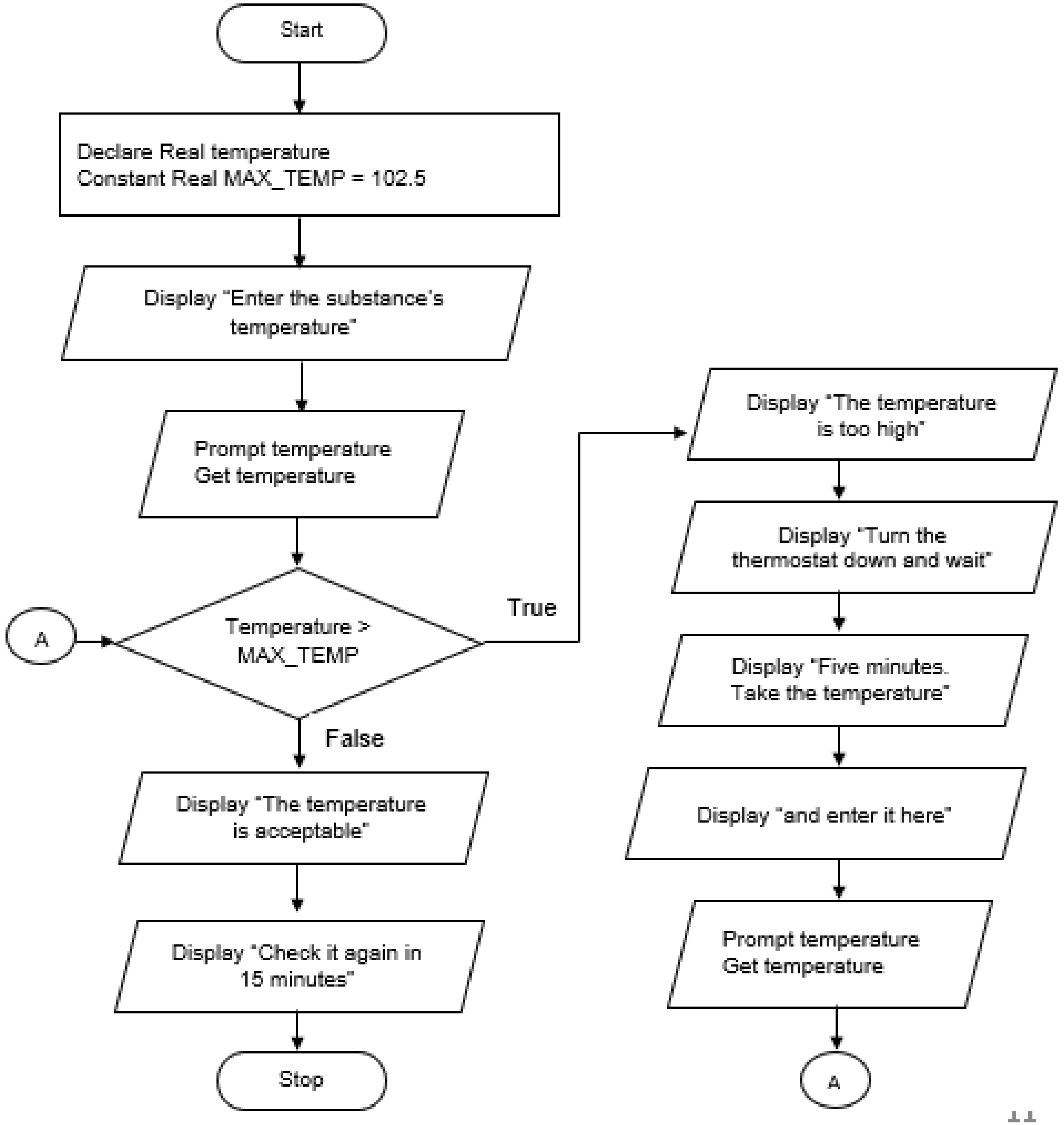
WHILE Loop

Program Output (with Input Shown in Bold)

```
Enter the substance's temperature.
104.7 [Enter]
The temperature is too high.
Turn the thermostat down and wait
five minutes. Take the temperature
again and enter it here.
103.2 [Enter]
The temperature is too high.
Turn the thermostat down and wait
five minutes. Take the temperature
again and enter it here.
102.1 [Enter]
The temperature is acceptable.
Check it again in 15 minutes.
```

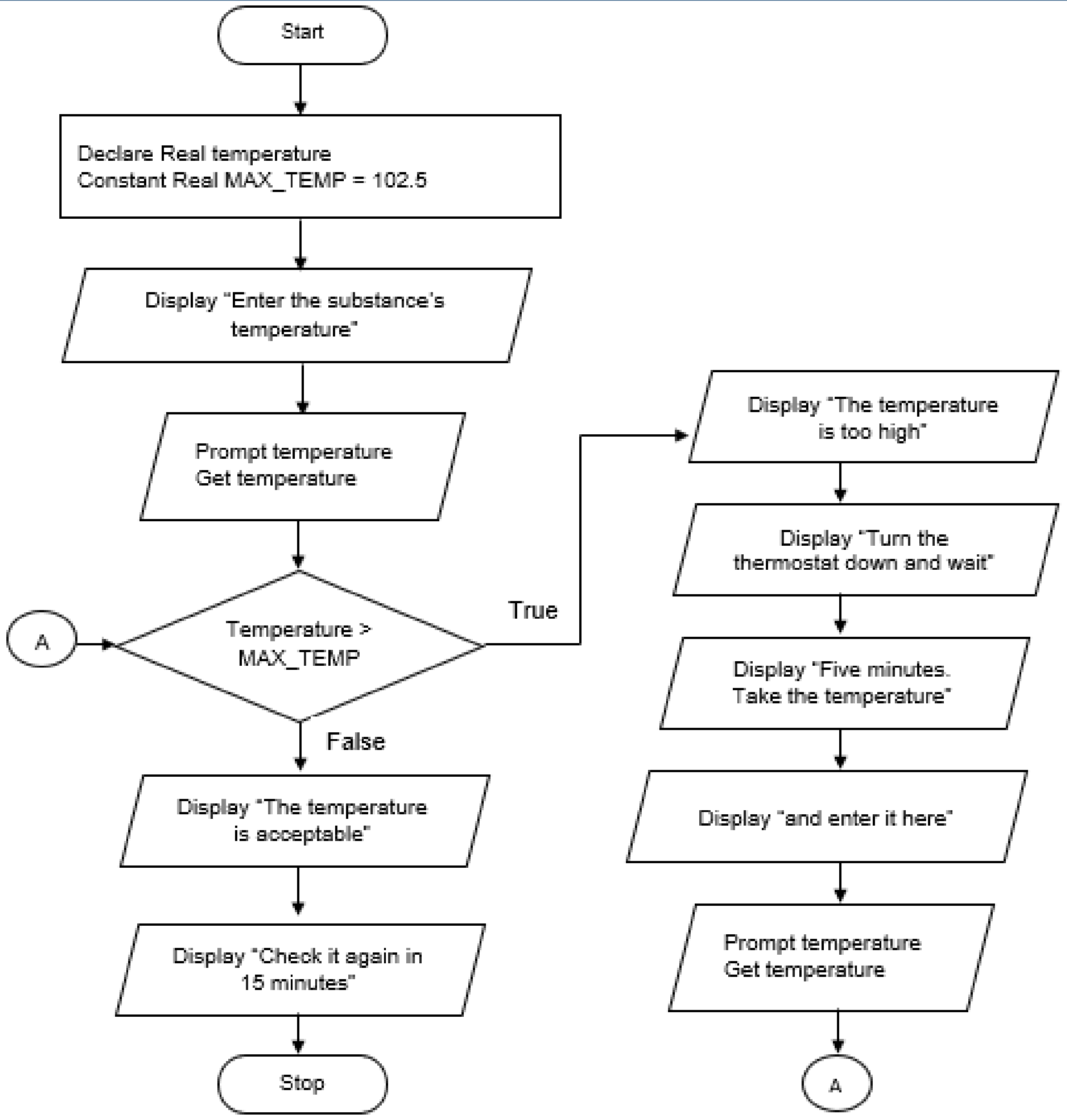
Program Output (with Input Shown in Bold)

```
Enter the substance's temperature.
102.1 [Enter]
The temperature is acceptable.
Check it again in 15 minutes.
```


Start
Declare Real temperature
Constant Real MAX_TEMP = 102.5
Display "Enter the substance's temperature"
Prompt temperature
Get temperature
A
Temperature > MAX_TEMP
True
False
Display "The temperature is acceptable"
Display "Check it again in 15 minutes"
Stop
Display "The temperature is too high"
Display "Turn the thermostat down and wait"
Display "Five minutes. Take the temperature"
Display "and enter it here"
Prompt temperature
Get temperature
A

WHILE Loop

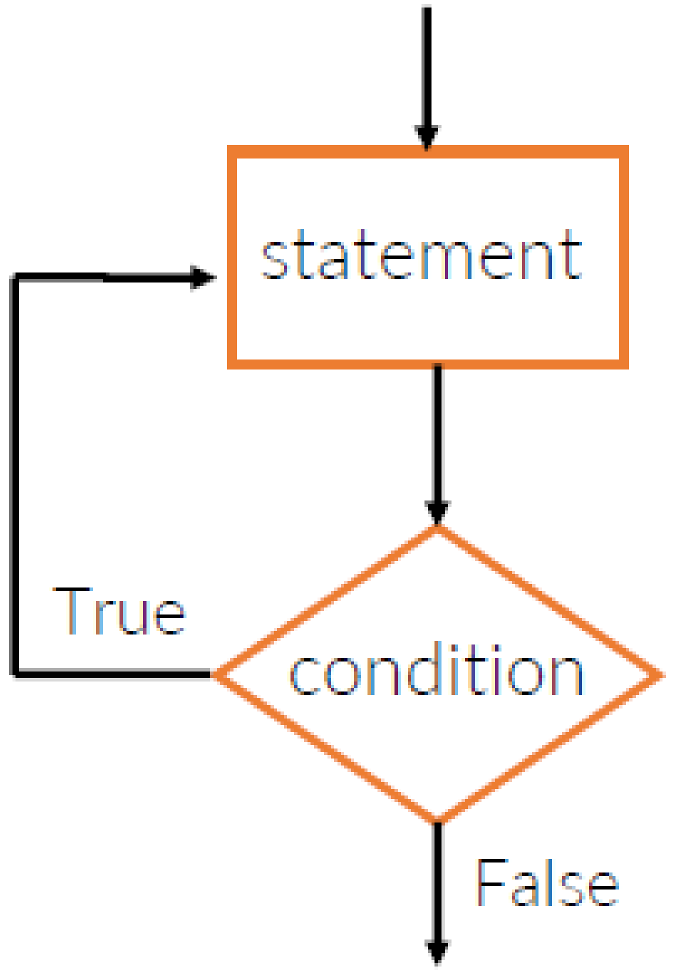
```
Check_temperature
  Declare Real temperature
  Declare Constant Real MAX_TEMP = 102.5
  Display “Enter the substance’s temperature”
  Prompt temperature
  Get temperature
  WHILE temperature > MAX_TEMP
    Display “The temperature is too high”
    Display “Turn the thermostat down and wait “
    Display “five minutes. Take the temperature “
    Display “and enter it here”
    Prompt temperature
    Get temperature
  ENDWHILE
  Display “The temperature is acceptable”
  Display “Check it again in 15 minutes”
END
```



Start
Declare Real temperature
Constant Real MAX_TEMP = 102.5
Display "Enter the substance's temperature"
Prompt temperature
Get temperature
A
Temperature > MAX_TEMP
True
False
Display "The temperature is acceptable"
Display "Check it again in 15 minutes"
Stop
Display "The temperature is too high"
Display "Turn the thermostat down and wait"
Display "Five minutes. Take the temperature"
Display "and enter it here"
Prompt temperature
Get temperature
A

DO-WHILE Loop

- The **Do-While** loop is a **posttest** loop. This means it performs an iteration before testing its condition.
- As a result, the Do-While loop **always performs at least one iteration**, even if its condition is false to begin with.

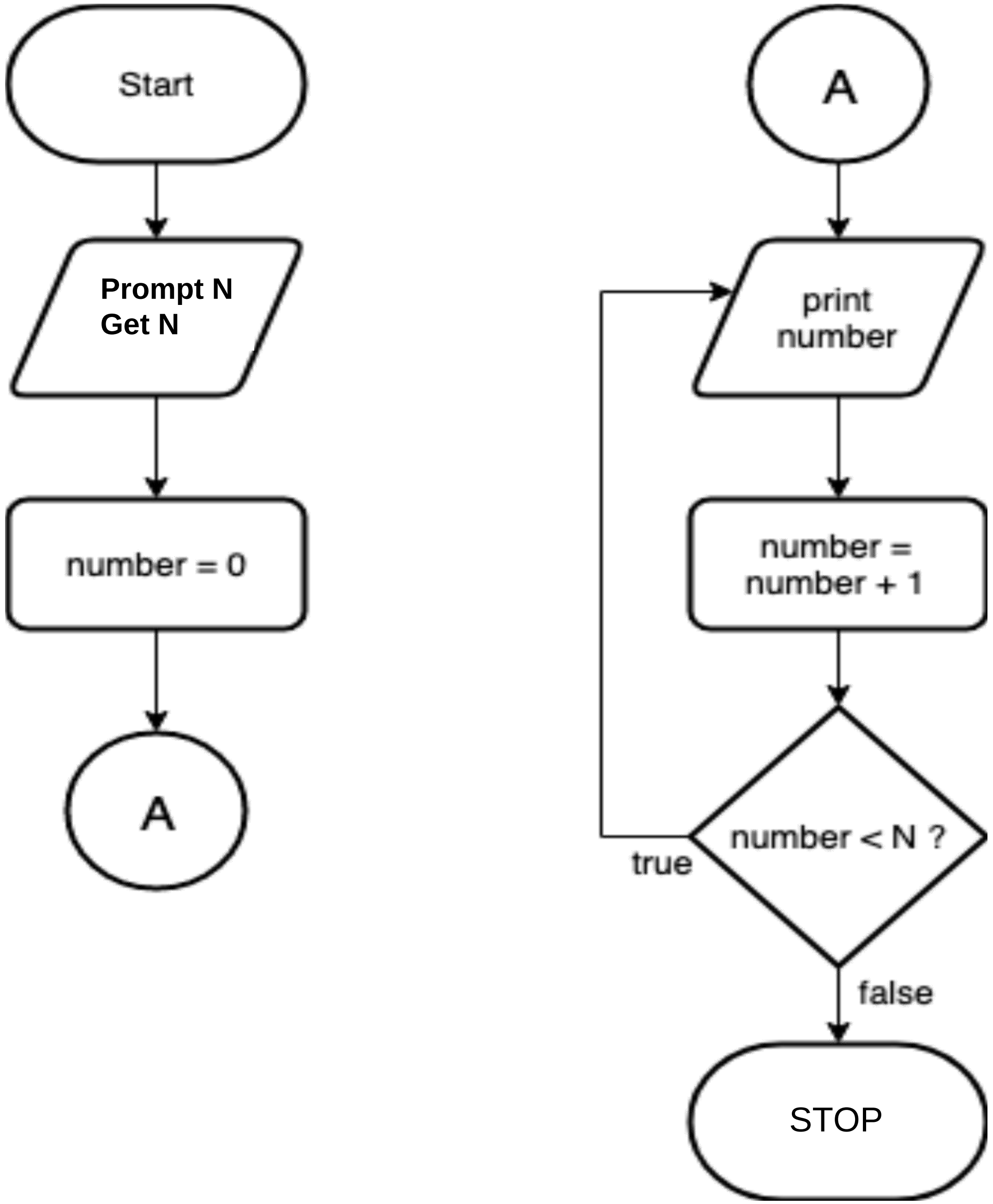


DO-WHILE Loop

Program output
(with Input Shown in Bold)

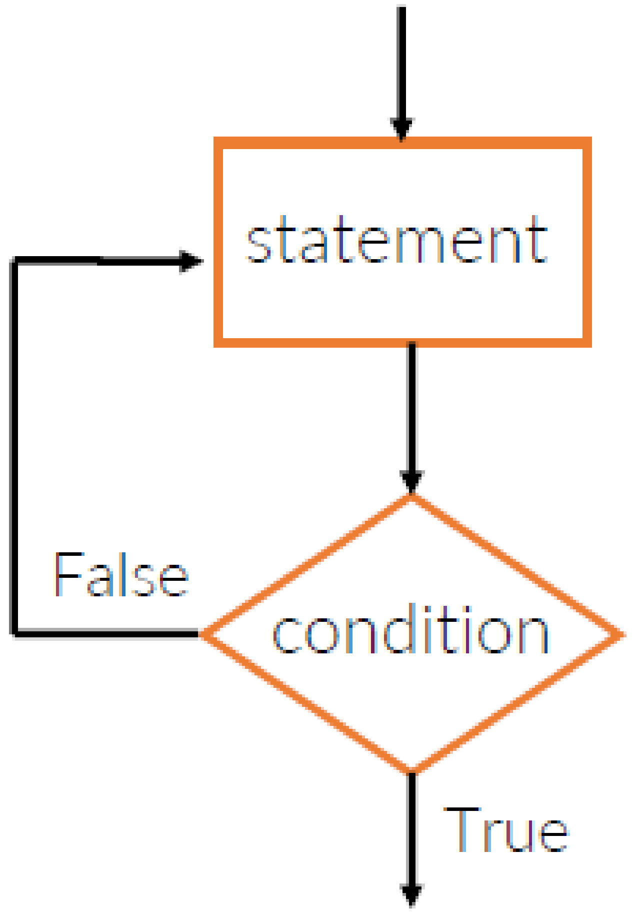
```
10 [enter]
0 1 2 3 4 5 6 7 8 9
```

```
Display_N_numbers
  Prompt for N
  Get N
  Set number to 0
  DO
    Print number
    number = number + 1
  WHILE number < N
END
```



DO-UNTIL Loop

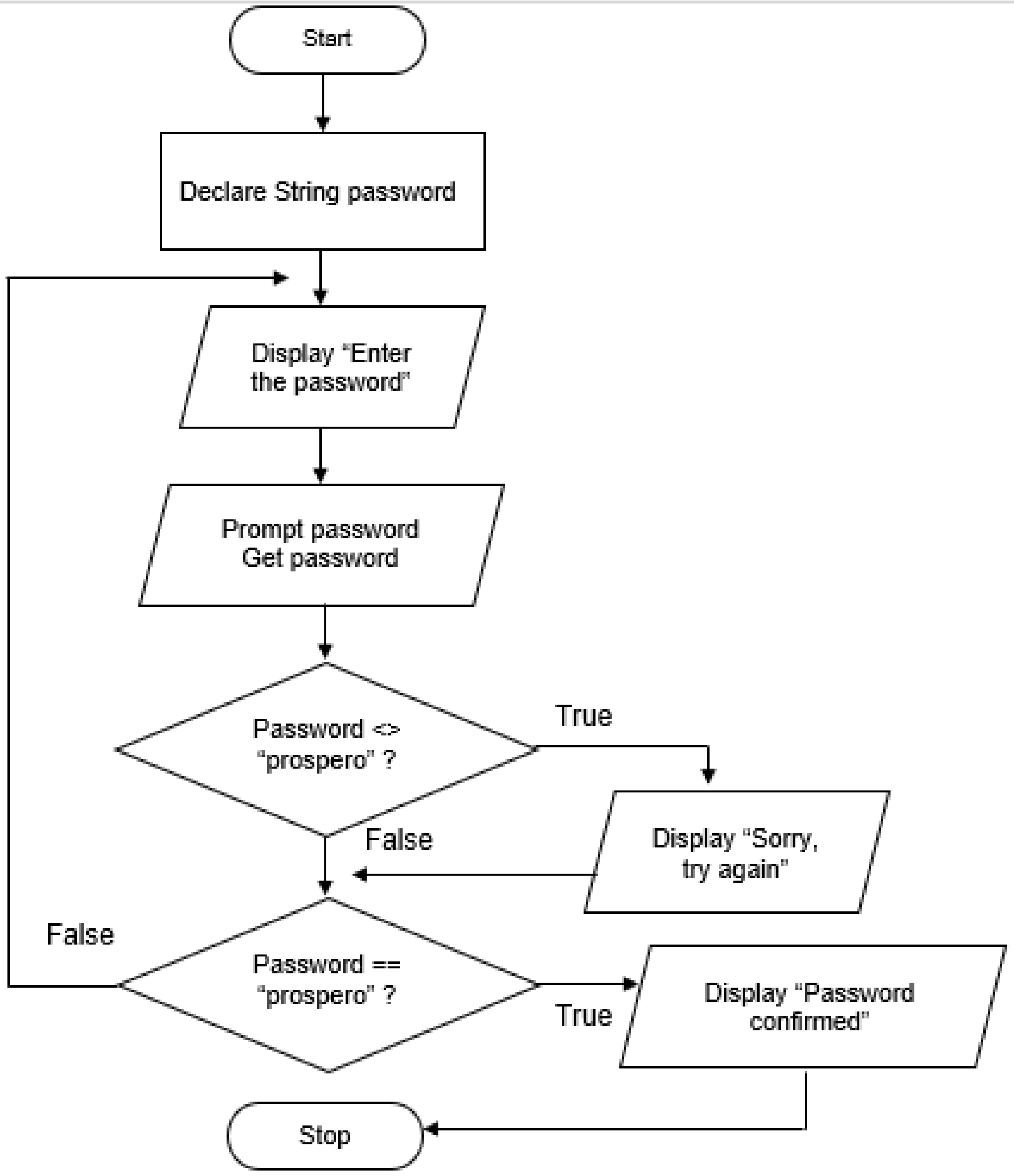
- Sometimes, however, it is more convenient to write a loop that iterates **until** a condition is **true**—that is, a loop that iterates as long as a condition is false, and then stops when the condition becomes true .



DO-UNTIL Loop

Program Output (with Input Shown in Bold)

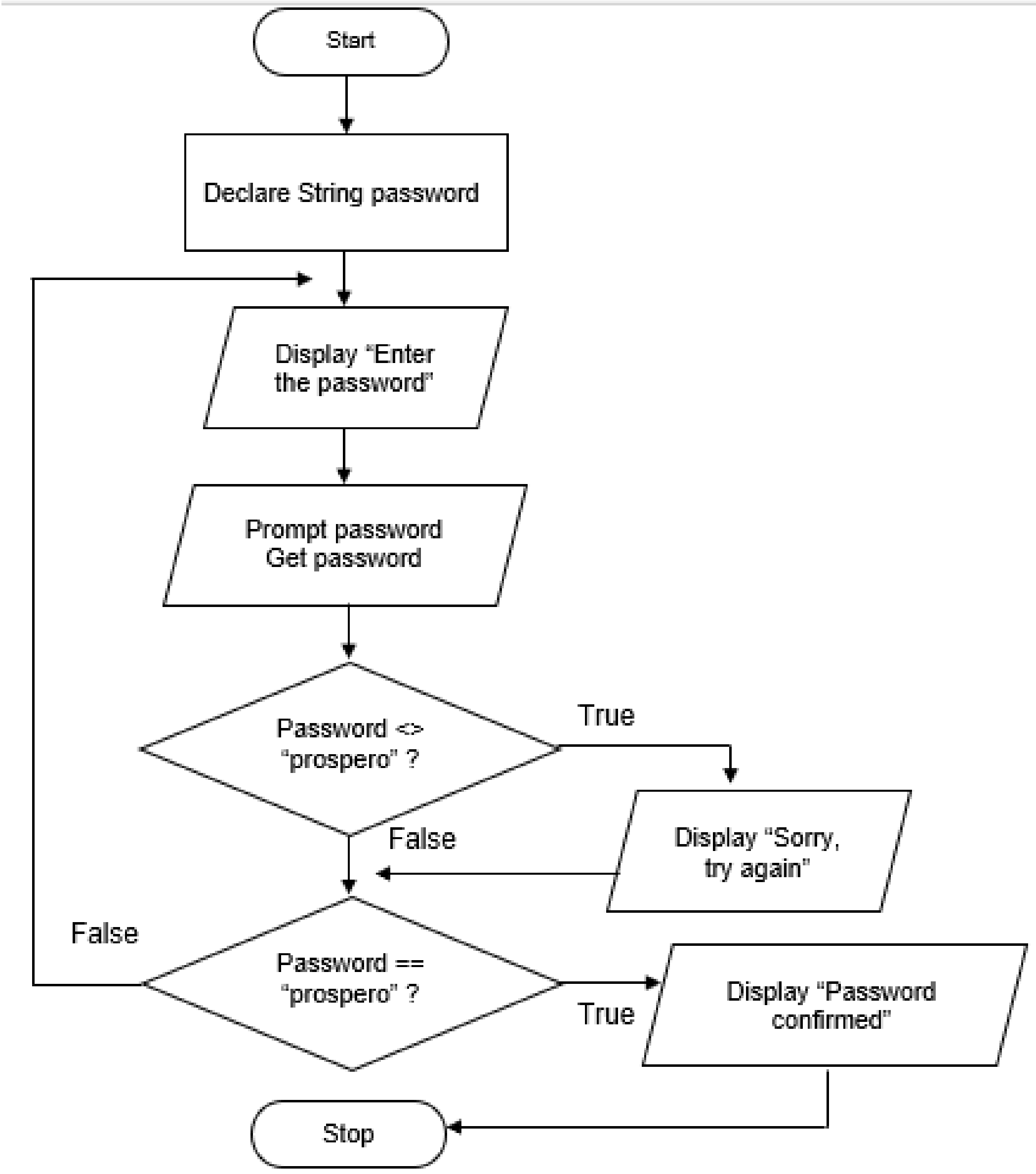
```
Enter the password.
**ariel [Enter]**
Sorry, try again.
Enter the password.
**caliban [Enter]**
Sorry, try again.
Enter the password.
**prospero [Enter]**
Password confirmed.
```



Start
Declare String password
Display "Enter the password"
Prompt password Get password
Password <> "prospero" ?
True
Display "Sorry, try again"
False
False
Password == "prospero" ?
True
Display "Password confirmed"
Stop

DO-UNTIL Loop

```
Check_password
  Declare string password
  DO
    Display ”Enter the password.”
    Prompt password
    Get password
    IF password != “prospero” THEN
      Display “Sorry, try again.
    ENDIF
  UNTIL password == “prospero”
  Display “Password confimed.”
END
```

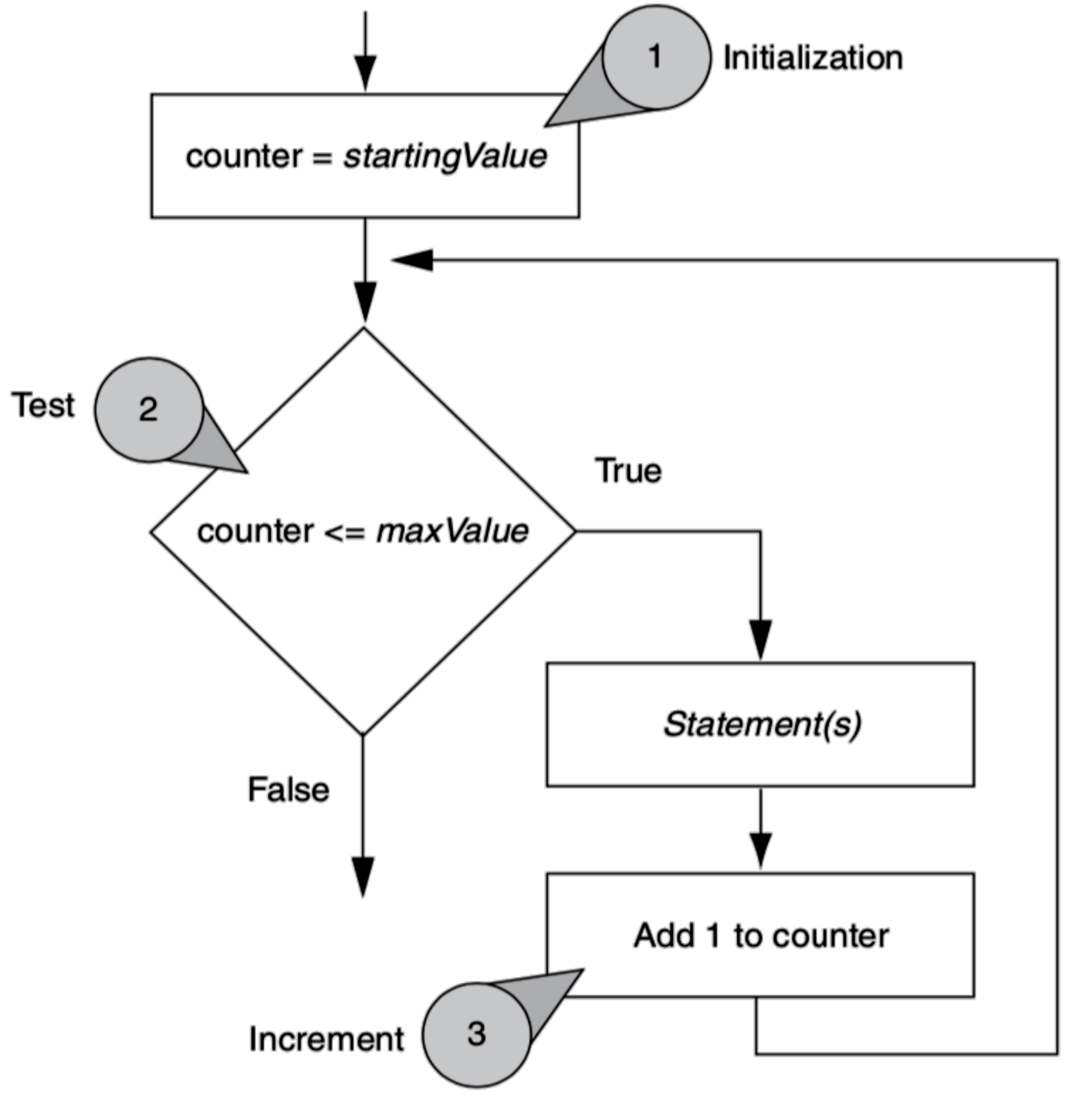


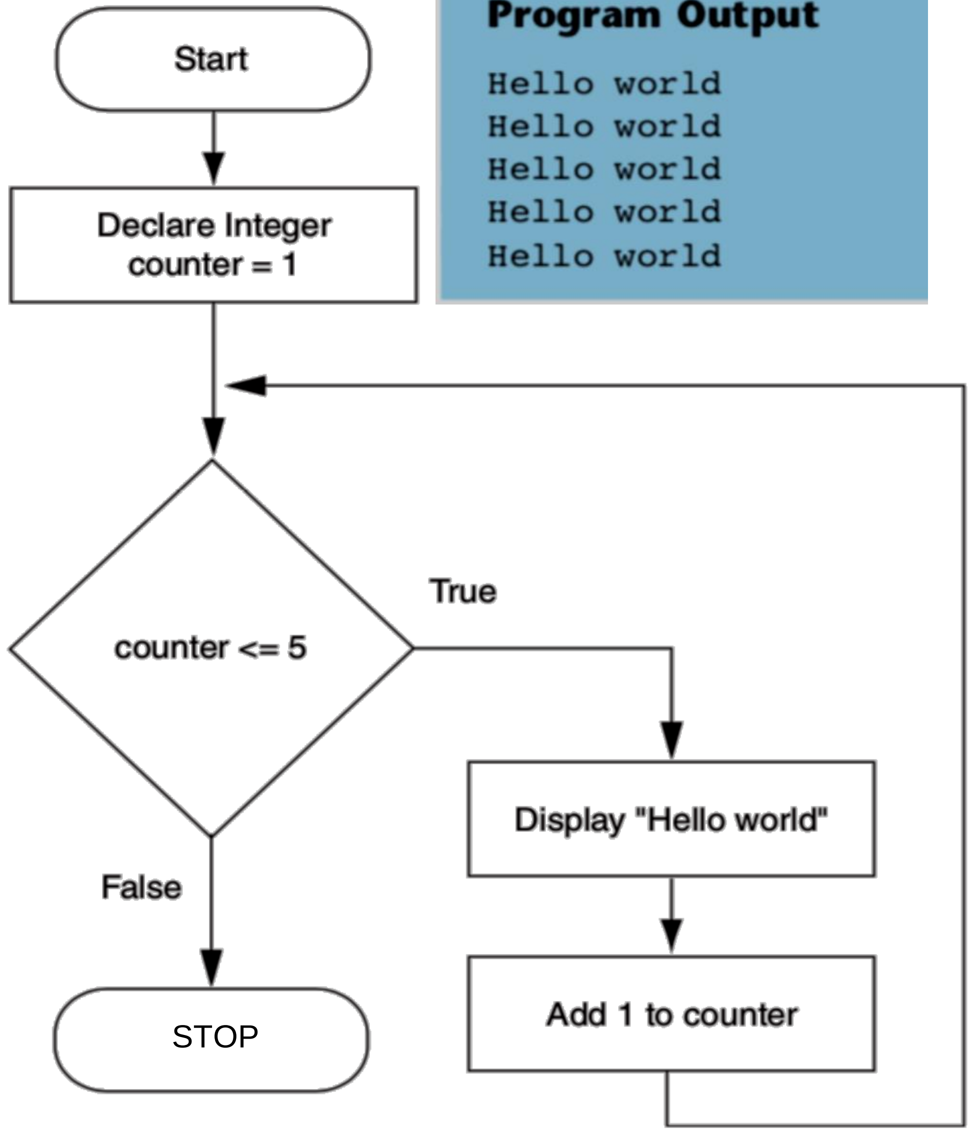
Start
Declare String password
Display "Enter the password"
Prompt password Get password
Password <> "prospero" ?
True
False
Display "Sorry, try again"
Password == "prospero" ?
False
True
Display "Password confirmed"
Stop

Counter-Controlled Repetition

- Counter-controlled repetition is sometimes called **definite repetition** because we know in advance exactly how many times the loop will be executed.
- This is usually called the **For statement.**
- A loop control variable is used to count the number of repetitions
  1. **Initialization:** Loop control variable is set to an initial value before the while statement is reached
  2. **Testing:** Loop control variable is tested before the start of each loop repetition
  3. **Updating/Increment:** Loop control variable is updated (incremented / decremented) during each iteration

Counter-Controlled Repetition



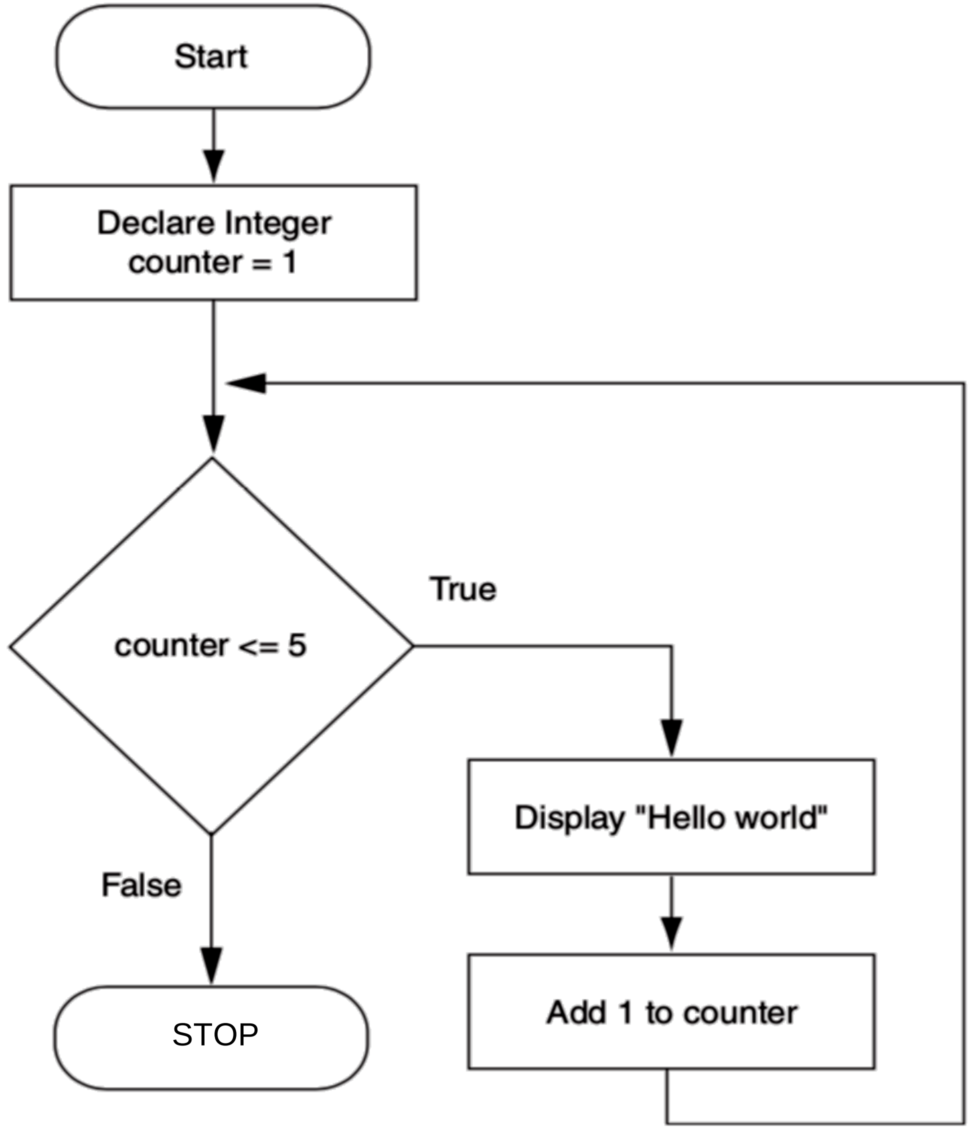


Program Output

```
Hello world
Hello world
Hello world
Hello world
Hello world
```

Counter-Controlled Repetition

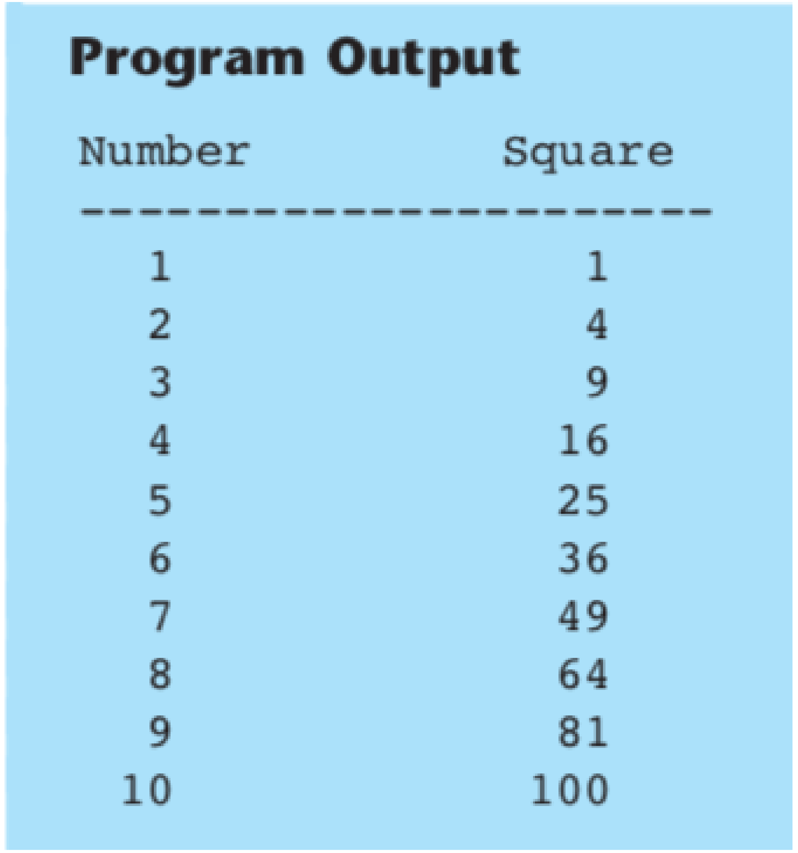
```
Counting_integers
  Declare integer counter = 1
  FOR counter = 1 TO 5
    Display "Hello world"
    counter = counter+1
  ENDFOR
END
```



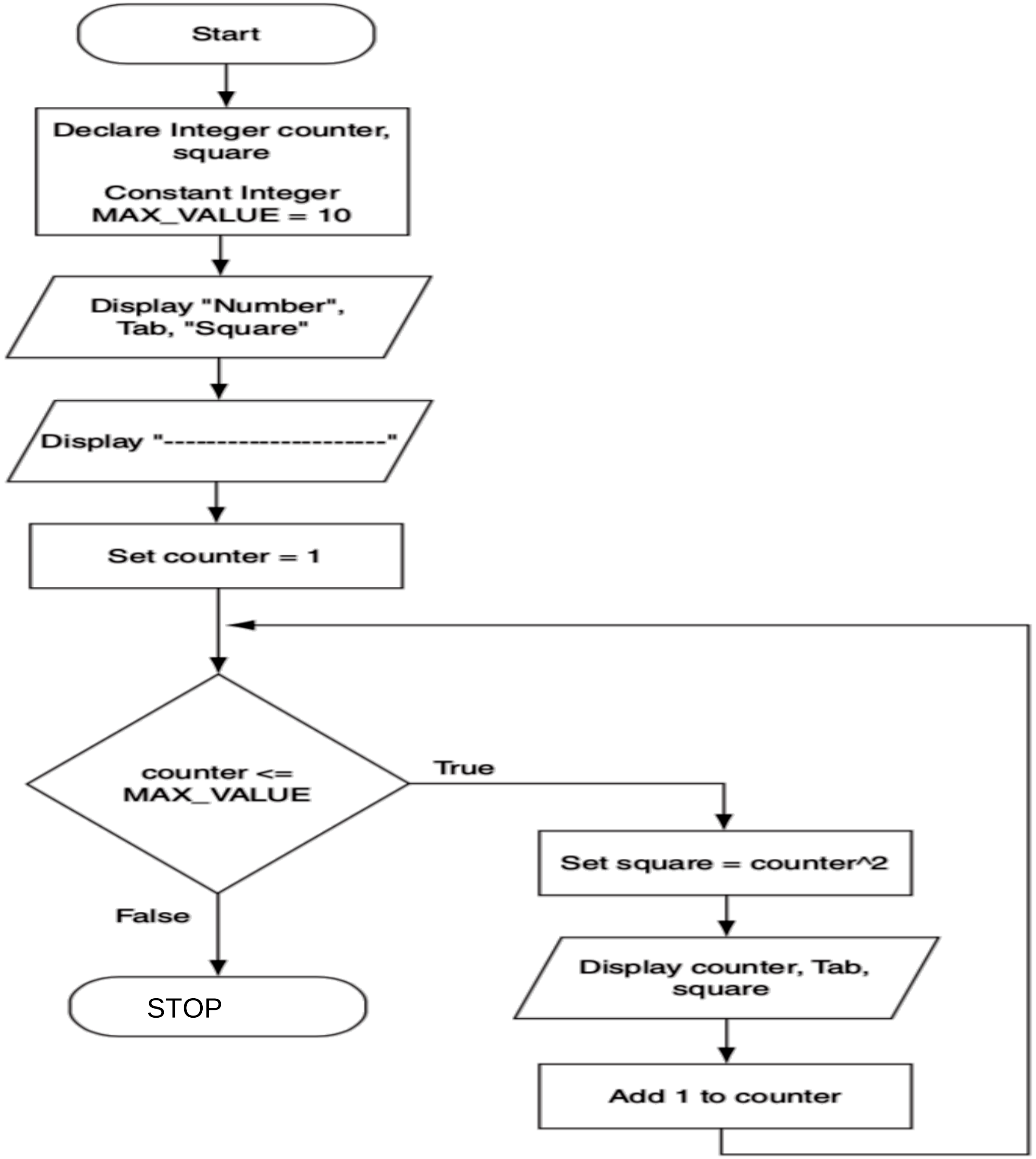
FOR Statement

- In some situations, it is also helpful to use the counter variable in a calculation or other task within the body of the loop.
- For example, suppose you need to write a program that displays the numbers 1 through 10 and their squares.

FOR Statement



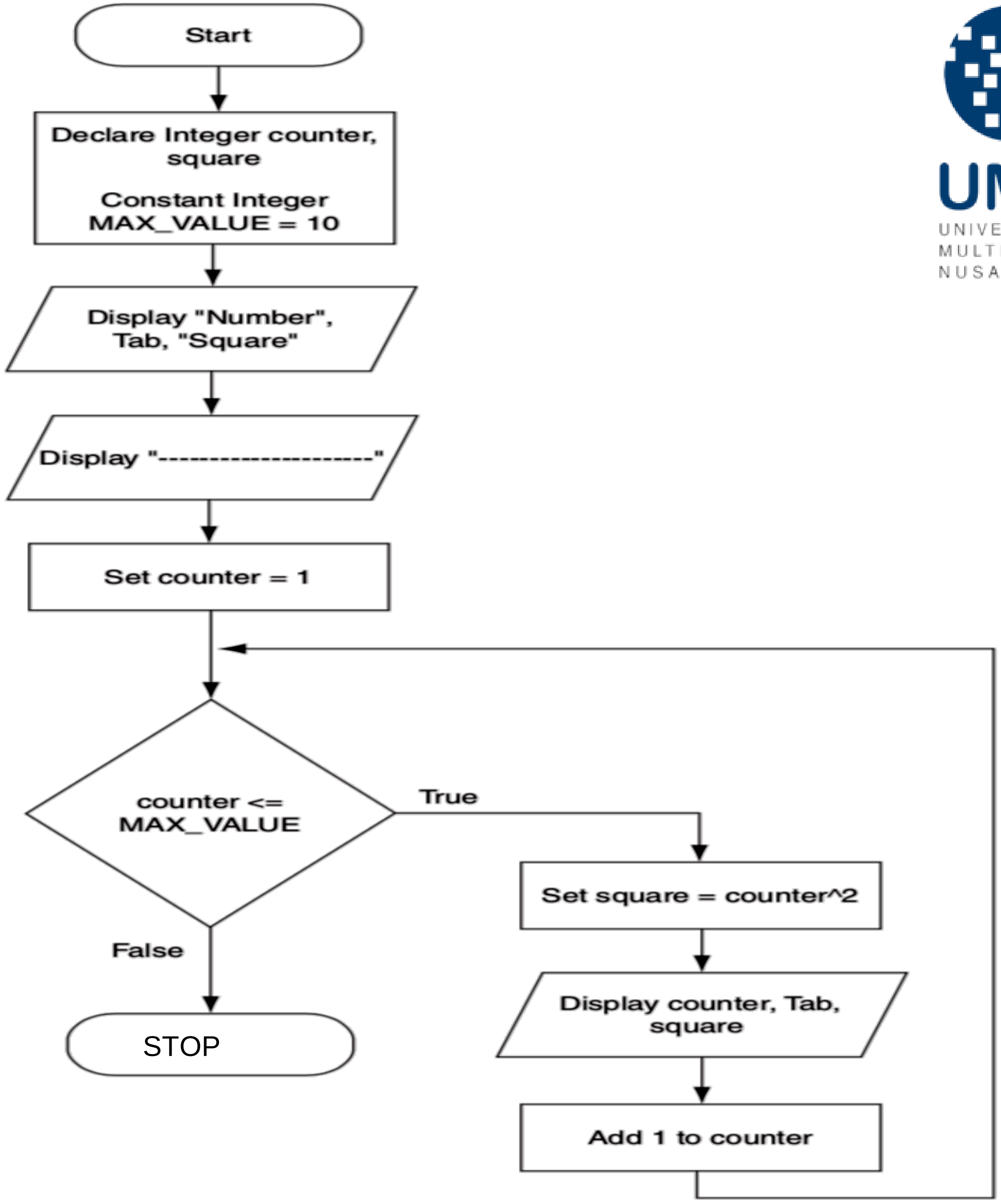
```
Program Output

Number              Square
----------------------------
   1                    1
   2                    4
   3                    9
   4                   16
   5                   25
   6                   36
   7                   49
   8                   64
   9                   81
  10                  100
```



Start
Declare Integer counter, square
Constant Integer MAX_VALUE = 10
Display "Number", Tab, "Square"
Display "----------------------"
Set counter = 1
counter <= MAX_VALUE
True
False
STOP
Set square = counter^2
Display counter, Tab, square
Add 1 to counter

FOR Statement

```
Square_integers
  Declare integer counter , square
  Declare Constant Integer MAX_VALUE = 10
  Display “Number”, Tab, “Square”
  Display “--------------”
  Set counter = 1
  FOR counter = 1 TO MAX_VALUE
    Set square to counter*counter
    Display counter, Tab, square
    counter = counter+1
  ENDFOR
END
```


Start
Declare Integer counter, square
Constant Integer MAX_VALUE = 10
Display "Number", Tab, "Square"
Display "--------------------"
Set counter = 1
counter <= MAX_VALUE
True
False
Set square = counter^2
Display counter, Tab, square
Add 1 to counter
STOP

Practice 1

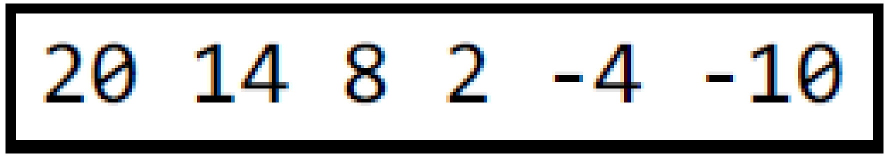
- Design an algorithm in **pseudocode** which displays the numbers 1 through the maximum value (user input) and their squares.
- Maximum value = 5

Number	Square
1	1
2	4
3	9
4	16
5	25

```
Square_integers
  Declare integer counter , square
  Display “Number”, Tab, “Square”
  Display “--------------”
  Set counter = 1
  Prompt Max_Value
  Get Max_Value
  FOR counter = 1 TO Max_Value
    Set square to counter*counter
    Display counter, Tab, square
    counter = counter+1
  ENDFOR
END
```

Practice 2

- Design an algorithm **in pseudocode** that print the following sequence of values

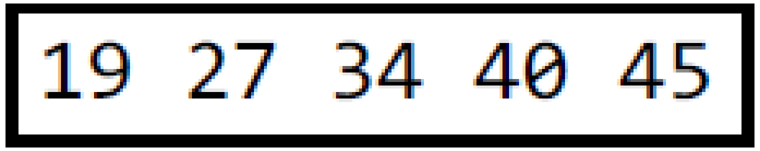


```
20 14 8 2 -4 -10
```

```
Sequence_Value
  Declare integer i, angka
  Set angka = 20
  FOR i= 1 TO 6
    Display angka
    angka = angka - 6
    i = i + 1
  ENDFOR
END
```

Practice 3

- Design an algorithm **in pseudocode** that print the following sequence of values

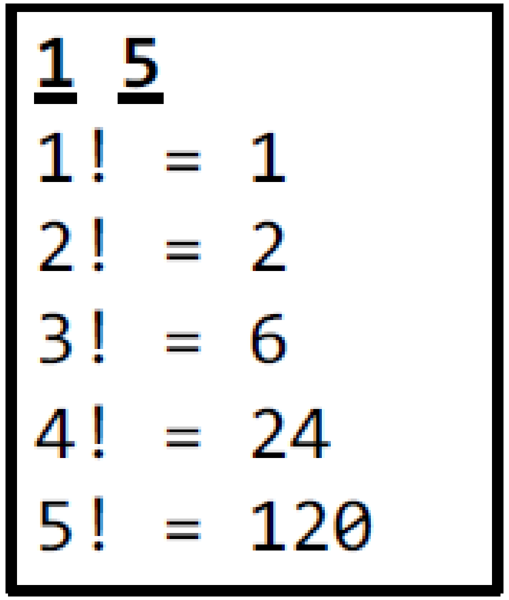


```
19 27 34 40 45
```

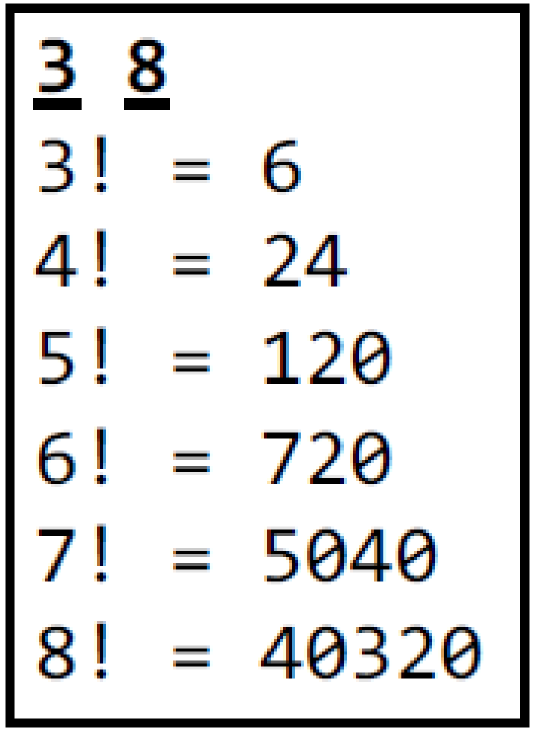
```
Sequence_Value
  Declare integer i, angka
  Set angka = 19
  FOR i = 0 TO 4
    Display angka
    angka = angka + (8 – i)
    i = i + 1
  ENDFOR
END
```

Practice 4

- The factorial function is used frequently in probability problems. The factorial of a positive integer n (written n! and pronounced “n factorial”) is equal to the product of the positive integers from 1 to n. Write in a **pseudocode** that evaluates the factorials of the integers from p to q (p and q are inputted by user). The screen dialogue should appear as follows:



```
1 5
1!  = 1
2!  = 2
3!  = 6
4!  = 24
5!  = 120
```



```
3 8
3!  = 6
4!  = 24
5!  = 120
6!  = 720
7!  = 5040
8!  = 40320
```

NEXT WEEK’S OUTLINE

1. Definition of modular programming
2. Modular flowchart
3. Modular Desk checking
4. Exercises

REFERENCES

1. Gaddis, Tony, 2019, Starting out with programming logic & design, Fifth edition, Pearson Education, Inc.
2. Robertson, Lesley Anne, 2007, Simple Program Design A Step-by-Step Approach, Fith Edition, Thomson Learning, Inc.
3. Informatics study program slides, 2023, Fundamentals of Programming, Universitas Multimedia Nusantara.

Visi

Menjadi Program Studi Strata Satu Informatika **unggulan** yang menghasilkan lulusan **berwawasan internasional** yang **kompeten** di bidang Ilmu Komputer (*Computer Science*), **berjiwa wirausaha** dan **berbudi pekerti luhur**.

INFORMATIKA
UMN
UNIVERSITAS MULTIMEDIA NUSANTARA

Misi

1. Menyelenggarakan pembelajaran dengan teknologi dan kurikulum terbaik serta didukung tenaga pengajar profesional.
2. Melaksanakan kegiatan penelitian di bidang Informatika untuk memajukan ilmu dan teknologi Informatika.
3. Melaksanakan kegiatan pengabdian kepada masyarakat berbasis ilmu dan teknologi Informatika dalam rangka mengamalkan ilmu dan teknologi Informatika.