CSE111 Introduction to Computer Applications

Lecture 12

Introduction to Algorithms

Prepared by Asst. Prof. Dr. Mohamed KURDI

Revised and presented by Asst. Prof. Dr. Samsun M. BAŞARICI

Learning Objectives

- Understand what is meant by an algorithm.
- Understand what is meant by a variable.
- Understand the six basic statements in pseudocode.
- Using and applying pseudocode for writing algorithms.
- Understand the building units of a flowchart.
- Understand the three basic types of control structures: sequence, decision and repetition.
- Using and applying flowcharts for writing algorithms.

T.C. Adnan Menderes University Faculty of Engineering

Outlines

- Humans & Computer Algorithms
- Variables
- Pseudocode
  - Basic Pseudocode Statements
- Flowcharts
  - Flowchart Building Units
  - Control Structures in Flowcharts
    - The Sequence Structure.
    - The Decision Structure.
    - The Repetition Structure.

T.C. Adnan Menderes University Faculty of Engineering

Humans & Computer Algorithms

- An algorithm is a step-by-step method for solving a particular problem or doing a specific task.
  - Has input data, and is expected to produce output data.
  - Each step can be carried out in a finite amount of time in a deterministic way.

T.C. Adnan Menderes University Faculty of Engineering

Humans & Computer Algorithms

- An algorithm is a step-by-step method for solving a particular problem or doing a specific task.
  - Has input data, and is expected to produce output data.
  - Each step can be carried out in a finite amount of time in a deterministic way.
- Do humans use algorithms in their daily life?
  - Driving a car.
  - Cooking.
  - Passing a course in an engineering faculty.
  - Many more.





T.C. Adnan Menderes University Faculty of Engineering

Humans & Computer Algorithms

- An algorithm is a step-by-step method for solving a particular problem or doing a specific task.
  - Has input data, and is expected to produce output data.
  - Each step can be carried out in a finite amount of time in a deterministic way.
- Do humans use algorithms in their daily life?
  - Driving a car.
  - Cooking.
  - Passing a course in an engineering faculty.
  - Many more.
- Do Computers use algorithms in their mechanism?
  - Operating system: Windows, Linux, Android, etc.
  - Deep blue (PC) which overcame Kasparov in chess board.
  - Many more.











T.C. Adnan Menderes University Faculty of Engineering

Humans & Computer Algorithms

- Humans can run algorithms using the five senses, brain, and memory.

T.C. Adnan Menderes University Faculty of Engineering

Humans & Computer Algorithms

- Humans can run algorithms using the five senses, brain, and memory.
- Computers can also run algorithms using the input and output devices, CPU, and storage device if they were written in a formal programming language such as C, C++, C #, Pascal, Java, Matlab, Android, etc.

T.C. Adnan Menderes University Faculty of Engineering

Humans & Computer Algorithms

- Humans can run algorithms using the five senses, brain, and memory.
- Computers can also run algorithms using the input and output devices, CPU, and storage device if they were written in a formal programming language such as C, C++, C #, Pascal, Java, Matlab, Android, etc.
- Before we learn how to write algorithms using a formal programming language, we need to learn how to write algorithms in simpler forms such as flowcharts and pseudocode.

T.C. Adnan Menderes University Faculty of Engineering

Humans & Computer Algorithms

- Humans can run algorithms using the five senses, brain, and memory.
- Computers can also run algorithms using the input and output devices, CPU, and storage device if they were written in a formal programming language such as C, C++, C #, Pascal, Java, Matlab, Android, etc.
- Before we learn how to write algorithms using a formal programming language, we need to learn how to write algorithms in simpler forms such as flowcharts and pseudocode.
- Why?

T.C. Adnan Menderes University Faculty of Engineering

Variables

- When humans run an algorithm, data are stored in their memory, humans can save and retrieve the data automatically without specifying a location and a name for each piece of data.

T.C. Adnan Menderes University Faculty of Engineering

Variables

- When humans run an algorithm, data are stored in their memory, humans can save and retrieve the data automatically without specifying a location and a name for each piece of data.
- Computers do not have such abilities, and each piece of data should have a specific name and unique location (cell or set of cells).

T.C. Adnan Menderes University Faculty of Engineering

Variables

- When humans run an algorithm, data are stored in their memory, humans can save and retrieve the data automatically without specifying a location and a name for each piece of data.
- Computers do not have such abilities, and each piece of data should have a specific name and unique location (cell or set of cells).
- Don't worry much, you need only to provide a name to a piece of data, and the computer will allocate a unique location for it in the computer memory (RAM).

T.C. Adnan Menderes University Faculty of Engineering

Variables

- When humans run an algorithm, data are stored in their memory, humans can save and retrieve the data automatically without specifying a location and a name for each piece of data.
- Computers do not have such abilities, and each piece of data should have a specific name and unique location (cell or set of cells).
- Don't worry much, you need only to provide a name to a piece of data, and the computer will allocate a unique location for it in the computer memory (RAM).
- In terminology of algorithms, the word **variable** means a place in the computer memory that can hold a piece of data, this place has a specific name and unique physical address.

T.C. Adnan Menderes University Faculty of Engineering

Variables

- When humans run an algorithm, data are stored in their memory, humans can save and retrieve the data automatically without specifying a location and a name for each piece of data.
- Computers do not have such abilities, and each piece of data should have a specific name and unique location (cell or set of cells).
- Don't worry much, you need only to provide a name to a piece of data, and the computer will allocate a unique location for it in the computer memory (RAM).
- In terminology of algorithms, the word **variable** means a place in the computer memory that can hold a piece of data, this place has a specific name and unique physical address.
- A variable can be also seen as a container to store certain values. While the program is running, variables are accessed and sometimes changed, i.e. a new value will be assigned to the variable.



http://www.python-course.eu/variables.php

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode

- Pseudocode means fake code, because it is not a real programming code.
- It is a type of structured English that is used to specify an algorithm.

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode

- Pseudocode means fake code, because it is not a real programming code.
- It is a type of structured English that is used to specify an algorithm.
- It cannot be compiled nor executed on computers, and has no real formatting or syntax rules.
- It enables the programmer to concentrate on the algorithms without worrying about all the syntactic details of a particular programming language

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode

- Pseudocode means fake code, because it is not a real programming code.
- It is a type of structured English that is used to specify an algorithm.
- It cannot be compiled nor executed on computers, and has no real formatting or syntax rules.
- It enables the programmer to concentrate on the algorithms without worrying about all the syntactic details of a particular programming language
- Example:

  what follows is a pseudocode for a program that gets two integer numbers from the user, and then prints out their product.

```
Declare  X1, X2, Result
Read X1,X2
Result = X1*X2
Write Result
```

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements

- There are six basic pseudocode statements that are used for writing algorithms:

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements

- There are six basic pseudocode statements that are used for writing algorithms:

  1. **Reading (**or **getting**) data from an input device (keyboard, file, etc.).

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements

- There are six basic pseudocode statements that are used for writing algorithms:

  1. **Reading (**or **getting**) data from an input device (keyboard, file, etc.).

  2. **Writing** (or **displaying)** data to an output device (monitor, file, printer, etc.).

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements

- There are six basic pseudocode statements that are used for writing algorithms:

1. **Reading (**or **getting**) data from an input device (keyboard, file, etc.).
2. **Writing** (or **displaying)** data to an output device (monitor, file, printer, etc.).
3. **Calculating (**or **performing) arithmetic operation** on data like +, -, *, /.

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements

- There are six basic pseudocode statements that are used for writing algorithms:

1. **Reading (**or **getting**) data from an input device (keyboard, file, etc.).
2. **Writing** (or **displaying)** data to an output device (monitor, file, printer, etc.).
3. **Calculating (**or **performing) arithmetic operation** on data like +, -, *, /.
4. **Assigning** a value to a variable (fill a container).

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements

- There are six basic pseudocode statements that are used for writing algorithms:

1. **Reading (**or **getting**) data from an input device (keyboard, file, etc.).
2. **Writing** (or **displaying)** data to an output device (monitor, file, printer, etc.).
3. **Calculating (**or **performing) arithmetic operation** on data like +, -, *, /.
4. **Assigning** a value to a variable (fill a container).
5. **Selection structures such as "IF-THEN-ELSE": compares** two pieces of information and select one of two alternative statements.

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements

- There are six basic pseudocode statements that are used for writing algorithms:

1. **Reading (**or **getting**) data from an input device (keyboard, file, etc.).
2. **Writing** (or **displaying)** data to an output device (monitor, file, printer, etc.).
3. **Calculating (**or **performing) arithmetic operation** on data like +, -, *, /.
4. **Assigning** a value to a variable (fill a container).
5. **Selection structures such as "IF-THEN-ELSE": compares** two pieces of information and select one of two alternative statements.
6. **Repetition structures such as "WHILE" and "FOR": repeats** a statement or a group of statements for a certain number of times.

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements

- There are six basic pseudocode statements that are used for writing algorithms:

  1. **Reading (**or **getting**) data from an input device (keyboard, file, etc.).
  2. **Writing** (or **displaying)** data to an output device (monitor, file, printer, etc.).
  3. **Calculating (**or **performing) arithmetic operation** on data like +, -, *, /.
  4. **Assigning** a value to a variable (fill a container).
  5. **Selection structures such as "IF-THEN-ELSE": compares** two pieces of information and select one of two alternative statements.
  6. **Repetition structures such as "WHILE" and "FOR": repeats** a statement or a group of statements for a certain number of times.

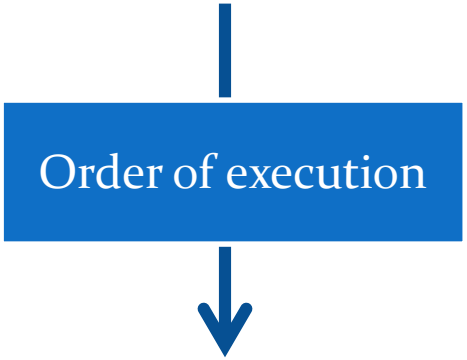
- According to the **structure theorem**, *any computer program can be written using only the three basic control structures*: **sequence**, **selection** and **repetition**.

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The Sequence Structure

- It is a series of actions are performed in a specific sequence indicated by the sequence they appear in the program (what comes first is executed first).

Statement 1
Statement 2
.
.
Statement n

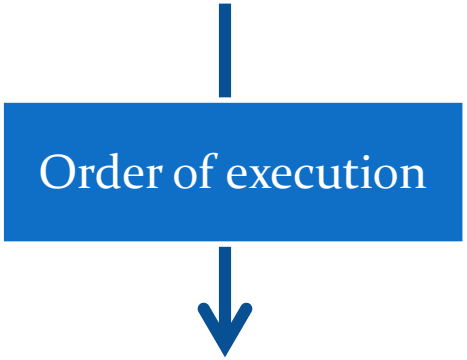


T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The Sequence Structure

- It is a series of actions are performed in a specific sequence indicated by the sequence they appear in the program (what comes first is executed first).

Statement 1
Statement 2
.
.
Statement n



- Example:

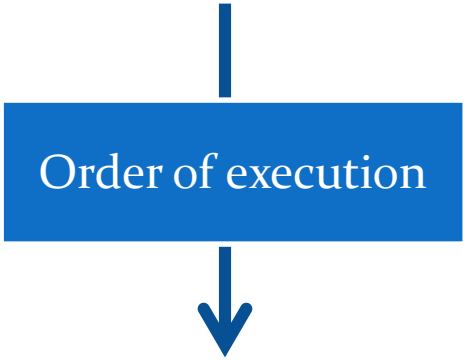
```
Declare  X1, X2, Sum
Read X1,X2
Sum = X1+X2
Write Sum
```


T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The Sequence Structure

- It is a series of actions are performed in a specific sequence indicated by the sequence they appear in the program (what comes first is executed first).

Statement 1
Statement 2
.
.
Statement n



- Example:

```
Declare  X1, X2, Sum
Read X1,X2
Sum = X1+X2
Write Sum
```

- Let us remember:



http://blogs.articulate.com/rapid-elearning/how-to-get-your-learners-to-remember-more/

A statement could be:

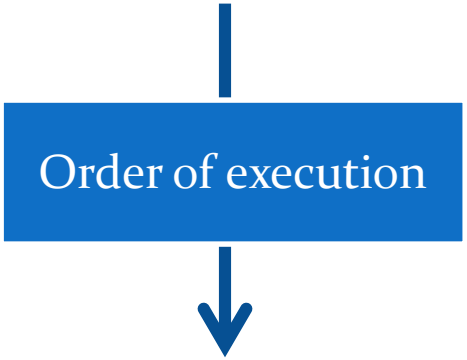
1.
2.
3.
4.
5.
6.

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The Sequence Structure

- It is a series of actions are performed in a specific sequence indicated by the sequence they appear in the program (what comes first is executed first).

Statement 1
Statement 2
.
.
Statement n



- Example:

```
Declare  X1, X2, Sum
Read X1,X2
Sum = X1+X2
Write Sum
```

- Let us remember:



http://blogs.articulate.com/rapid-elearning/how-to-get-your-learners-to-remember-more/

A statement could be:

1. Read
2. Write
3. Calculate
4. **Assign**
5. **Select**
6. **Repeat**

T.C. Adnan Menderes University Faculty of Engineering

www.cse.unr.edu/~fredrick/class/201/lecture/Pseudocode.doc

Basic Pseudocode Statements
Assigning a value

- **Assigning** a value to a variable (fill a container):

  On the left side of =, we put the name of a variable and on the right side we put a value or an expression.

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

- **Assigning** a value to a variable (fill a container).

  On the left side of =, we put the name of a variable and on the right side we put a value or an expression.

  An assignement is executed in two steps:

  1-Evaluation of the expression found on the right side.

  2-Setting the value returned from the evaluation in the memory cell corresponding to the variable on the left side.

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

- **Assigning** a value to a variable (fill a container).

On the left side of =, we put the name of a variable and on the right side we put a value or an expression.

An assignment is executed in two steps:

1-Evaluation of the expression found on the right side.

2-Setting the value returned from the evaluation in the memory cell corresponding to the variable on the left side.

Example 1:

RAM

?
?
?
?
?
?
?
?

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

- **Assigning** a value to a variable (fill a container).

On the left side of =, we put the name of a variable and on the right side we put a value or an expression.

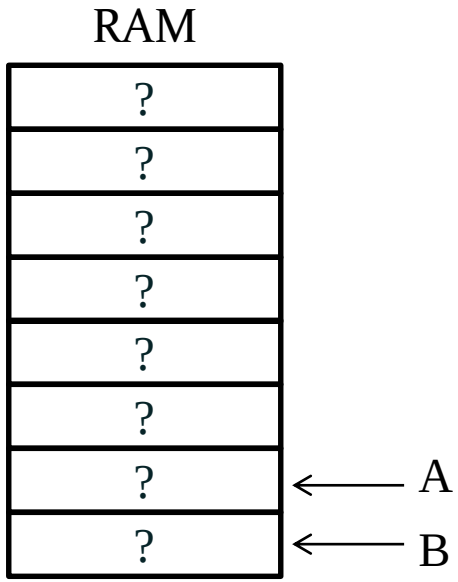
An assignment is executed in two steps:

1-Evaluation of the expression found on the right side.

2-Setting the value returned from the evaluation in the memory cell corresponding to the variable on the left side.

Example 1:

Declare A, B



T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

- **Assigning** a value to a variable (fill a container).

On the left side of =, we put the name of a variable and on the right side we put a value or an expression.

An assignment is executed in two steps:

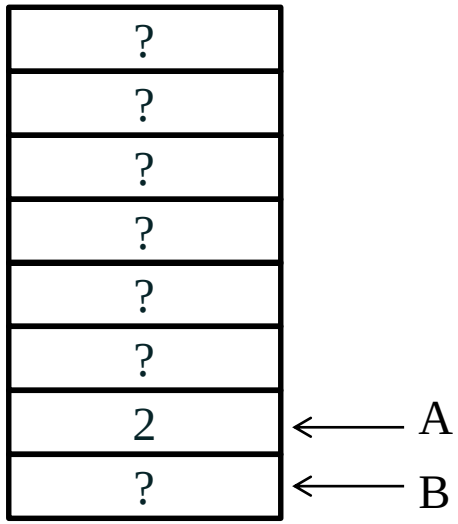
1-Evaluation of the expression found on the right side.

2-Setting the value returned from the evaluation in the memory cell corresponding to the variable on the left side.

Example 1:

Declare A, B

A=2



T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

- **Assigning** a value to a variable (fill a container).

On the left side of =, we put the name of a variable and on the right side we put a value or an expression.

An assignment is executed in two steps:

1-Evaluation of the expression found on the right side.

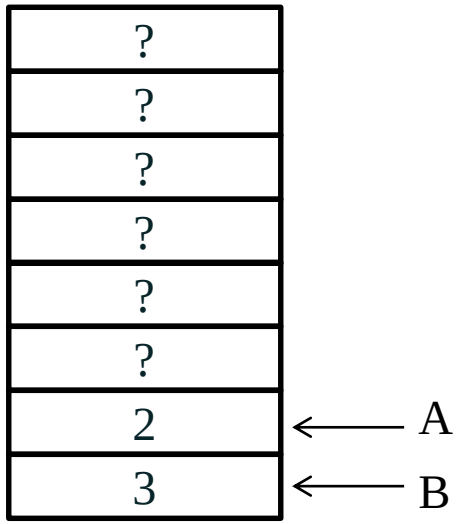
2-Setting the value returned from the evaluation in the memory cell corresponding to the variable on the left side.

Example 1:

Declare A, B

A=2

B=3



T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

- **Assigning** a value to a variable (fill a container).

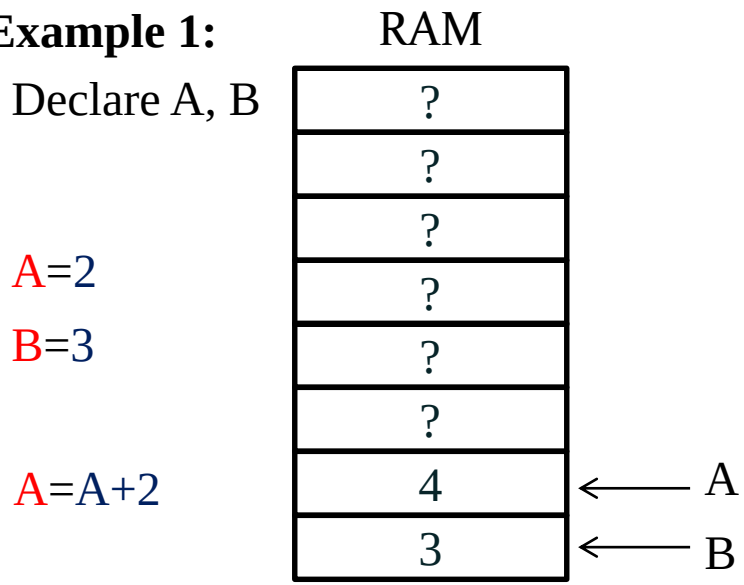
On the left side of =, we put the name of a variable and on the right side we put a value or an expression.

An assignment is executed in two steps:

1-Evaluation of the expression found on the right side.

2-Setting the value returned from the evaluation in the memory cell corresponding to the variable on the left side.

Example 1:



T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

- **Assigning** a value to a variable (fill a container).

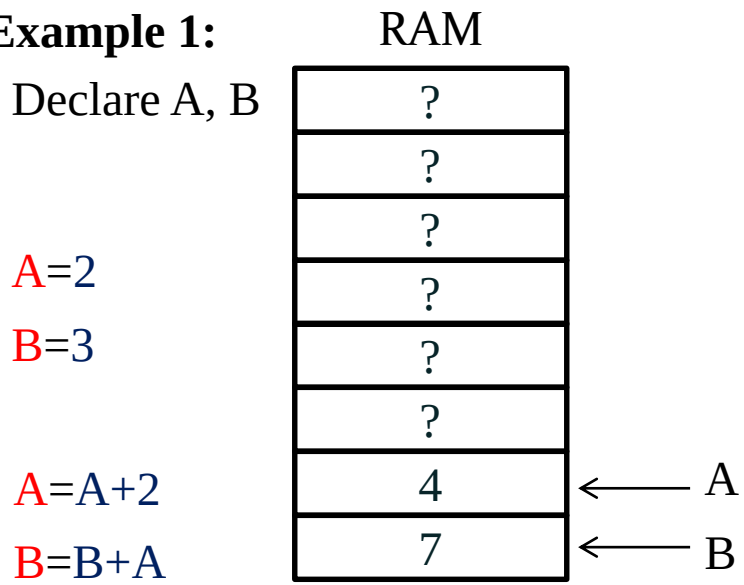
On the left side of =, we put the name of a variable and on the right side we put a value or an expression.

An assignment is executed in two steps:

1-Evaluation of the expression found on the right side.

2-Setting the value returned from the evaluation in the memory cell corresponding to the variable on the left side.

Example 1:



T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

- **Assigning** a value to a variable (fill a container).

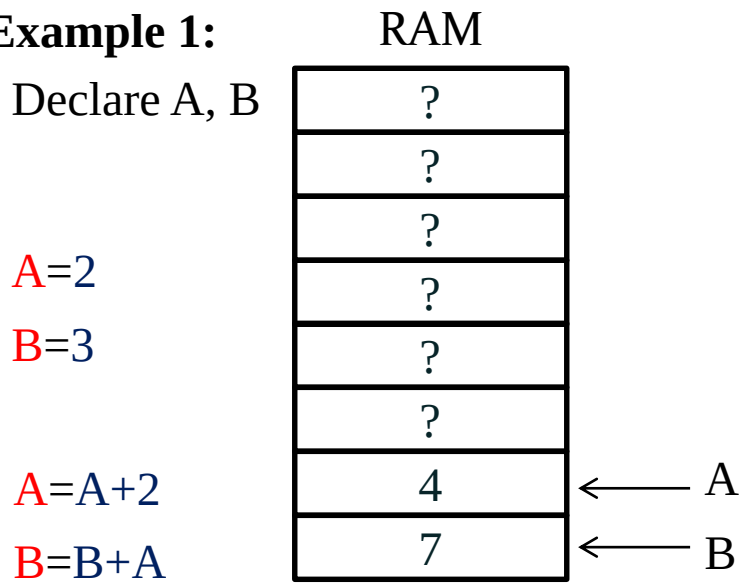
On the left side of =, we put the name of a variable and on the right side we put a value or an expression.

An assignment is executed in two steps:

1-Evaluation of the expression found on the right side.

2-Setting the value returned from the evaluation in the memory cell corresponding to the variable on the left side.

Example 1:



Notes:

- In computer science, the instruction A=A+2 means the new value of A is equal to the old one plus two.

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

- **Assigning** a value to a variable (fill a container).

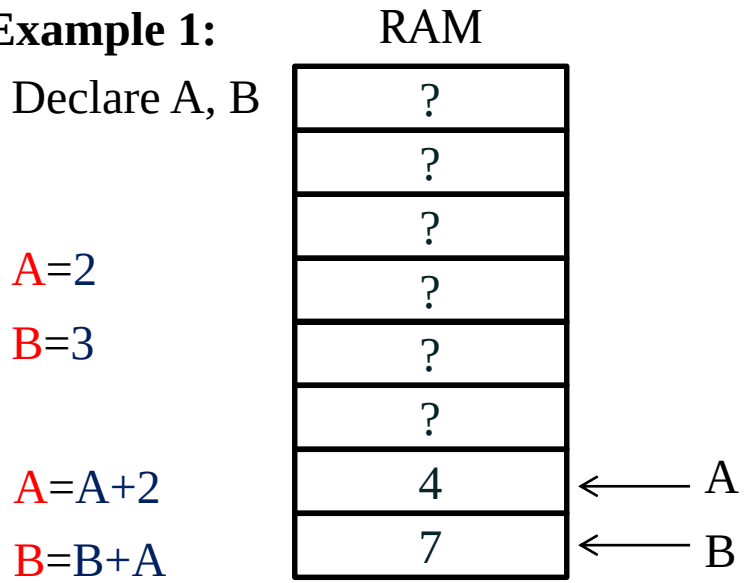
On the left side of =, we put the name of a variable and on the right side we put a value or an expression.

An assignment is executed in two steps:

1-Evaluation of the expression found on the right side.

2-Setting the value returned from the evaluation in the memory cell corresponding to the variable on the left side.

Example 1:



Notes:

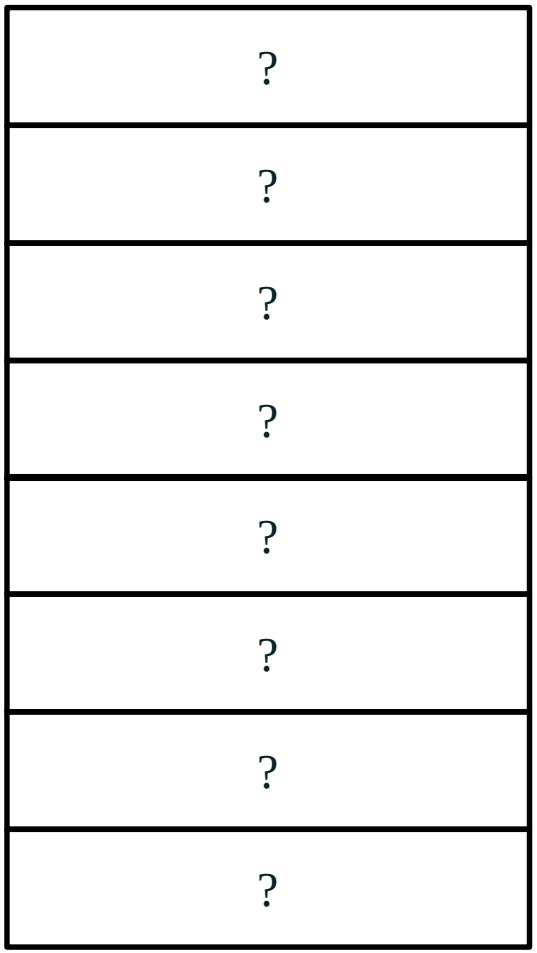
- In computer science, the instruction A=A+2 means the new value of A is equal to the old one plus two.
- The instruction A+2=A has no meaning in computer science (the left side must be a variable).

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

Example 2:

RAM

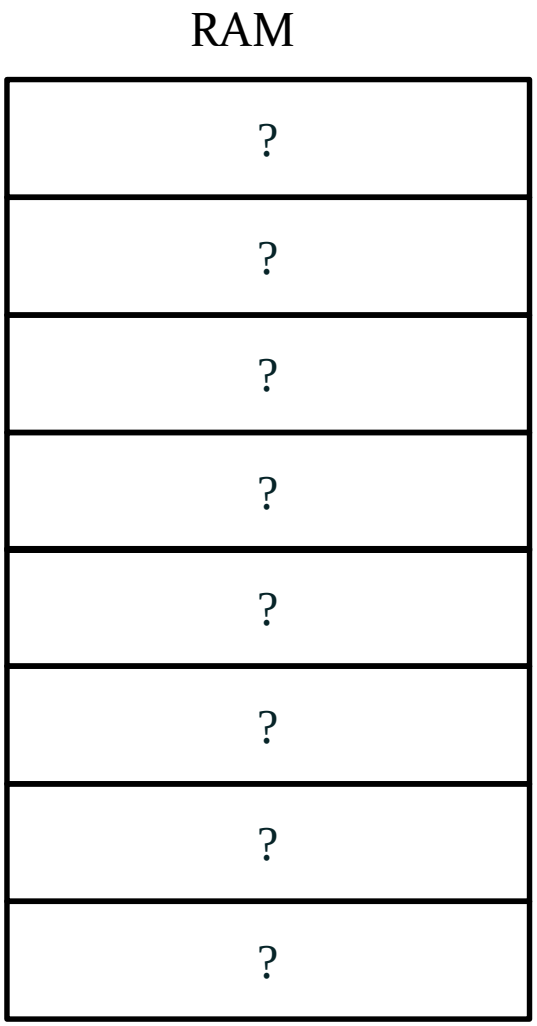


T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

Example 2:

Declare X, Y

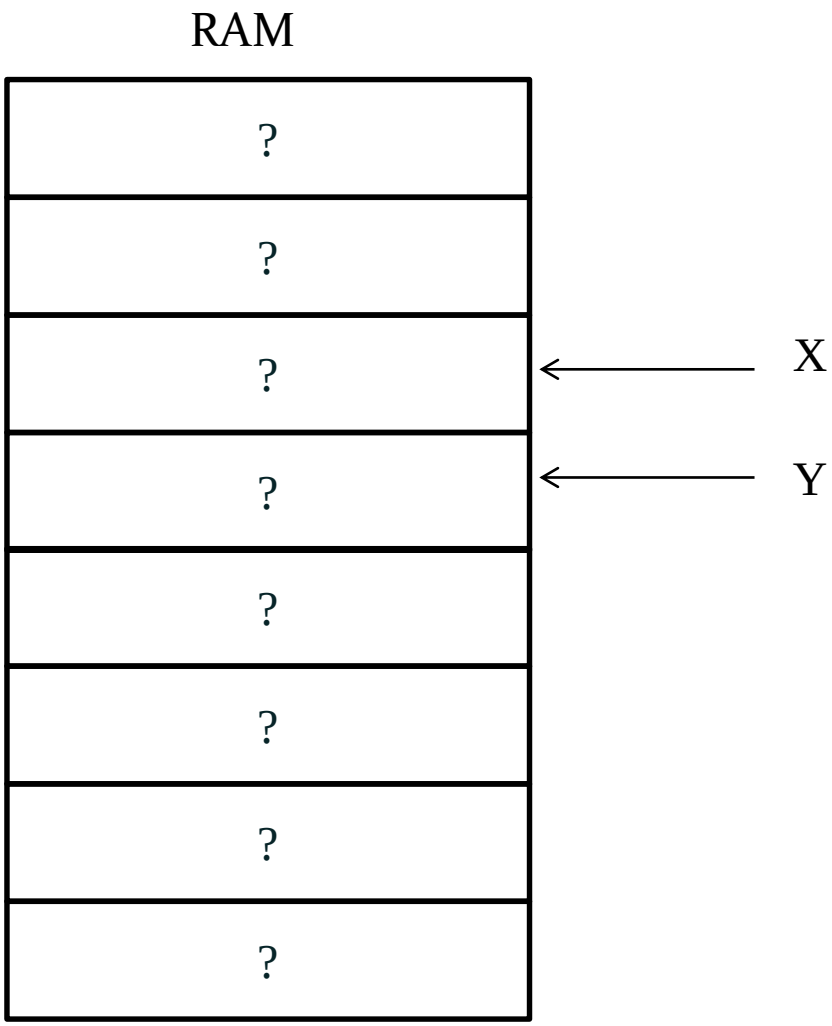


T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

Example 2:

Declare X, Y



T.C. Adnan Menderes University Faculty of Engineering

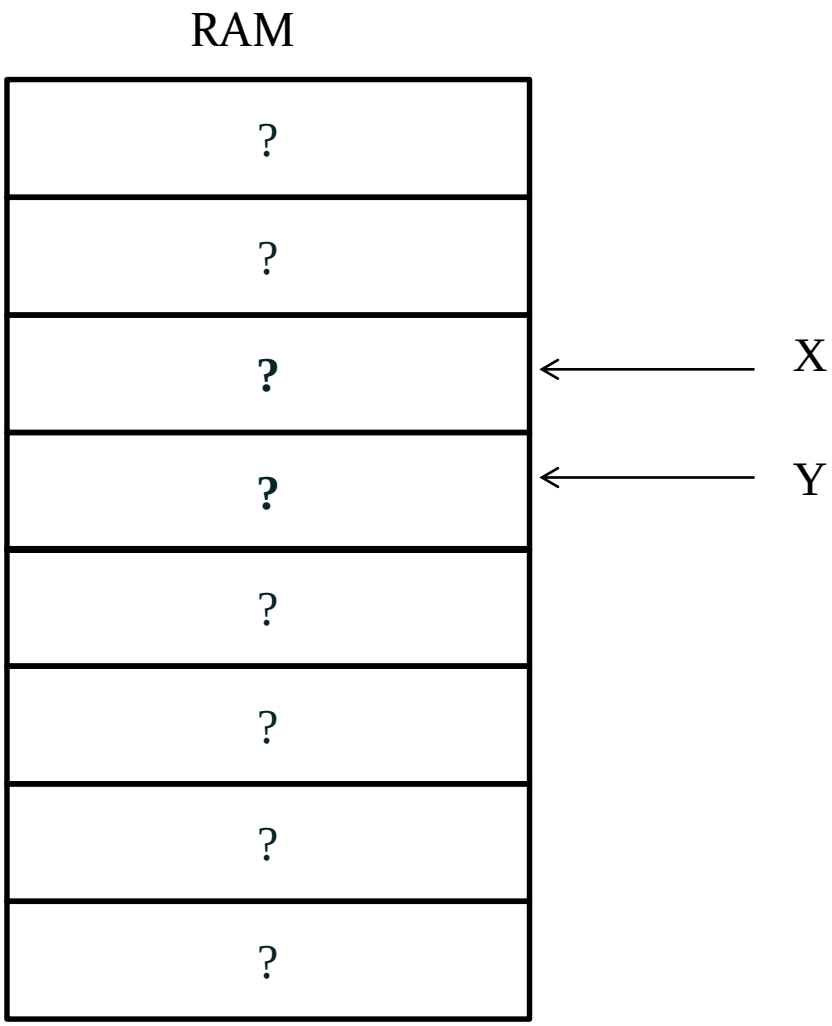
Basic Pseudocode Statements
Assigning a value

Example 2:

Declare X, Y

X=2

Y=3



T.C. Adnan Menderes University Faculty of Engineering

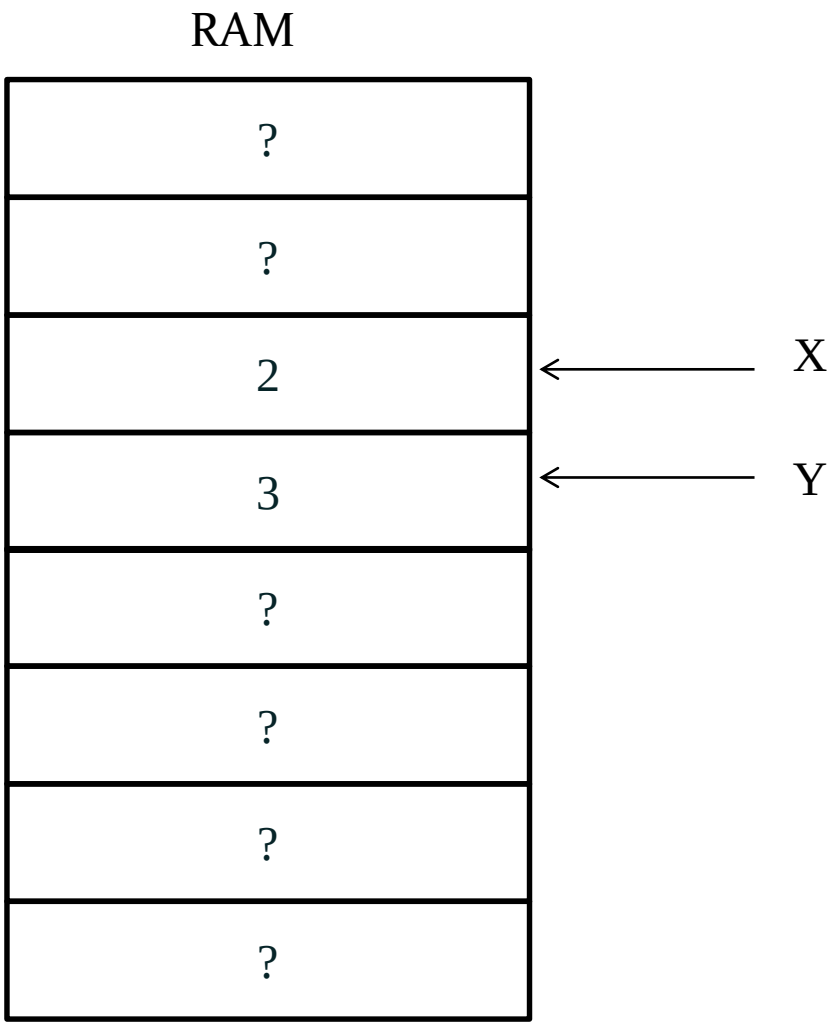
Basic Pseudocode Statements
Assigning a value

Example 2:

Declare X, Y

X=2

Y=3



T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

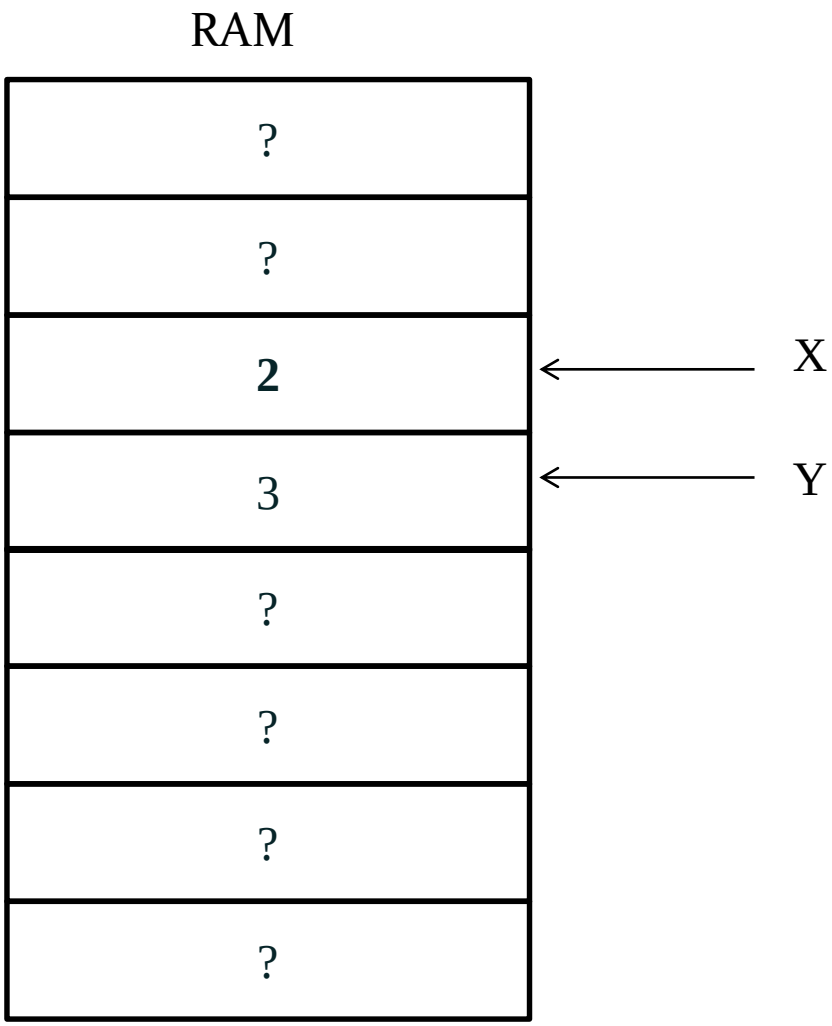
Example 2:

Declare X, Y

X=2

Y=3

X=X+1



T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

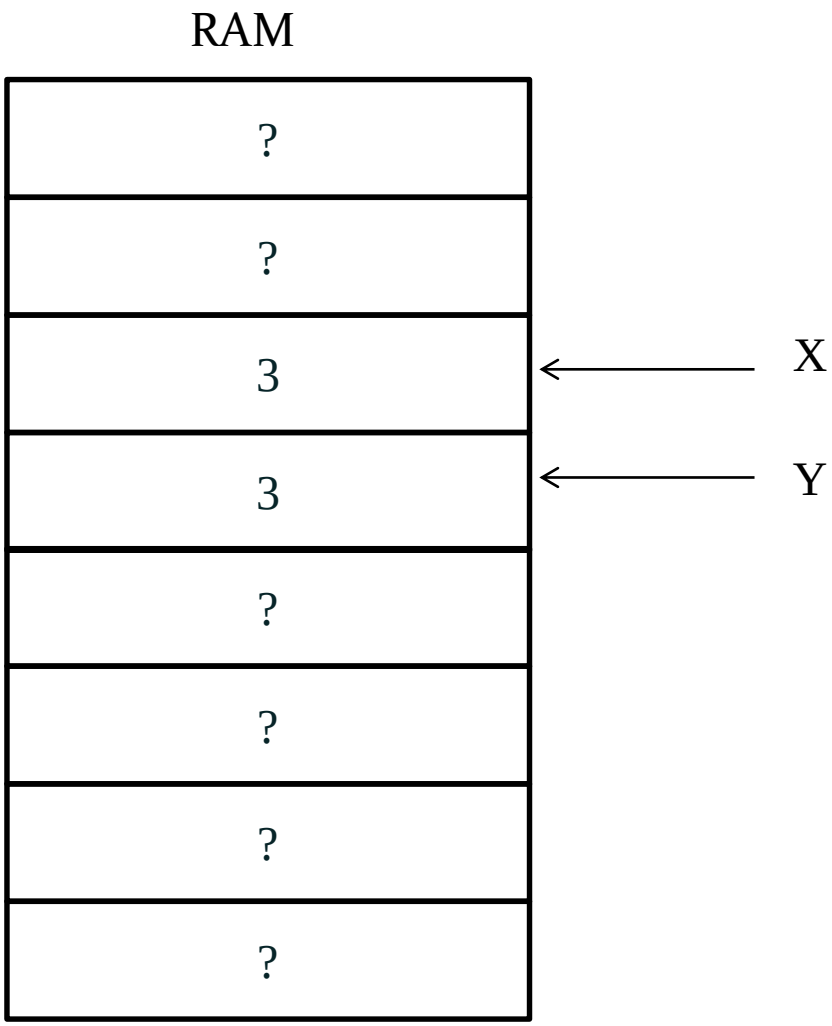
Example 2:

Declare X, Y

X=2

Y=3

X=X+1



T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

Example 2:

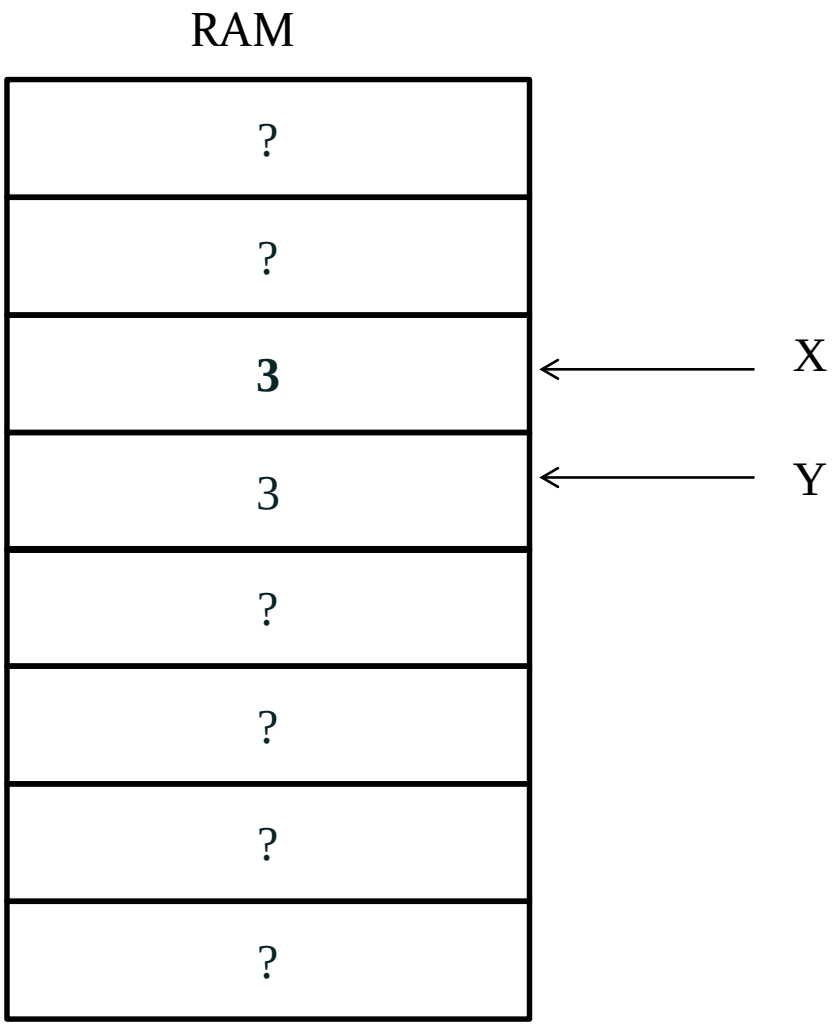
Declare X, Y

X=2

Y=3

X=X+1

X=X+1



T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

Example 2:

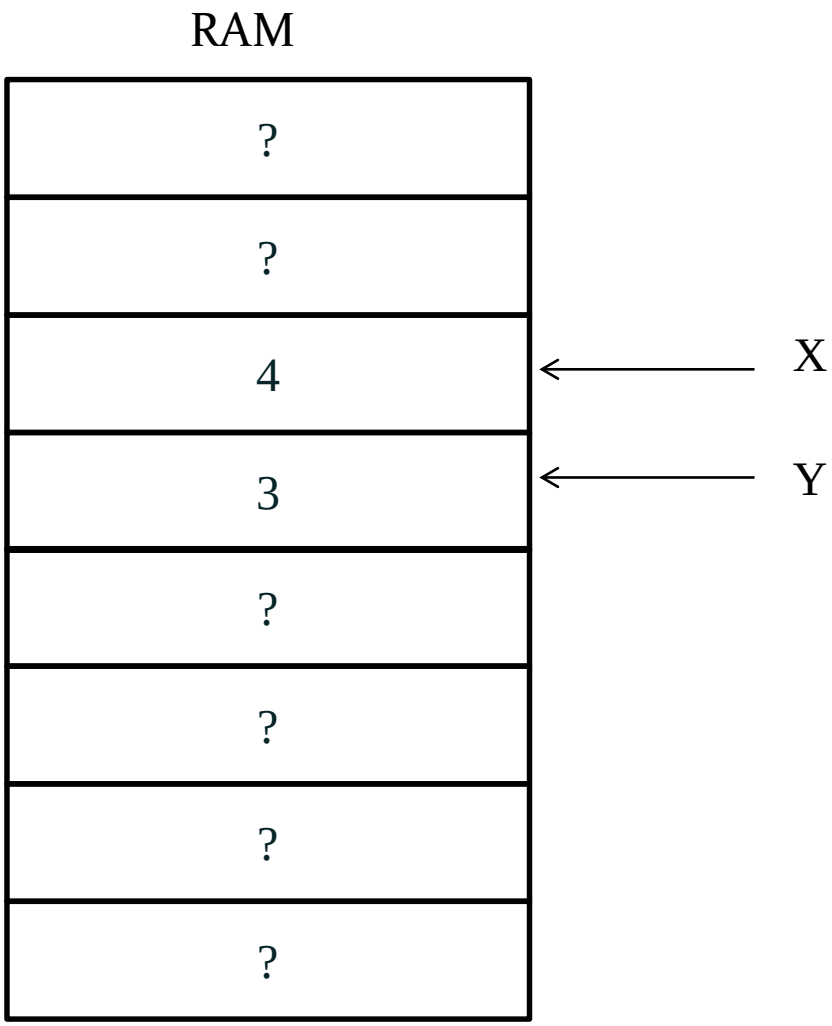
Declare X, Y

X=2

Y=3

X=X+1

X=X+1



T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

Example 2:

Declare X, Y

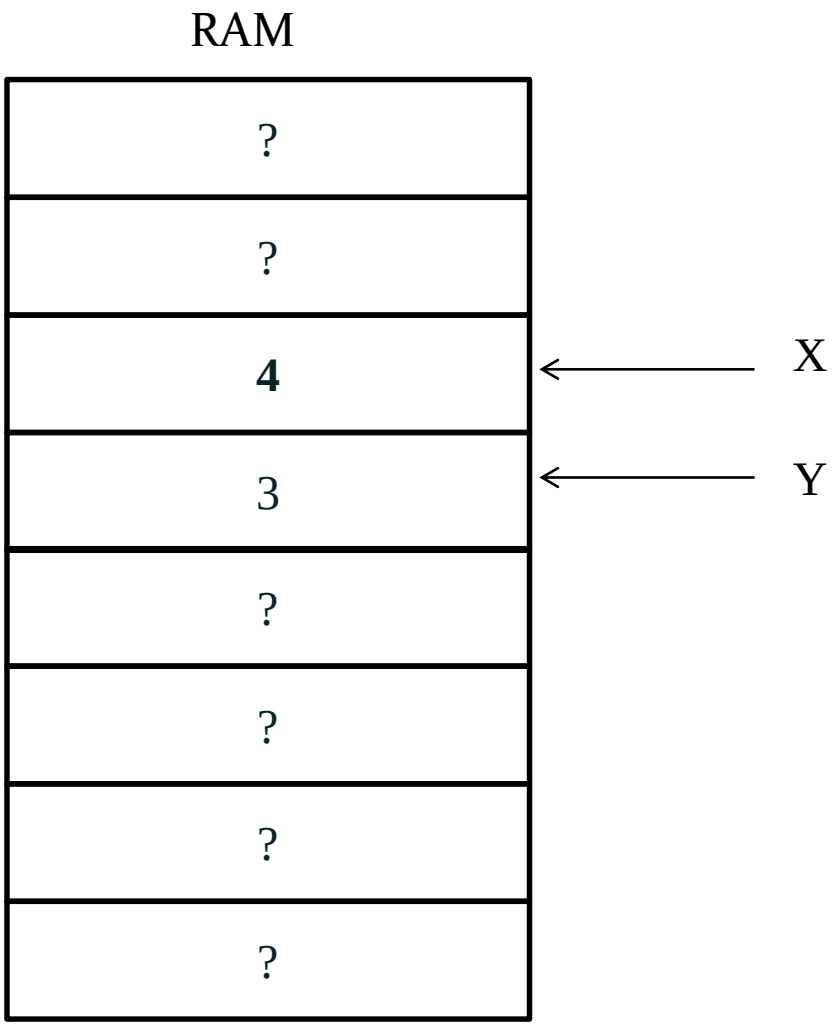
X=2

Y=3

X=X+1

X=X+1

X=X-Y



T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

Example 2:

Declare X, Y

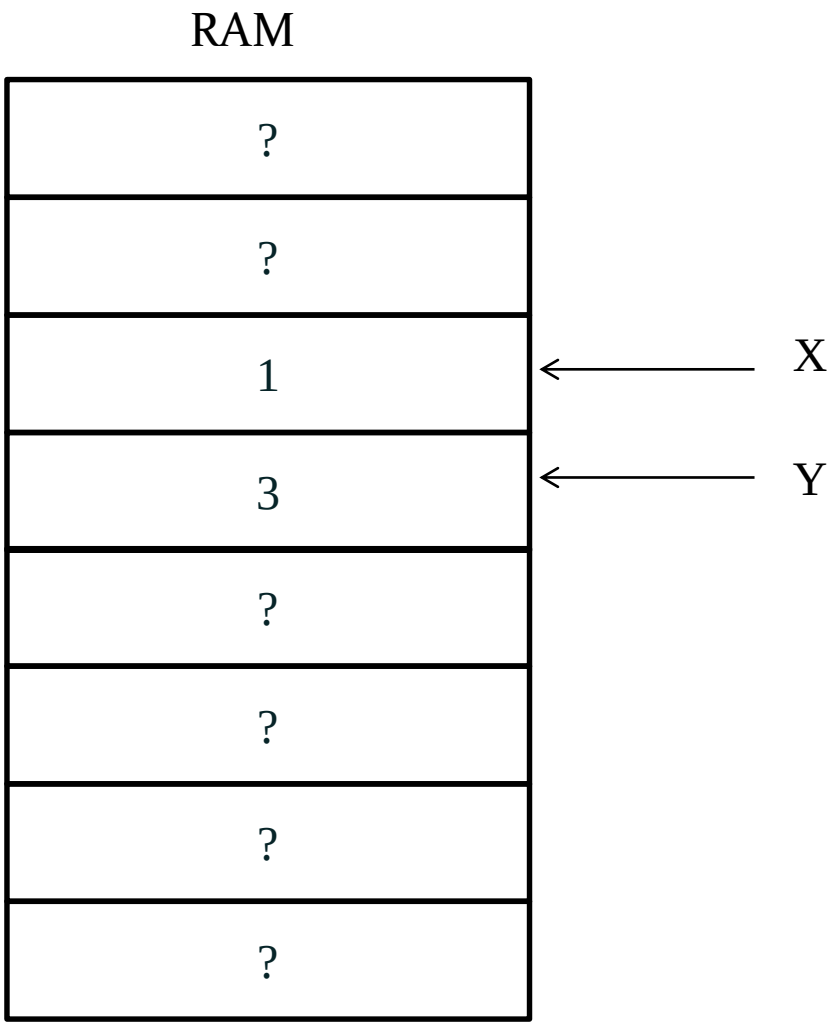
X=2

Y=3

X=X+1

X=X+1

X=X-Y



T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

Example 2:

Declare X, Y

X=2

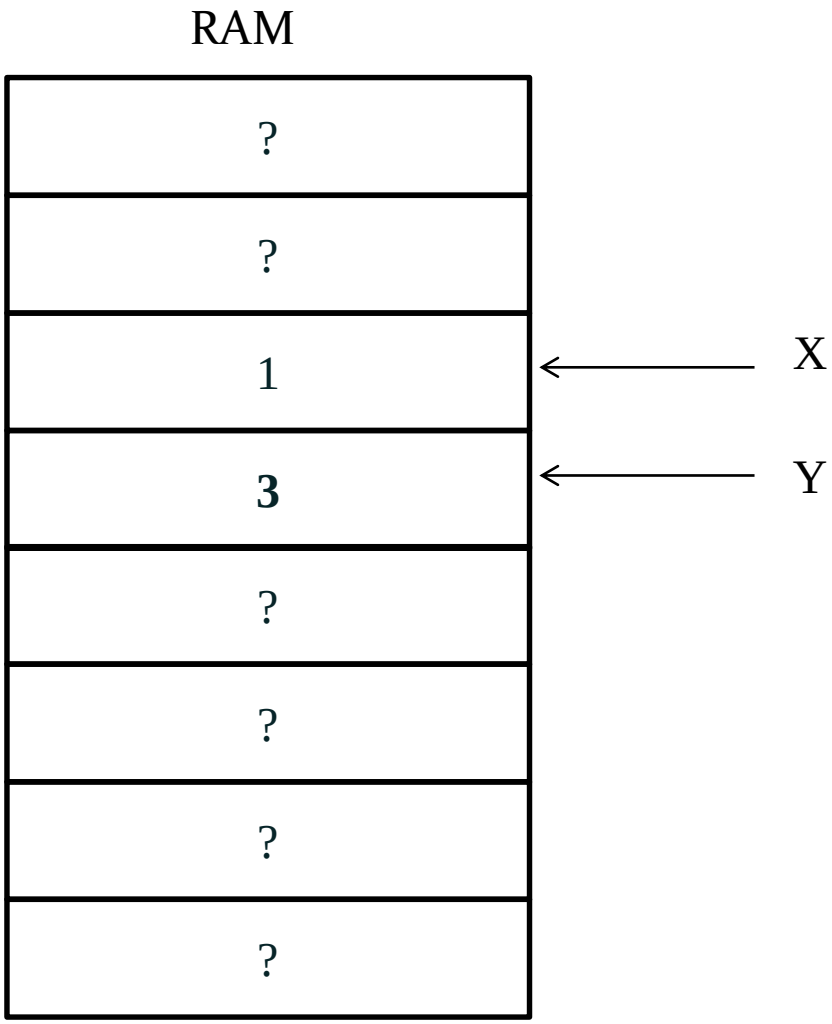
Y=3

X=X+1

X=X+1

X=X-Y

Y=X+1



T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

Example 2:

Declare X, Y

X=2

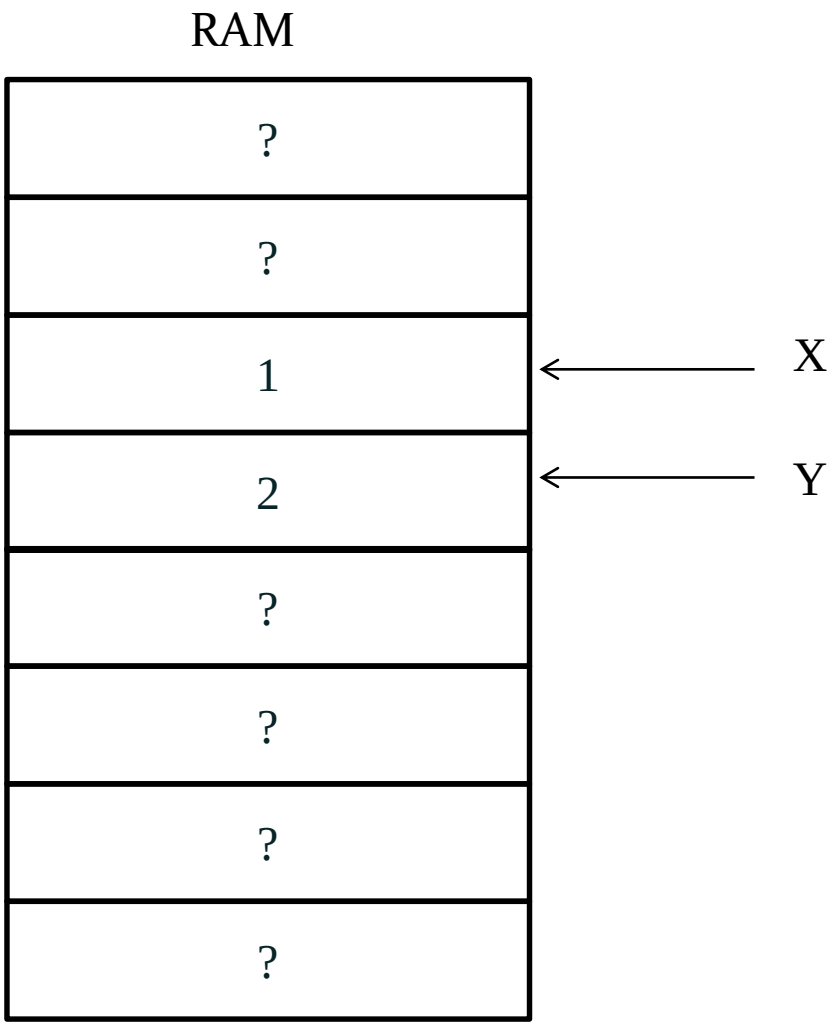
Y=3

X=X+1

X=X+1

X=X-Y

Y=X+1



T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

Example 2:

Declare X, Y

X=2

Y=3

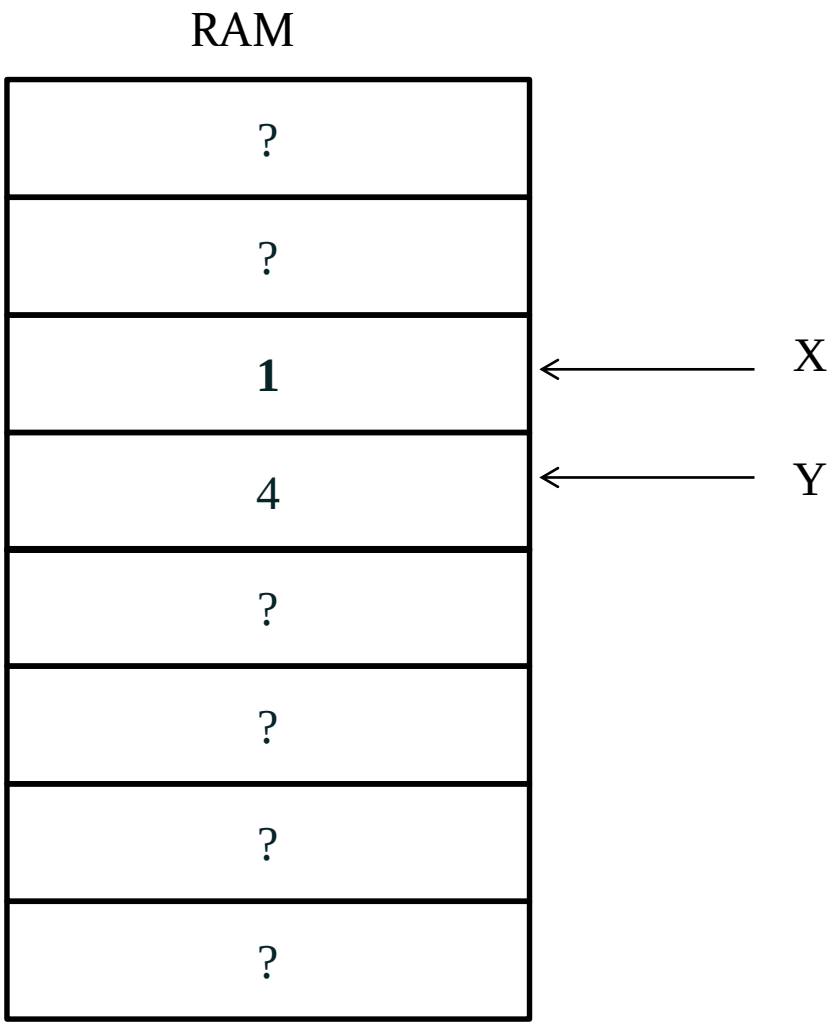
X=X+1

X=X+1

X=X-Y

Y=X+1

X=Y*X



T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

Example 2:

Declare X, Y

X=2

Y=3

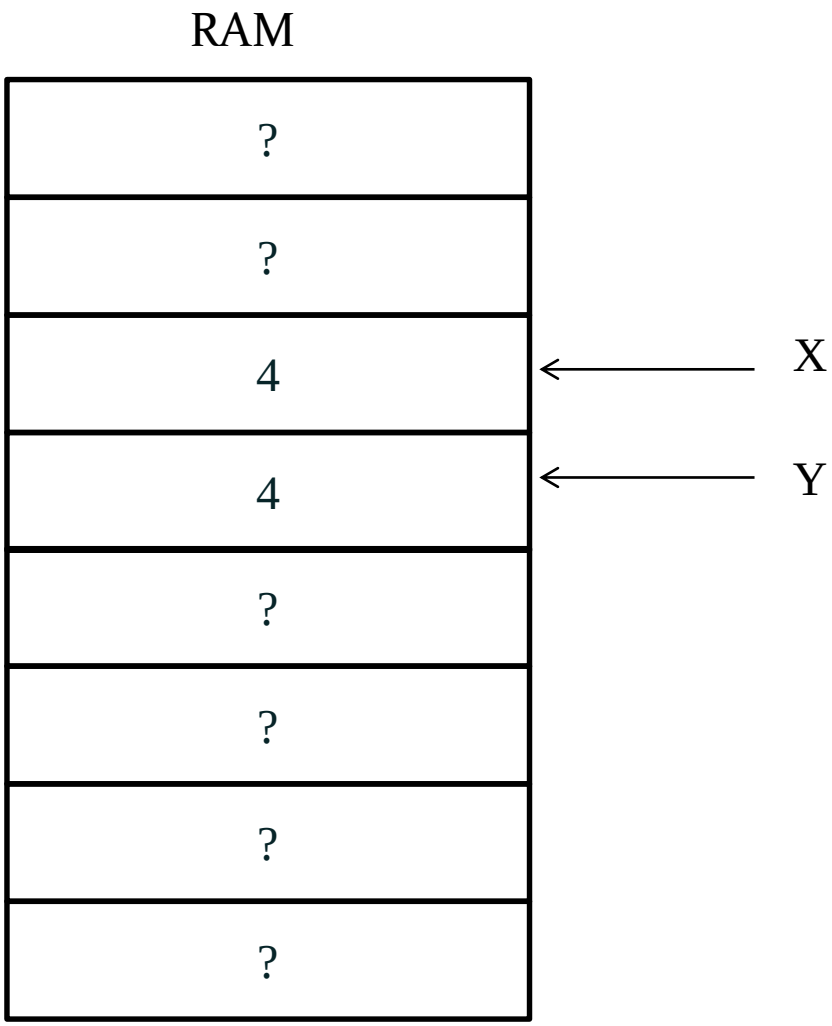
X=X+1

X=X+1

X=X-Y

Y=X+1

X=Y*X



T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

Example 2:

Declare X, Y

X=2

Y=3

X=X+1

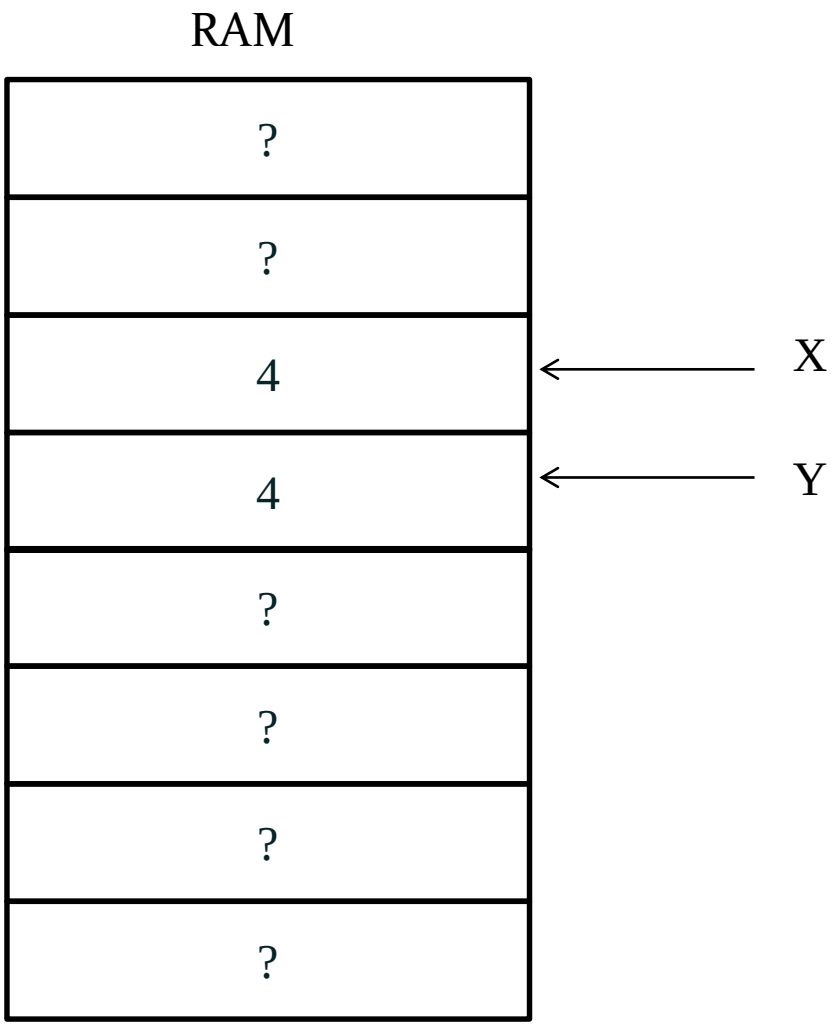
X=X+1

X=X-Y

Y=X+1

X=Y*X

Declare Z=100



T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

Example 2:

Declare X, Y

X=2

Y=3

X=X+1

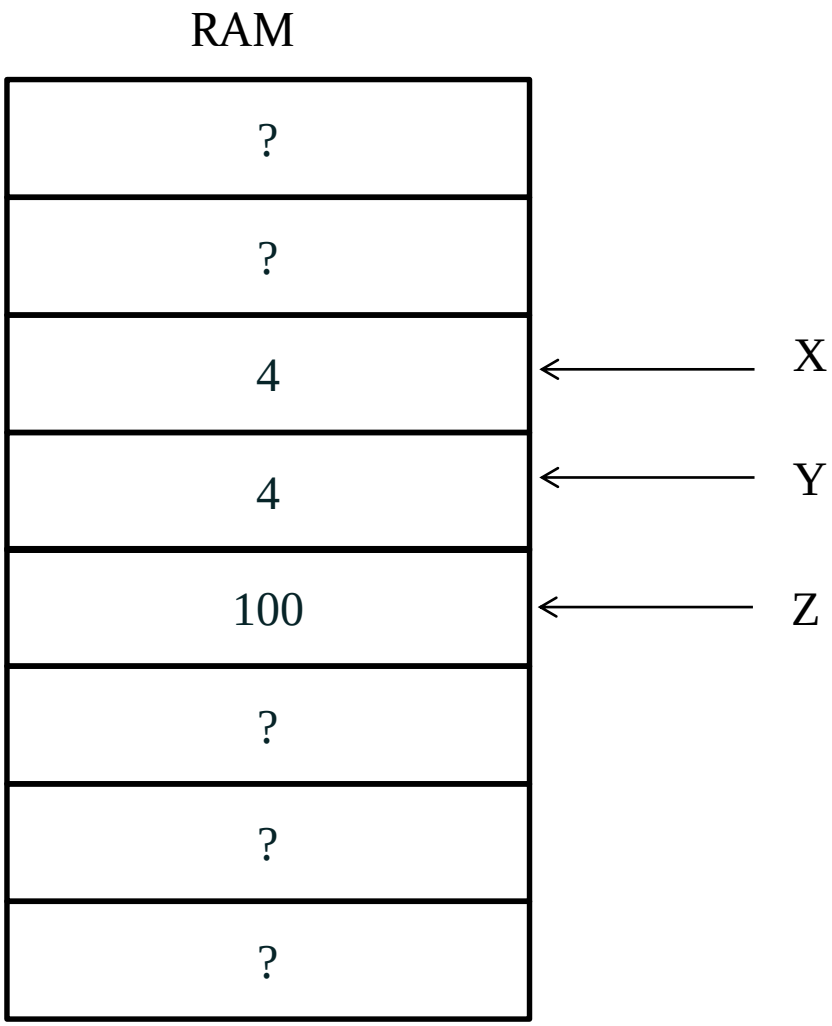
X=X+1

X=X-Y

Y=X+1

X=Y*X

Declare Z=100



T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

Example 2:

Declare X, Y

X=2

Y=3

X=X+1

X=X+1

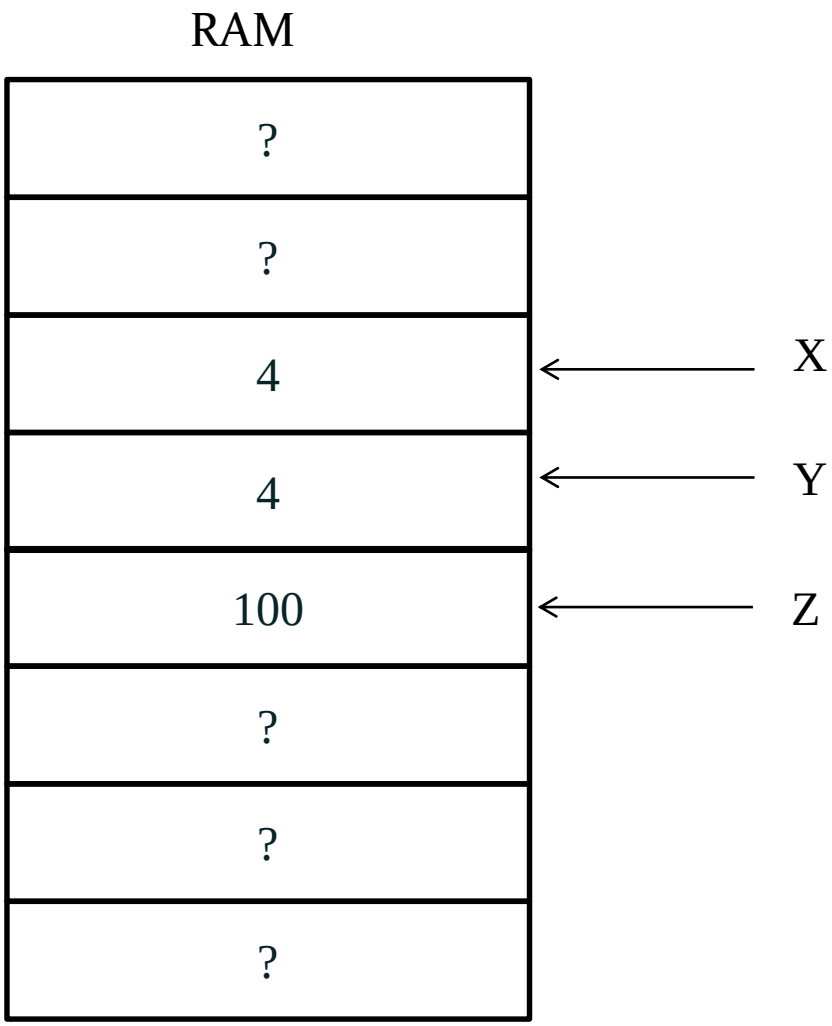
X=X-Y

Y=X+1

X=Y*X

Declare Z=100

Initialize Counter=0



T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Assigning a value

Example 2:

Declare X, Y

X=2

Y=3

X=X+1

X=X+1

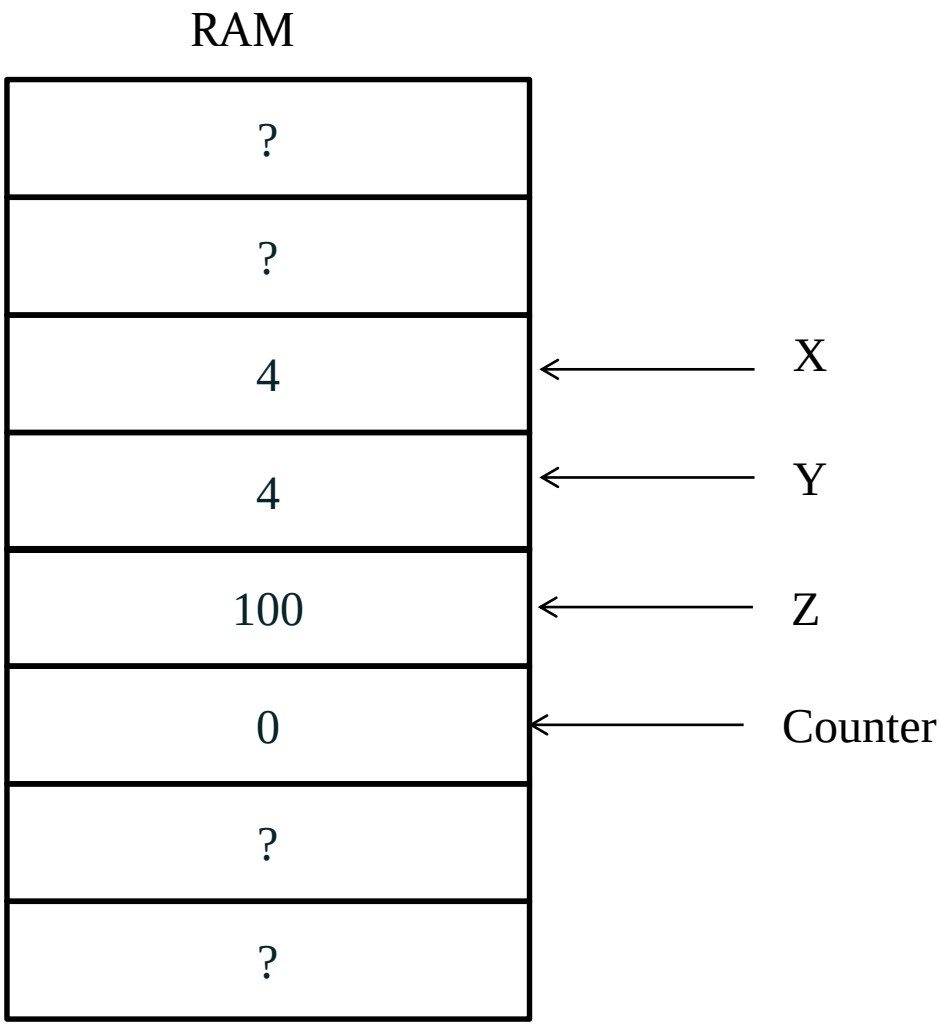
X=X-Y

Y=X+1

X=Y*X

Declare Z=100

Initialize Counter=0



T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Relational Operators

- A relational operator is a binary operator that compares two values.

T.C. Adnan Menderes University Faculty of Engineering

http://web.fuhsd.org/debbie_frazier/Lessons/L06if-else/L06if-else.html

Basic Pseudocode Statements
Relational Operators

- A relational operator is a binary operator that compares two values.
- The following symbols are almost used by all programmers:

<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to
==	equal to
!=	not equal to

T.C. Adnan Menderes University Faculty of Engineering

http://web.fuhsd.org/debbie_frazier/Lessons/L06if-else/L06if-else.html

Basic Pseudocode Statements
Relational Operators

- A relational operator is a binary operator that compares two values.
- The following symbols are almost used by all programmers:

<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to
==	equal to
!=	not equal to

- A condition (or relational expression) is an expression that contains at least one relational operators.
- The result of a relational expression is always
  boolean value, **TRUE** or **FALSE**.

T.C. Adnan Menderes University Faculty of Engineering

http://web.fuhsd.org/debbie_frazier/Lessons/L06if-else/L06if-else.html

Basic Pseudocode Statements
Relational Operators

- A relational operator is a binary operator that compares two values.
- The following symbols are almost used by all programmers:

<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to
==	equal to
!=	not equal to

- A condition (or relational expression) is an expression that contains at least one relational operators.
- The result of a relational expression is always **boolean** value, **TRUE** or **FALSE**.
- Examples:
  - X1=12,X2=10, (X1>X2)→ TRUE.
  - X1=12,X2=10, (X1<X2)→ FALSE.
  - X1=12,X2=12, (X1!=X2)→ FALSE.
  - A1=12,A2=10, (A1>=A2)→ .
  - B2=0,C1=10, (B2==C1)→ .

http://web.fuhsd.org/debbie_frazier/Lessons/L06if-else/L06if-else.html

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Relational Operators

- A relational operator is a binary operator that compares two values.
- The following symbols are almost used by all programmers:

<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to
==	equal to
!=	not equal to

- A condition (or relational expression) is an expression that contains at least one relational operators.
- The result of a relational expression is always **boolean** value, **TRUE** or **FALSE**.
- Examples:
  - X1=12,X2=10, (X1>X2)→ TRUE.
  - X1=12,X2=10, (X1<X2)→ FALSE.
  - X1=12,X2=12, (X1!=X2)→ FALSE.
  - A1=12,A2=10, (A1>=A2)→ .
  - B2=0,C1=10, (B2==C1)→ .
  - (100>2)→ TRUE.
  - ((100>2) ==FALSE)→ FALSE.
  - (-10<-100)→ FALSE.
  - (2==(3-1))→ .
  - B2=0,(B2==(10-1))→ .

http://web.fuhsd.org/debbie_frazier/Lessons/L06if-else/L06if-else.html

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Logical Operators

- The three logical operators of programming are **AND**, **OR**, and **NOT**.

T.C. Adnan Menderes University Faculty of Engineering

http://web.fuhsd.org/debbie_frazier/Lessons/L06if-else/L06if-else.html

Basic Pseudocode Statements
Logical Operators

- The three logical operators of programming are **AND**, **OR**, and **NOT**.
- The **AND** operator requires both operands (values) to be true for the result to be true.

TRUE **AND** TRUE	= TRUE
TRUE **AND** FALSE	= FALSE
FALSE **AND** TRUE	= FALSE
FALSE **AND** FALSE	= FALSE

http://web.fuhsd.org/debbie_frazier/Lessons/L06if-else/L06if-else.html

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Logical Operators

- The three logical operators of programming are **AND**, **OR**, and **NOT**.

- The **AND** operator requires both operands (values) to be true for the result to be true.

TRUE **AND** TRUE	= TRUE
TRUE **AND** FALSE	= FALSE
FALSE **AND** TRUE	= FALSE
FALSE **AND** FALSE	= FALSE

- The **OR** operator requires one operand (values) to be true for the result to be true.

TRUE **OR** TRUE	= TRUE
TRUE **OR** FALSE	= TRUE
FALSE **OR** TRUE	= TRUE
FALSE **OR** FALSE	= FALSE

http://web.fuhsd.org/debbie_frazier/Lessons/L06if-else/L06if-else.html

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Logical Operators

- The three logical operators of programming are **AND**, **OR**, and **NOT**.

- The **AND** operator requires both operands (values) to be true for the result to be true.

TRUE **AND** TRUE	= TRUE
TRUE **AND** FALSE	= FALSE
FALSE **AND** TRUE	= FALSE
FALSE **AND** FALSE	= FALSE

- The **OR** operator requires one operand (values) to be true for the result to be true.

TRUE **OR** TRUE	= TRUE
TRUE **OR** FALSE	= TRUE
FALSE **OR** TRUE	= TRUE
FALSE **OR** FALSE	= FALSE

- The **NOT** operator is a unary operator that changes a boolean value to its opposite.

NOT TRUE	= FALSE
NOT FALSE	= TRUE

T.C. Adnan Menderes University Faculty of Engineering

http://web.fuhsd.org/debbie_frazier/Lessons/L06if-else/L06if-else.html

Basic Pseudocode Statements
Logical Operators

- The three logical operators of programming are **AND**, **OR**, and **NOT**.

- The **AND** operator requires both operands (values) to be true for the result to be true.

TRUE **AND** TRUE	= TRUE
TRUE **AND** FALSE	= FALSE
FALSE **AND** TRUE	= FALSE
FALSE **AND** FALSE	= FALSE

- The **OR** operator requires one operand (values) to be true for the result to be true.

TRUE **OR** TRUE	= TRUE
TRUE **OR** FALSE	= TRUE
FALSE **OR** TRUE	= TRUE
FALSE **OR** FALSE	= FALSE

- The **NOT** operator is a unary operator that changes a boolean value to its opposite.

NOT TRUE	= FALSE
NOT FALSE	= TRUE

- Examples:
- ((2+3 < 10) AND (21 > 19)) → T.
- ((15< 10) AND (21 > 19)) → F.
- ((5==10) AND (21 <= 19)) → F.
- ((TRUE) OR (2 > 19)) → T.

http://web.fuhsd.org/debbie_frazier/Lessons/L06if-else/L06if-else.html

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
Logical Operators

- The three logical operators of programming are **AND**, **OR**, and **NOT**.

- The **AND** operator requires both operands (values) to be true for the result to be true.

TRUE **AND** TRUE	= TRUE
TRUE **AND** FALSE	= FALSE
FALSE **AND** TRUE	= FALSE
FALSE **AND** FALSE	= FALSE

- The **OR** operator requires one operand (values) to be true for the result to be true.

TRUE **OR** TRUE	= TRUE
TRUE **OR** FALSE	= TRUE
FALSE **OR** TRUE	= TRUE
FALSE **OR** FALSE	= FALSE

- The **NOT** operator is a unary operator that changes a boolean value to its opposite.

NOT TRUE	= FALSE
NOT FALSE	= TRUE

- Examples:
- ((2+3 < 10) AND (21 > 19)) → T.
- ((15< 10) AND (21 > 19)) → F.
- ((5==10) AND (21 <= 19)) → F.
- ((TRUE) OR (2 > 19)) → T.
- (NOT (100>2))→ F.
- ((NOT(10!=100) AND (10<=12))→F.
- ((NOT(3<1)) OR (1>19))→ .
- ((9==(10-1) OR (NOT(20>2)))→ .

http://web.fuhsd.org/debbie_frazier/Lessons/L06if-else/L06if-else.html

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The "IF-THEN-ELSE" Selection Structure

- **The decesion statement " If-Then-Else" Compares** two pieces of information and select one of two alternative statements.
- The If-Then-Else may have this form:

IF (condition)
THEN
Statement(s)
ELSE
Alternative Statement (s)
ENDIF

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The "IF-THEN-ELSE" Selection Structure

- **The decesion statement " If-Then-Else" Compares** two pieces of information and select one of two alternative statements.
- The If-Then-Else may have this form:

IF (condition)
THEN
Statement(s)
ELSE
Alternative Statement (s)
ENDIF

Example 1:

IF (Total-Mark >49)
THEN
Write Pass
ELSE
Write Fail
ENDIF

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The "IF-THEN-ELSE" Selection Structure

- **The decesion statement " If-Then-Else" Compares** two pieces of information and select one of two alternative statements.
- The If-Then-Else may have this form:

```
IF (condition)
  THEN
    Statement(s)
  ELSE
    Alternative Statement (s)
ENDIF
```

Example 2:

```
IF (Absence-Hours >12)
  THEN
    Write Fail
  ELSE
    Write Pass
ENDIF
```

Example 1:

```
IF (Total-Mark >49)
  THEN
    Write Pass
  ELSE
    Write Fail
ENDIF
```

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The "IF-THEN-ELSE" Selection Structure

- **The decesion statement " If-Then-Else" Compares** two pieces of information and select one of two alternative statements.
- The If-Then-Else may have this form:

IF (condition)
THEN
Statement(s)
ELSE
Alternative Statement (s)
ENDIF

Example 2:
IF (Absence-Hours >12)
THEN
Write Fail
ELSE
Write Pass
ENDIF

Example 1:
IF (Total-Mark >49)
THEN
Write Pass
ELSE
Write Fail
ENDIF

Example 3:
IF (The class is finished)
THEN
Write Turn on your mobile
ELSE
Write Turn off your mobile
ENDIF

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The "IF-THEN-ELSE" Selection Structure

- **The decesion statement " If-Then-Else" Compares** two pieces of information and select one of two alternative statements.
- The If-Then-Else may have this form:

IF (condition)
→**THEN**
→Statement(s)
→**ELSE**
→Alternative Statement (s)
ENDIF

Adding Indentation to show the dependency is essential.

Example 1:

IF (Total-Mark >49)
→**THEN**
→Write Pass
→**ELSE**
→Write Fail
ENDIF

Example 2:

IF (Absence-Hours >12)
→**THEN**
→Write Fail
→**ELSE**
→Write Pass
ENDIF

Example 3:

IF (The class is finished)
→**THEN**
→Write Turn on your mobile
→**ELSE**
→Write Turn off your mobile
ENDIF

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The "IF-THEN" Selection Structure

- We may need to do nothing if the condition is evaluated to FALSE, in this case we can omit the ELSE part, and The If-Then will have this form:

IF (condition)

THEN

Statement(s)

ENDIF

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The "IF-THEN" Selection Structure

- We may need to do nothing if the condition is evaluated to FALSE, in this case we can omit the ELSE part, and The If-Then will have this form:

IF (condition)
THEN
Statement(s)
ENDIF

Example 1:
IF (a student is absent)
THEN
Increase his absence hours
EENDIF

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The "IF-THEN" Selection Structure

- We may need to do nothing if the condition is evaluated to FALSE, in this case we can omit the ELSE part, and The If-Then will have this form:

IF (condition)
THEN
Statement(s)
ENDIF

Example 1:
IF (a student is absent)
THEN
Increase his absence hours
EENDIF

Example 2:
IF (student grade > 49)
THEN
Add his name to the list
ENDIF

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The "IF-THEN" Selection Structure

- We may need to do nothing if the condition is evaluated to FALSE, in this case we can omit the ELSE part, and The If-Then will have this form:

IF (condition)
THEN
Statement(s)
ENDIF

Example 1:
IF (a student is absent)
THEN
Increase his absence hours
EENDIF

Example 2:
IF (student grade > 49)
THEN
Add his name to the list
ENDIF

Example 3:
IF (a student has an active paticipation record)
THEN
He may get a bonus from his teacher
ENDIF

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The "WHILE" Repetition Structure

- **The "WHILE" repetition statements repeats** a statement or a group of statements for a certain number of times.
- It may have the following form.

WHILE (condition)

Statement(s)

ENDWHILE

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The "WHILE" Repetition Structure

- **The "WHILE" repetition statements repeats** a statement or a group of statements for a certain number of times.
- It may have the following form.

WHILE (condition)

Statement(s)

ENDWHILE

- Example 1:

Initialize Counter=0

WHILE (Counter< 5)

Prepare a new quiz.

Let the student do it.

Counter=Counter+1

ENDWHILE

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The "WHILE" Repetition Structure

- **The "WHILE" repetition statements repeats** a statement or a group of statements for a certain number of times.
- It may have the following form.

WHILE (condition)
  Statement(s)
ENDWHILE

- Example 1:

Initialize Counter=0
WHILE (Counter< 5)
  Prepare a new quiz.
  Let the student do it.
  Counter=Counter+1
ENDWHILE

- Example 2:

Declare Number
Initialize Counter=0,Total=0
WHILE (Counter <3)
  Read Number
  Total=Total+Number
ENDWHILE
Write Total

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The "WHILE" Repetition Structure

- **The "WHILE" repetition statements repeats** a statement or a group of statements for a certain number of times.
- It may have the following form.

WHILE (condition)
  Statement(s)
ENDWHILE

- Example 1:

Initialize Counter=0
WHILE (Counter< 5)
  Prepare a new quiz.
  Let the student do it.
  Counter=Counter+1
ENDWHILE

How many iterations?
Will this program ever stop?

- Example 2:

Declare Number
Initialize Counter=0,Total=0
WHILE (Counter <3)
  Read Number
  Total=Total+Number
ENDWHILE
Write Total

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The "WHILE" Repetition Structure

- **The "WHILE" repetition statements repeats** a statement or a group of statements for a certain number of times.
- It may have the following form.

WHILE (condition)
  Statement(s)
ENDWHILE

- Example 1:

Initialize Counter=0
WHILE (Counter< 5)
  Prepare a new quiz.
  Let the student do it.
  Counter=Counter+1
ENDWHILE

- Example 2:

Declare Number
Initialize Counter=0,Total=0
WHILE (Counter <3)
  Read Number
  Total=Total+Number
  What is missed here?
ENDWHILE
Write Total

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The "WHILE" Repetition Structure

- **The "WHILE" repetition statements repeats** a statement or a group of statements for a certain number of times.
- It may have the following form.

WHILE (condition)
  Statement(s)
ENDWHILE

- Example 1:

Initialize Counter=0
WHILE (Counter< 5)
  Prepare a new quiz.
  Let the student do it.
  Counter=Counter+1
ENDWHILE

- Example 2:

Declare Number
Initialize Counter=0,Total=0
WHILE (Counter <3)
  Read Number
  Total=Total+Number
  Counter=Counter+1
ENDWHILE
Write Total

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The "FOR" Repetition Structure

- **The "FOR" repetition statements repeats** a statement or a group of statements for a certain number of times.
- It may have the following form.

FOR (a certain number of iterations)

    Statement(s)

ENDFOR

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The "FOR" Repetition Structure

- **The "FOR" repetition statements repeats** a statement or a group of statements for a certain number of times.
- It may have the following form.

FOR (a certain number of iterations)

    Statement(s)

ENDFOR

Example 1:

FOR (Counter =0; Counter <10; Counter =Counter +1)

  Prepare a new quiz

  Let the students do it

ENDFOR

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The "FOR" Repetition Structure

- **The "FOR" repetition statements repeats** a statement or a group of statements for a certain number of times.
- It may have the following form.

FOR (a certain number of iterations)
    Statement(s)
ENDFOR

Example 1:

FOR (Counter =0; Counter <10; Counter =Counter +1)
  Prepare a new quiz
  Let the students do it
ENDFOR
Or
FOR (Counter =1:10)
  Prepare a new quiz
  Let the students do it
ENDFOR

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The "FOR" Repetition Structure

- **The "FOR" repetition statements repeats** a statement or a group of statements for a certain number of times.
- It may have the following form.

FOR (a certain number of iterations)
    Statement(s)
ENDFOR

Example 1:

FOR (Counter =0; Counter <10; Counter =Counter +1)
  Prepare a new quiz
  Let the students do it
ENDFOR
Or
FOR (Counter =1:10)
  Prepare a new quiz
  Let the students do it
ENDFOR

Example 2:

FOR (counter=1:16)
  Come to the class
ENDFOR

T.C. Adnan Menderes University Faculty of Engineering

Basic Pseudocode Statements
The "FOR" Repetition Structure

- **The "FOR" repetition statements repeats** a statement or a group of statements for a certain number of times.
- It may have the following form.

FOR (a certain number of iterations)
    Statement(s)
ENDFOR

Example 1:

FOR (Counter =0; Counter <10; Counter =Counter +1)
  Prepare a new quiz
  Let the students do it
ENDFOR
Or
FOR (Counter =1:10)
  Prepare a new quiz
  Let the students do it
ENDFOR

Example 2:

FOR (counter=1:16)
  Come to the class
ENDFOR

Example 3:

FOR (counter=1:1)
  Violate the class rules
ENDFOR

T.C. Adnan Menderes University Faculty of Engineering

How to Think? Think Logically

If you want to solve a problem, you should ask yourself these questions:

- What is the **input** of the problem?
- What is the **output** of the problem?
- What are **processes** that could be applied on the input to get the output.
- Do I need **selection** & **repetition structures**? If yes, where and how?
- What are the **variables** that I may need to use?
  - Variable(s) for storing the **input**.
  - Variable(s) for storing the **output**.
  - Variable(s) for **counting** (if you have a repetition).
  - **Intermediate** variable(s) for the simplification of calculations.

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 1

Problem 1:

Write a pseudocode for a program that gets two integer numbers from the user, and then prints out the sum of those numbers.

Pseudocode:

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 1

Problem 1:

Write a pseudocode for a program that gets two integer numbers from the user, and then prints out the sum of those numbers.

Pseudocode:

Declare X1, X2, Sum

X1	?
X2	?
Sum	?

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 1

Problem 1:

Write a pseudocode for a program that gets two integer numbers from the user, and then prints out the sum of those numbers.

Pseudocode:

Declare X1, X2, Sum

Read X1,X2



http://www.danskebank.co.uk/en-gb/Business/Small-business/Online-services/Pages/business-ebanking.aspx

X1 Value 1

X2 Value 2

Sum ?

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 1

Problem 1:

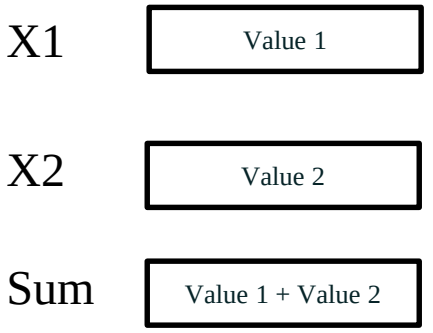
Write a pseudocode for a program that gets two integer numbers from the user, and then prints out the sum of those numbers.

Pseudocode:

Declare X1, X2, Sum

Read X1,X2

Sum = X1+X2



T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 1

Problem 1:

Write a pseudocode for a program that gets two integer numbers from the user, and then prints out the sum of those numbers.

Pseudocode:

Declare X1, X2, Sum

Read X1,X2

Sum = X1+X2

Write Sum



http://goo.gl/ArNUmV

X1	Value 1
X2	Value 2
Sum	Value 1 + Value 2

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 2

Problem 2:

Write a pseudocode for a program that computes and prints the average of any three numbers.

Pseudocode:

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 2

Problem 2:

Write a pseudocode for a program that computes and prints the average of any three numbers.

Pseudocode:

Declare X1, X2, X3, Sum, Average

Read X1,X2,X3

Sum = X1+X2+X3

Average=Sum/3

Write Average

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 3

Problem 3:

Write a pseudocode for a program that calculates the area of an rectangle.

Pseudocode:

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 3

Problem 3:

Write a pseudocode for a program that calculates the area of an rectangle.

Pseudocode:

Declare Length, Width, Area

Read Length, Width

Area= Length * Width

Write Area

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 4

Problem 4:

Write a pseudocode for a program that calculates and displays the square (second power) and cube (third power) of an integer number.

Pseudocode:

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 4

Problem 4:

Write a pseudocode for a program that calculates and displays the square (second power) and cube (third power) of an integer number.

Pseudocode:

Declare X, Square, Cube

Read X

Square = X* X

Cube= Square *X

Write Square, Cube

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Exercise 1

Exercise 1:

Write a pseudocode for a program that reads a temperature given in Fahrenheit, converts it into Celsius degree, and prints it.

Hint: use this formula c=(f-32)*5/9 to convert Fahrenheit into Celsius.

Pseudocode:

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Exercise 2

Exercise 2:

Write a pseudocode for a program that calculates the area and circumference of a circle.

Hints: area = PI * radius * radius; circumference = 2 * PI * radius.

Pseudocode:

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 5

Problem 5:

Write a pseudocode for a program that reads a student's grade, and then check if it is bigger than 49 it will print PASS otherwise it will print FAIL.

Pseudocode:

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 5

Problem 5:

Write a pseudocode for a program that reads a student's grade, and then check if it is bigger than 49 it will print PASS otherwise it will print FAIL.

Pseudocode:

Declare grade

Read grade

IF (grade > 49)

THEN

Write Pass

ELSE

Write Fail

ENDIF

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 6

Problem 6:

Write a pseudocode for a program that gets from the user 3 integer numbers, and then returns the maximum one between them.

Pseudocode:

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 6

Problem 6:

Write a pseudocode for a program that gets from the user 3 integer numbers, and then returns the maximum one between them.

Pseudocode:

Declare X1, X2, X3, Max

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 6

Problem 6:

Write a pseudocode for a program that gets from the user 3 integer numbers, and then returns the maximum one between them.

Pseudocode:

Declare X1, X2, X3, Max

Read X1,X2,X3

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 6

Problem 6:

Write a pseudocode for a program that gets from the user 3 integer numbers, and then returns the maximum one between them.

Pseudocode:

Declare X1, X2, X3, Max

Read X1,X2,X3

IF (X1> X2 & X1>X3)

THEN

ELSE

ENDIF

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 6

Problem 6:

Write a pseudocode for a program that gets from the user 3 integer numbers, and then returns the maximum one between them.

Pseudocode:

```
Declare  X1, X2, X3, Max
Read X1,X2,X3
IF (X1> X2 & X1>X3)
THEN
   Max=X1
ELSE


ENDIF
```

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 6

Problem 6:

Write a pseudocode for a program that gets from the user 3 integer numbers, and then returns the maximum one between them.

Pseudocode:

```
Declare  X1, X2, X3, Max
Read X1,X2,X3
IF (X1> X2 & X1>X3)
THEN
  Max=X1
ELSE
  IF (X2>X1 & X2>X3)
    THEN

    ELSE

  ENDIF
ENDIF
```

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 6

Problem 6:

Write a pseudocode for a program that gets from the user 3 integer numbers, and then returns the maximum one between them.

Pseudocode:

```
Declare  X1, X2, X3, Max
Read X1,X2,X3
IF (X1> X2 & X1>X3)
THEN
  Max=X1
ELSE
  IF (X2>X1 & X2>X3)
    THEN
      Max=X2
    ELSE

  ENDIF
ENDIF
```

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 6

Problem 6:

Write a pseudocode for a program that gets from the user 3 integer numbers, and then returns the maximum one between them.

Pseudocode:

```
Declare  X1, X2, X3, Max
Read X1,X2,X3
IF (X1> X2 & X1>X3)
THEN
  Max=X1
ELSE
  IF (X2>X1 & X2>X3)
    THEN
      Max=X2
    ELSE
     Max=X3
  ENDIF
ENDIF
```

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 6

Problem 6:

Write a pseudocode for a program that gets from the user 3 integer numbers, and then returns the maximum one between them.

Pseudocode:

```
Declare  X1, X2, X3, Max
Read X1,X2,X3
IF (X1> X2 & X1>X3)
THEN
  Max=X1
ELSE
  IF (X2>X1 & X2>X3)
    THEN
     Max=X2
    ELSE
     Max=X3
  ENDIF
ENDIF
Write MAX
```

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Exercise 3

Exercise 2:

Write a pseudocode for a program that calculates the roots of a quadratic equation $ax^2 + bx + c = 0$.

Hint: d = sqrt ($b^2 - 4ac$), and the roots are: X1= (–b + d)/2a and X2 = (–b – d)/2a.

Pseudocode:

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 7

Problem 7:

Write a pseudocode for a program that reads the grades of any five students, finds the average of these grades, and then prints it.

Pseudocode (using FOR):

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 7

Problem 7:

Write a pseudocode for a program that reads the grades of any five students, finds the average of these grades, and then prints it.

Pseudocode (using FOR):

Declare Counter, Grade,Average

Initialize TotalGrade=0

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 7

Problem 7:

Write a pseudocode for a program that reads the grades of any five students, finds the average of these grades, and then prints it.

Pseudocode (using FOR):

Declare Counter, Grade,Average

Initialize TotalGrade=0

FOR (specific number of iterations)

    Statement(s)

ENDFOR

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 7

Problem 7:

Write a pseudocode for a program that reads the grades of any five students, finds the average of these grades, and then prints it.

Pseudocode (using FOR):

Declare Counter, Grade,Average

Initialize TotalGrade=0

FOR (Counter = 1:5)

    Statement(s)

ENDFOR

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 7

Problem 7:

Write a pseudocode for a program that reads the grades of any five students, finds the average of these grades, and then prints it.

Pseudocode (using FOR):

Declare Counter, Grade,Average

Initialize TotalGrade=0

FOR (Counter = 1:5)

Read Grade

Total=Total +Grade

ENDFOR

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 7

Problem 7:

Write a pseudocode for a program that reads the grades of any five students, finds the average of these grades, and then prints it.

Pseudocode (using FOR):

```
Declare  Counter, Grade,Average
Initialize TotalGrade=0
FOR (Counter = 1:5)
   Read Grade
   Total=Total +Grade
ENDFOR
Average = Total/5
Write Average
```

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 7

Problem 7:

Write a pseudocode for a program that reads the grades of any five students, finds the average of these grades, and then prints it.

Pseudocode (using FOR):

Declare Counter, Grade,Average

Initialize TotalGrade=0

FOR (Counter = 1:5)

Read Grade

Total=Total +Grade

ENDFOR

Average = Total/5

Write Average

Pseudocode (using WHILE):

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 7

Problem 7:

Write a pseudocode for a program that reads the grades of any five students, finds the average of these grades, and then prints it.

Pseudocode (using FOR):

Declare Counter, Grade,Average

Initialize TotalGrade=0

FOR (Counter = 1:5)

Read Grade

Total=Total +Grade

ENDFOR

Average = Total/5

Write Average

Pseudocode (using WHILE):

Declare Grade,Average

Initialize TotalGrade=0, Counter=0

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 7

Problem 7:

Write a pseudocode for a program that reads the grades of any five students, finds the average of these grades, and then prints it.

Pseudocode (using FOR):

Declare Counter, Grade,Average

Initialize TotalGrade=0

FOR (Counter = 1:5)

Read Grade

Total=Total +Grade

ENDFOR

Average = Total/5

Write Average

Pseudocode (using WHILE):

Declare Grade,Average

Initialize TotalGrade=0, Counter=0

WHILE (Condition)

Statement(s)

ENDFOR

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 7

Problem 7:

Write a pseudocode for a program that reads the grades of any five students, finds the average of these grades, and then prints it.

Pseudocode (using FOR):

Declare Counter, Grade,Average

Initialize TotalGrade=0

FOR (Counter = 1:5)

Read Grade

Total=Total +Grade

ENDFOR

Average = Total/5

Write Average

Pseudocode (using WHILE):

Declare Grade,Average

Initialize TotalGrade=0, Counter=0

WHILE (Counter <5)

Statement(s)

ENDFOR

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 7

Problem 7:

Write a pseudocode for a program that reads the grades of any five students, finds the average of these grades, and then prints it.

Pseudocode (using FOR):

```
Declare  Counter, Grade,Average
Initialize TotalGrade=0
FOR (Counter = 1:5)
   Read Grade
   Total=Total +Grade
ENDFOR
Average = Total/5
Write Average
```

Pseudocode (using WHILE):

```
Declare  Grade,Average
Initialize TotalGrade=0, Counter=0
WHILE (Counter <5)
   Read Grade
   Total=Total +Grade
   Counter= Counter +1
ENDFOR
```

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 7

Problem 7:

Write a pseudocode for a program that reads the grades of any five students, finds the average of these grades, and then prints it.

Pseudocode (using FOR):

Declare Counter, Grade,Average

Initialize TotalGrade=0

FOR (Counter = 1:5)

Read Grade

Total=Total +Grade

ENDFOR

Average = Total/5

Write Average

Pseudocode (using WHILE):

Declare Grade,Average

Initialize TotalGrade=0, Counter=0

WHILE (Counter <5)

Read Grade

Total=Total +Grade

Counter= Counter +1

ENDFOR

Average = Total/5

Write Average

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 8

Problem 8:

Write a pseudocode for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user.

n! = n*(n-1)*(n-2)*…….*1

1!=1

2!=2*1

3!=3*2*1

4!=4*3*2*1

5!=5*4*3*2*1

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 8

Problem 8:

Write a pseudocode for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user.

n! = n*(n-1)*(n-2)*…….*1

1!=1

2!=2*1

3!=3*2*1

4!=4*3*2*1

5!=5*4*3*2*1

What are the input and output?

How many iterations?

What should we do in each iterations?

What are the variables that I may need?

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 8

Problem 8:

Write a pseudocode for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user.

n! = n*(n-1)*(n-2)*…….*1

What are the input and output?

How many iterations?

What should we do in each iterations?

What are the variables that I may need?

Pseudocode (using FOR):

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 8

Problem 8:

Write a pseudocode for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user.

n! = n*(n-1)*(n-2)*…….*1

Pseudocode (using FOR):

Declare N, Factorial=1, Temp, Counter

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 8

Problem 8:

Write a pseudocode for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user.

n! = n*(n-1)*(n-2)*…….*1

Pseudocode (using FOR):

Declare N, Factorial=1, Temp, Counter

Read N

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 8

Problem 8:

Write a pseudocode for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user.

n! = n*(n-1)*(n-2)*…….*1

Pseudocode (using FOR):

Declare N, Factorial=1, Temp, Counter

Read N

Temp=N

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 8

Problem 8:

Write a pseudocode for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user.

n! = n*(n-1)*(n-2)*…….*1

Pseudocode (using FOR):

Declare N, Factorial=1, Temp, Counter

Read N

Temp=N

FOR (specific number of iterations)

Statement(s)

ENDFOR

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 8

Problem 8:

Write a pseudocode for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user.

n! = n*(n-1)*(n-2)*…….*1

Pseudocode (using FOR):

Declare N, Factorial=1, Temp, Counter

Read N

Temp=N

FOR (Counter =1:N)

Statement(s)

ENDFOR

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 8

Problem 8:

Write a pseudocode for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user.

n! = n*(n-1)*(n-2)*…….*1

Pseudocode (using FOR):

```
Declare  N, Factorial=1, Temp, Counter
Read N
Temp=N
FOR (Counter =1:N)
  Factorial=Factorial*Temp
  Temp=Temp-1
ENDFOR
```

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 8

Problem 8:

Write a pseudocode for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user.

n! = n*(n-1)*(n-2)*…….*1

Pseudocode (using FOR):

```
Declare  N, Factorial=1, Temp, Counter
Read N
Temp=N
FOR (Counter =1:N)
  Factorial=Factorial*Temp
  Temp=Temp-1
ENDFOR
Write Factorial
```

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 8

Problem 8:

Write a pseudocode for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user.

n! = n*(n-1)*(n-2)*…….*1

Pseudocode (using FOR):

Declare N, Factorial=1, Temp, Counter

Read N

Temp=N

FOR (Counter =1:N)

Factorial=Factorial*Temp

Temp=Temp-1

ENDFOR

Write Factorial

Pseudocode (using WHILE):

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Problem 8

Problem 8:

Write a pseudocode for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user.

n! = n*(n-1)*(n-2)*…….*1

Pseudocode (using FOR):

```
Declare  N, Factorial=1, Temp, Counter
Read N
Temp=N
FOR (Counter =1:N)
  Factorial=Factorial*Temp
  Temp=Temp-1
ENDFOR
Write Factorial
```

Pseudocode (using WHILE):

```
Declare N, Factorial=1, Counter=1
Read N
Temp=N
WHILE (Counter <=N)
  Factorial=Factorial*Temp
  Temp=Temp-1
  Counter= Counter +1
ENDFOR
Write Factorial
```

T.C. Adnan Menderes University Faculty of Engineering

Pseudocode – Exercise 4

- Write a pseudocode for a program that accepts two nonnegative integer numbers A and B, divides A by B (provided B is not zero), and reports the quotient and remainder, taking in consideration that you can not use the division operation (/), and you can use only subtraction (-) and addition (+).
- Examples:
  - Input: A=11,B=3. Output: quotient =3, remainder=2.
  - Input A=6, B=2. Output: quotient =2, remainder= 0.
  - Input A=4, B=6. Output: quotient =0, remainder= 4.

T.C. Adnan Menderes University Faculty of Engineering

Flowcharts

- A Flowchart is a graphical representation composed of shapes, lines, and text that graphically illustrates the steps of an algorithm.

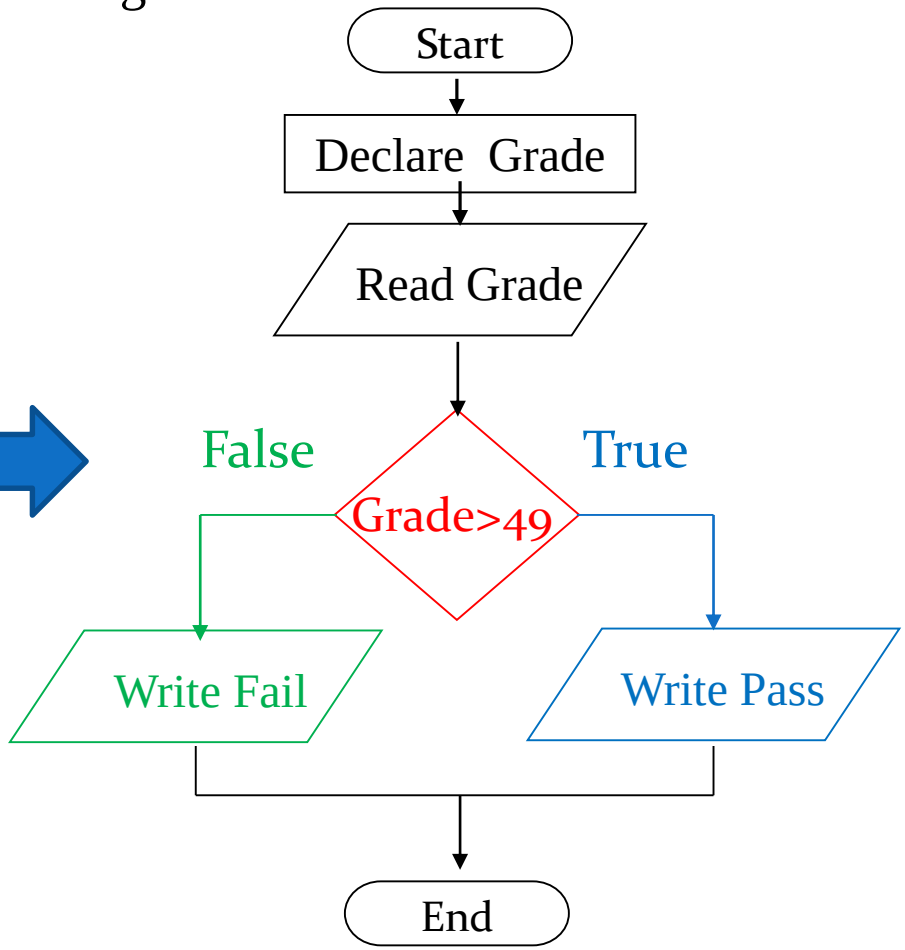
T.C. Adnan Menderes University Faculty of Engineering

Flowcharts

- A Flowchart is a graphical representation composed of shapes, lines, and text that graphically illustrates the steps of an algorithm.
- Example:

```
Declare Grade
Read Grade
IF (Grade> 49)
   THEN
    Write Pass
   ELSE
    Write Fail
ENDIF
```

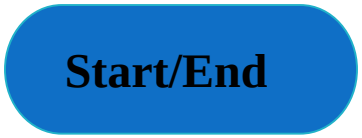




T.C. Adnan Menderes University Faculty of Engineering

Flowchart Building Units

Oval



The start/end symbol :represents the start or the end of an algorithm.

T.C. Adnan Menderes University Faculty of Engineering

Flowchart Building Units

Arrow

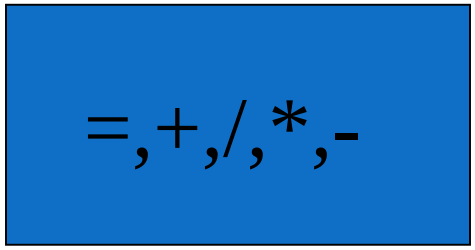


The sequence symbol: represents (flow of control) the sequence of steps in an algorithms

T.C. Adnan Menderes University Faculty of Engineering

Flowchart Building Units

Rectangle



1. Read
2. Write
3. **Calculate**
4. **Assign**
5. Select
6. Repeat

The process symbol represents:

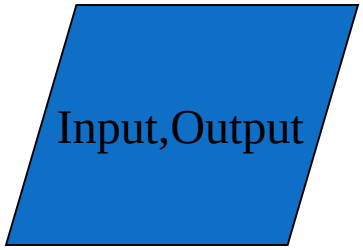
- Calculating (or performing) arithmetic operation on data like +, -, *, /.
- Or, Assigning a value to a variable (fill a container).

.

T.C. Adnan Menderes University Faculty of Engineering

Flowchart Building Units

Parallelograms



1. **Read**
2. **Write**
3. ~~**Calculate**~~
4. ~~**Assign**~~
5. Select
6. Repeat

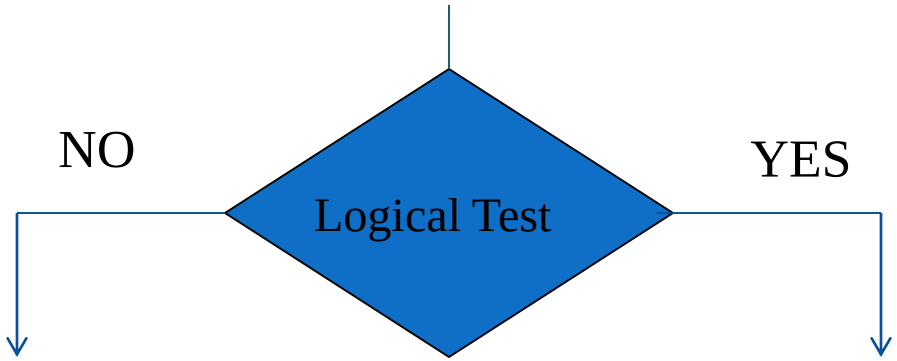
The input (read)/output (write) symbol represents:

- Reading (or getting) data from an input device (keyboard, file, etc.).
- Writing (or displaying) data to an output device (monitor, file, printer, etc.).

T.C. Adnan Menderes University Faculty of Engineering

Flowchart Building Units

Diamond



1. ~~**Read**~~
2. ~~**Write**~~
3. ~~**Calculate**~~
4. ~~**Assign**~~
5. **Select**
6. Repeat

The conditional (or **decision**) **symbol**: is the symbol that is equivalent to the **logical condition** in If /then/ else statement in pseudocode.

T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

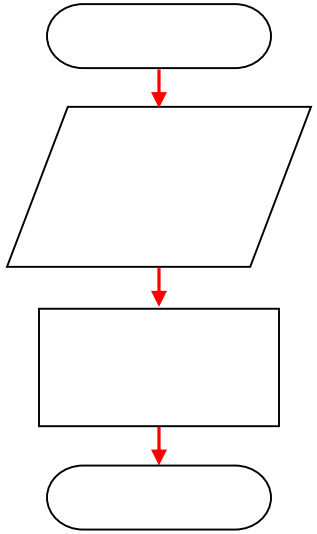
There three basic types of control structures in flowcharts:

1. The sequence structure.
2. The selection structure.
3. The repetition structure.

T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

1-The sequence structure is a series of actions are performed in a specific sequence indicated by the arrows symbols.

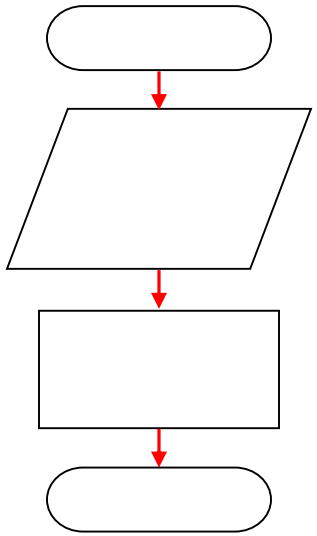


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

1-The sequence structure is a series of actions are performed in a specific sequence indicated by the arrows symbols.

Problem1:
Draw a flowchart for a program that gets two integer numbers from the user, and then print out the sum of those numbers.

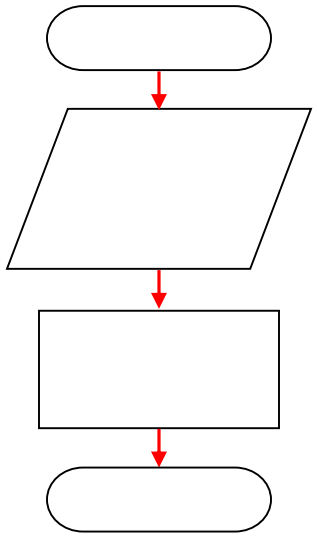


Pseudocode:
Declare X1, X2, Sum
Read X1,X2
Sum = X1+X2
Write Sum

T.C. Adnan Menderes University Faculty of Engineering

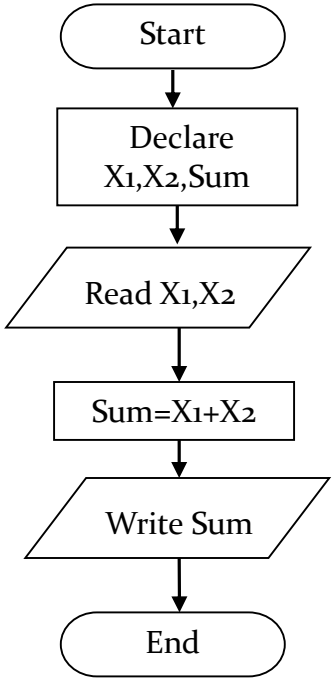
Basic Flowchart Structures

1-The sequence structure is a series of actions are performed in a specific sequence indicated by the arrows symbols.



Problem1:
Draw a flowchart for a program that gets two integer numbers from the user, and then print out the sum of those numbers.

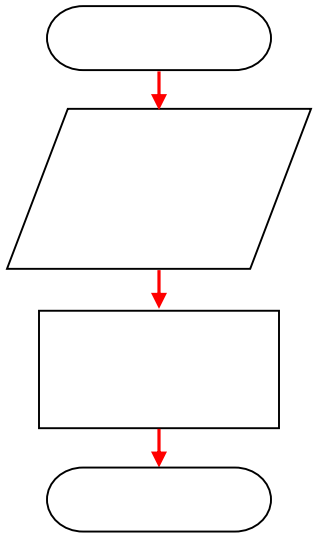
Pseudocode:
Declare X1, X2, Sum
Read X1,X2
Sum = X1+X2
Write Sum



T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

1-The sequence structure is a series of actions are performed in a specific sequence indicated by the arrows symbols.



Problem 2:
Draw a flowchart for a program that computes and prints the average of any three numbers.

Pseudocode:

Declare X1, X2, X3
Sum, Average
Read X1,X2,X3
Sum = X1+X2+X3
Average=Sum/3
Write Average

T.C. Adnan Menderes University Faculty of Engineering

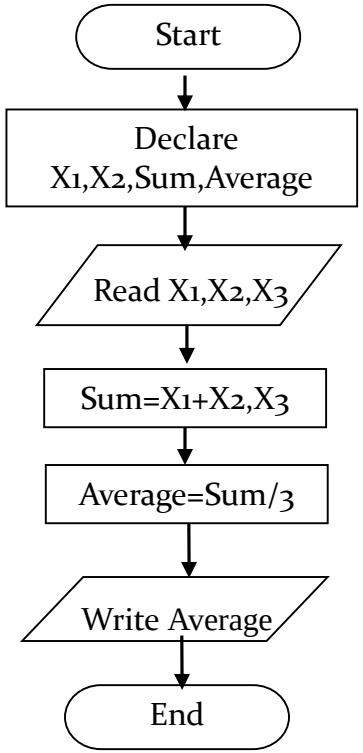
Basic Flowchart Structures

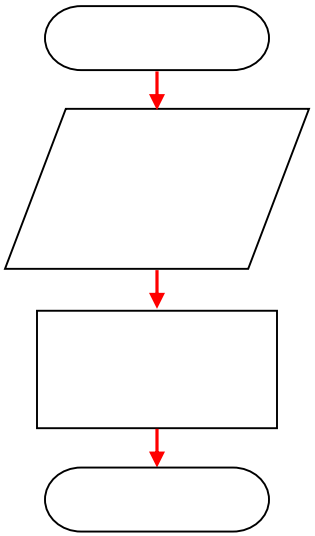
1-The sequence structure is a series of actions are performed in a specific sequence indicated by the arrows symbols.

Problem 2:
Draw a flowchart for a program that computes and prints the average of any three numbers.

Pseudocode:

Declare X1, X2, X3
Sum, Average
Read X1,X2,X3
Sum = X1+X2+X3
Average=Sum/3
Write Average



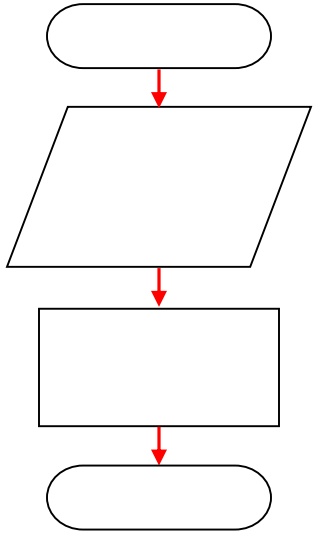


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

1-The sequence structure is a series of actions are performed in a specific sequence indicated by the arrows symbols.

Problem 3:
Draw a flowchart for a program that calculates the area of an rectangle.



Pseudocode:

Declare Length, Width, Area
Read Length, Width
Area= Length * Width
Write Area

T.C. Adnan Menderes University Faculty of Engineering

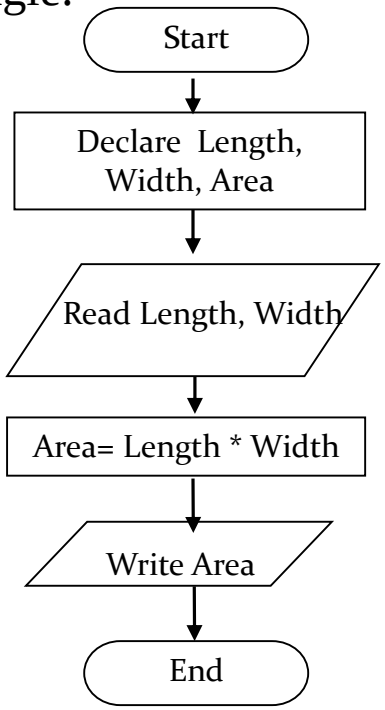
Basic Flowchart Structures

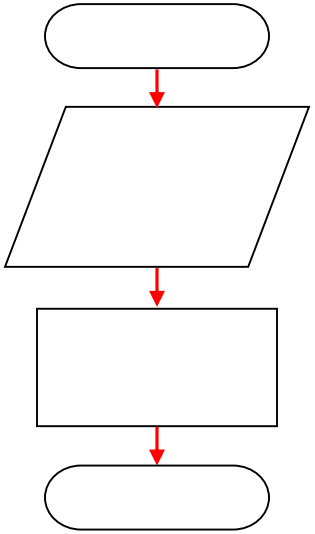
1-The sequence structure is a series of actions are performed in a specific sequence indicated by the arrows symbols.

Problem 3:
Draw a flowchart for a program that calculates the area of an rectangle.

Pseudocode:

Declare Length, Width, Area
Read Length, Width
Area= Length * Width
Write Area

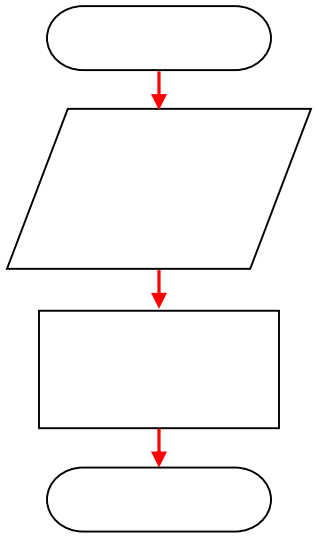




T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

1-The sequence structure is a series of actions are performed in a specific sequence indicated by the arrows symbols.



Problem 4:

Draw a flowchart for a program that calculates and displays the square (second power) and cube (third power) of an integer number.

Pseudocode:

Declare X, Square, Cube
Read X
Square = X* X
Cube= Square *X
Write Square, Cube

T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

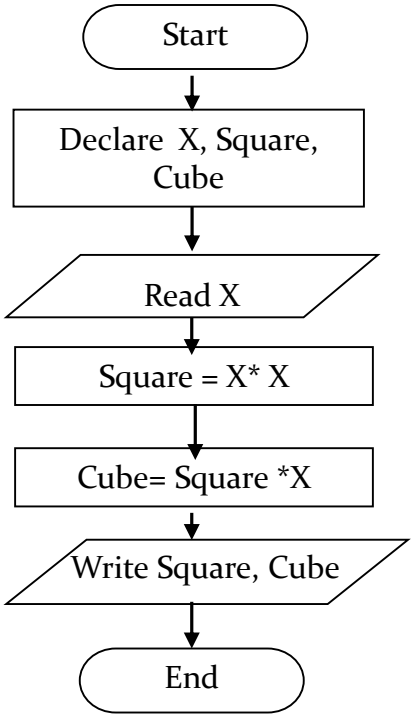
1-The sequence structure is a series of actions are performed in a specific sequence indicated by the arrows symbols.

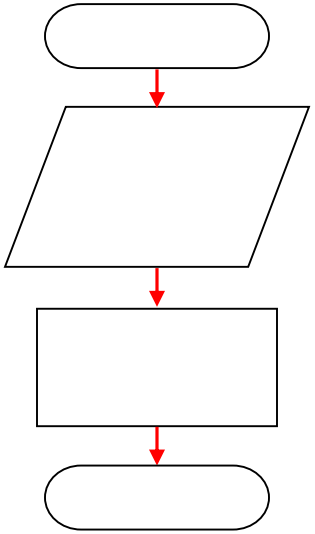
Problem 4:

Draw a flowchart for a program that calculates and displays the square (second power) and cube (third power) of an integer number.

Pseudocode:

Declare X, Square, Cube
Read X
Square = X* X
Cube= Square *X
Write Square, Cube

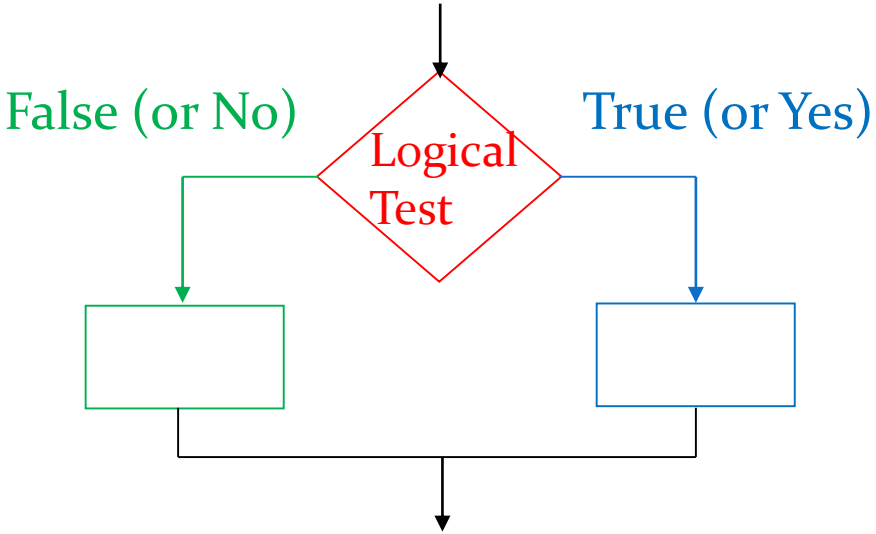




T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

2-**The decision (or conditional) structure** is equivalent to the if/then/else statement in pseudocode and the if function in Excel.



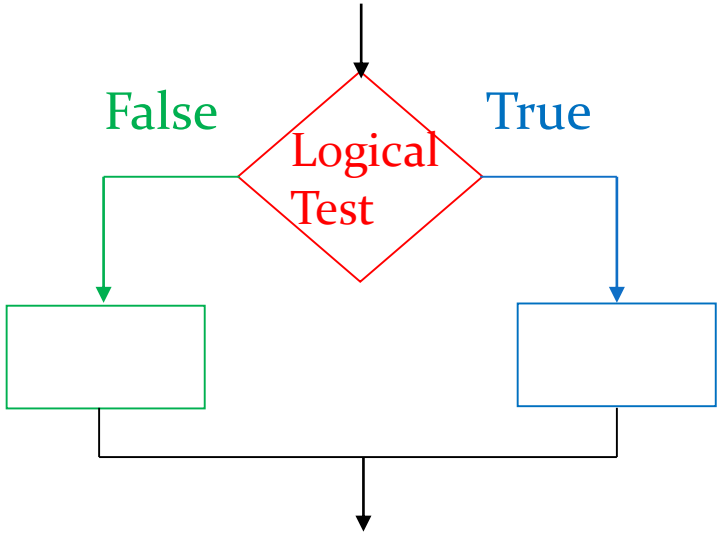
T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

2-**The decision (or conditional) structure** is equivalent to the if/then/else statement in pseudocode and the if function in Excel.

Problem 5:
Draw a flowchart for a program that reads a student's grade, tests it if PASS (more than 49) or FAIL, and prints the result.

Pseudocode:
Declare Grade
Read Grade
IF (Grade > 49)
THEN
Write Pass
ELSE
Write Fail
ENDIF

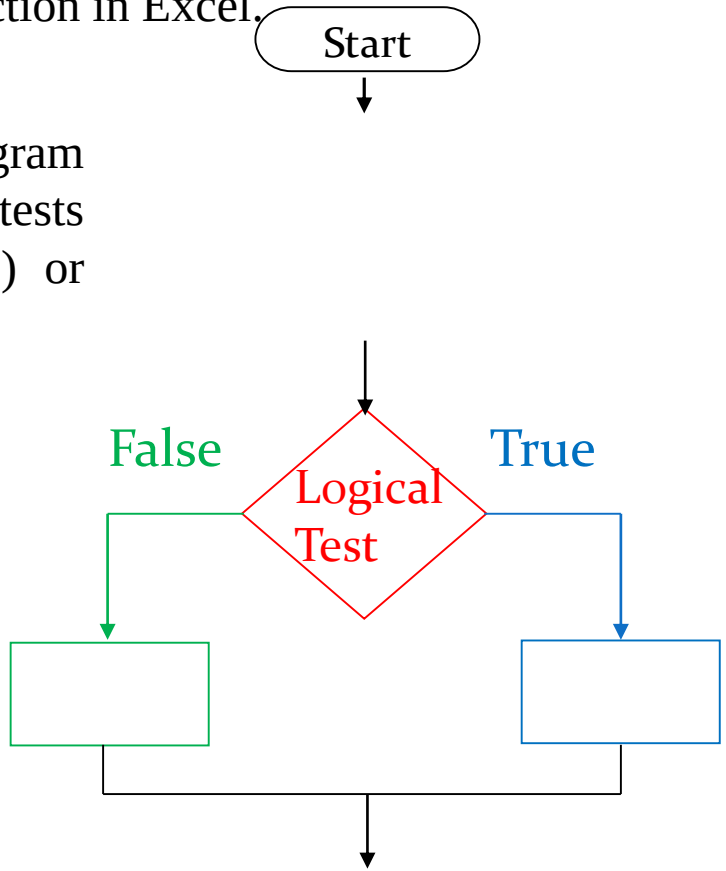


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

2-**The decision (or conditional) structure** is equivalent to the if/then/else statement in pseudocode and the if function in Excel.

Problem 5:
Draw a flowchart for a program that reads a student's grade, tests it if PASS (more than 49) or FAIL, and prints the result.



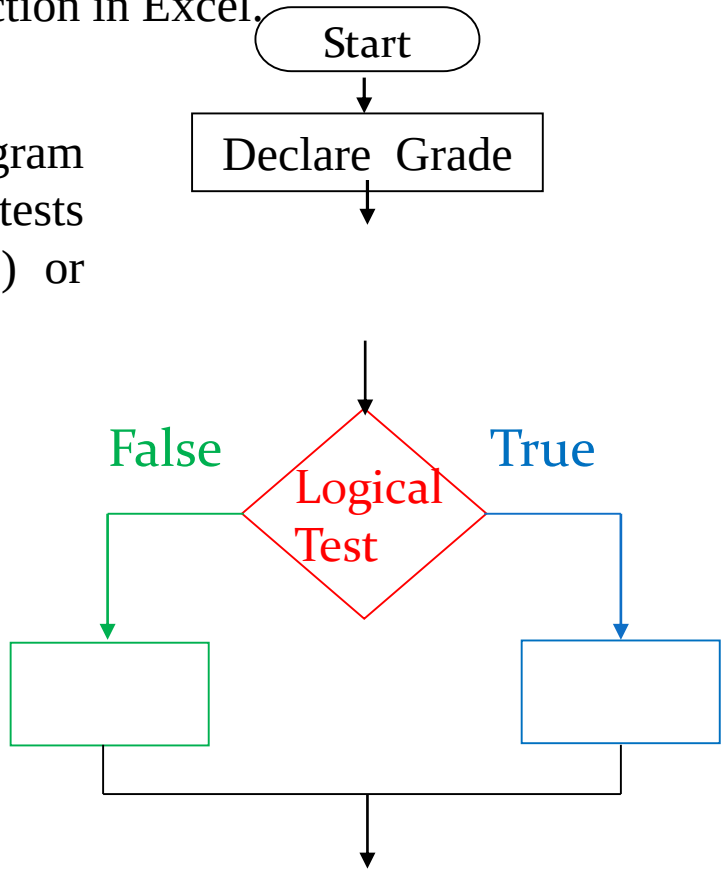
Pseudocode:
Declare Grade
Read Grade
IF (Grade > 49)
THEN
Write Pass
ELSE
Write Fail
ENDIF

T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

2-**The decision (or conditional) structure** is equivalent to the if/then/else statement in pseudocode and the if function in Excel.

Problem 5:
Draw a flowchart for a program that reads a student's grade, tests it if PASS (more than 49) or FAIL, and prints the result.



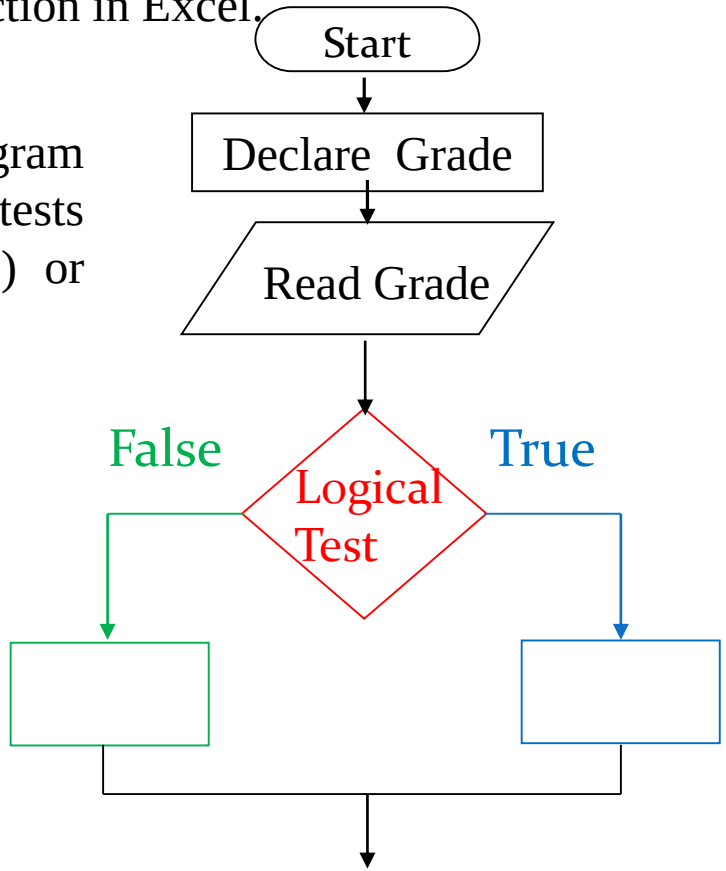
Pseudocode:
Declare Grade
Read Grade
IF (Grade > 49)
THEN
Write Pass
ELSE
Write Fail
ENDIF

T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

2-**The decision (or conditional) structure** is equivalent to the if/then/else statement in pseudocode and the if function in Excel.

Problem 5:
Draw a flowchart for a program that reads a student's grade, tests it if PASS (more than 49) or FAIL, and prints the result.



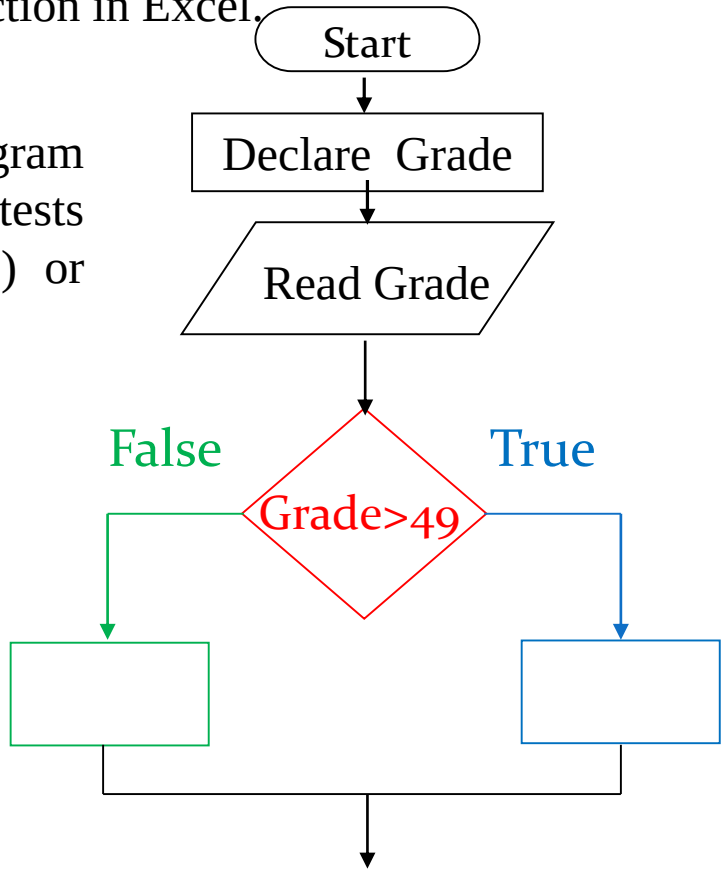
Pseudocode:
Declare Grade
Read Grade
IF (Grade > 49)
THEN
Write Pass
ELSE
Write Fail
ENDIF

T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

2-**The decision (or conditional) structure** is equivalent to the if/then/else statement in pseudocode and the if function in Excel.

Problem 5:
Draw a flowchart for a program that reads a student's grade, tests it if PASS (more than 49) or FAIL, and prints the result.



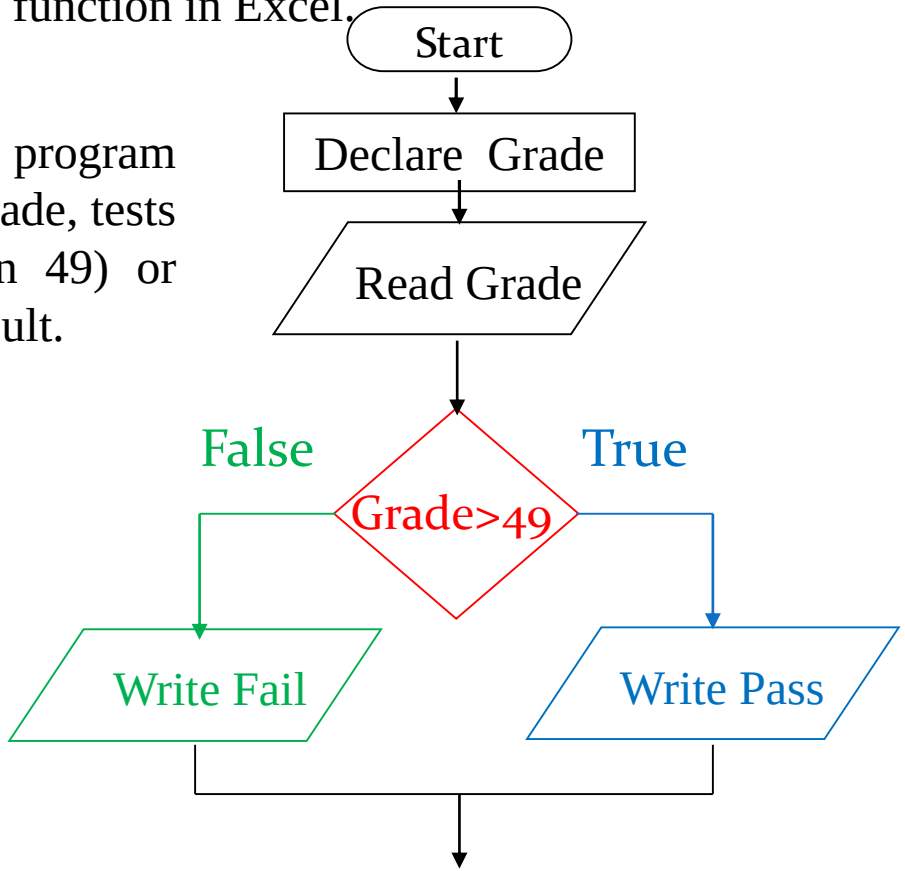
Pseudocode:
Declare Grade
Read Grade
IF (Grade > 49)
THEN
Write Pass
ELSE
Write Fail
ENDIF

T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

2-**The decision (or conditional) structure** is equivalent to the if/then/else statement in pseudocode and the if function in Excel.

Problem 5:
Draw a flowchart for a program that reads a student's grade, tests it if PASS (more than 49) or FAIL, and prints the result.



Pseudocode:
Declare Grade
Read Grade
IF (Grade > 49)
THEN
Write Pass
ELSE
Write Fail
ENDIF

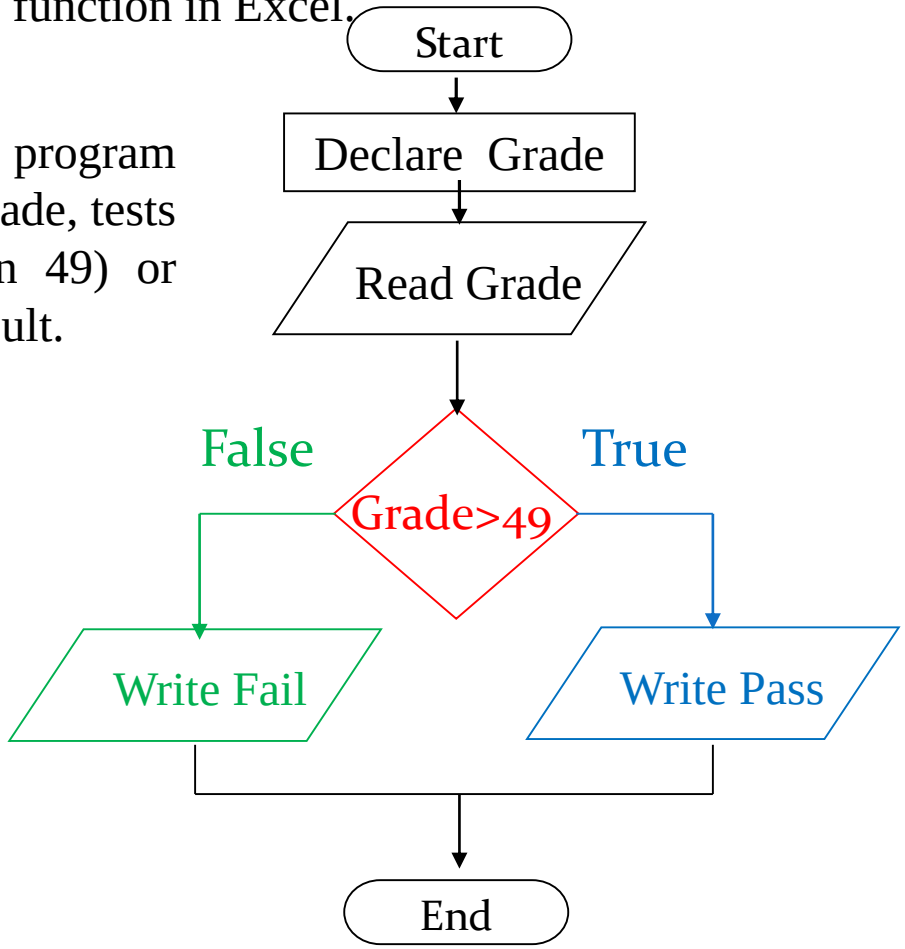
T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

2-**The decision (or conditional) structure** is equivalent to the if/then/else statement in pseudocode and the if function in Excel.

Problem 5:
Draw a flowchart for a program that reads a student's grade, tests it if PASS (more than 49) or FAIL, and prints the result.

Pseudocode:
Declare Grade
Read Grade
IF (Grade > 49)
THEN
Write Pass
ELSE
Write Fail
ENDIF



T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

2-**The decision (or conditional) structure** is equivalent to the if/then/else statement in pseudocode and the if function in excel.

Problem 6 :

Draw a flowchart for a program that gets from the user 3 integer numbers, and then returns the maximum one between them.

Pseudocode:

```
Declare  X1, X2, X3, Max
Read X1,X2,X3
IF (X1> X2)
 THEN
   IF (X1>X3)
    THEN
      Max=X1
    ELSE
      Max=X3
  END
ELSE
   IF (X2>X3)
     THEN
       Max=X2
     ELSE
       Max=X3
     ENDIF
ENDIF
Write MAX
```

T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

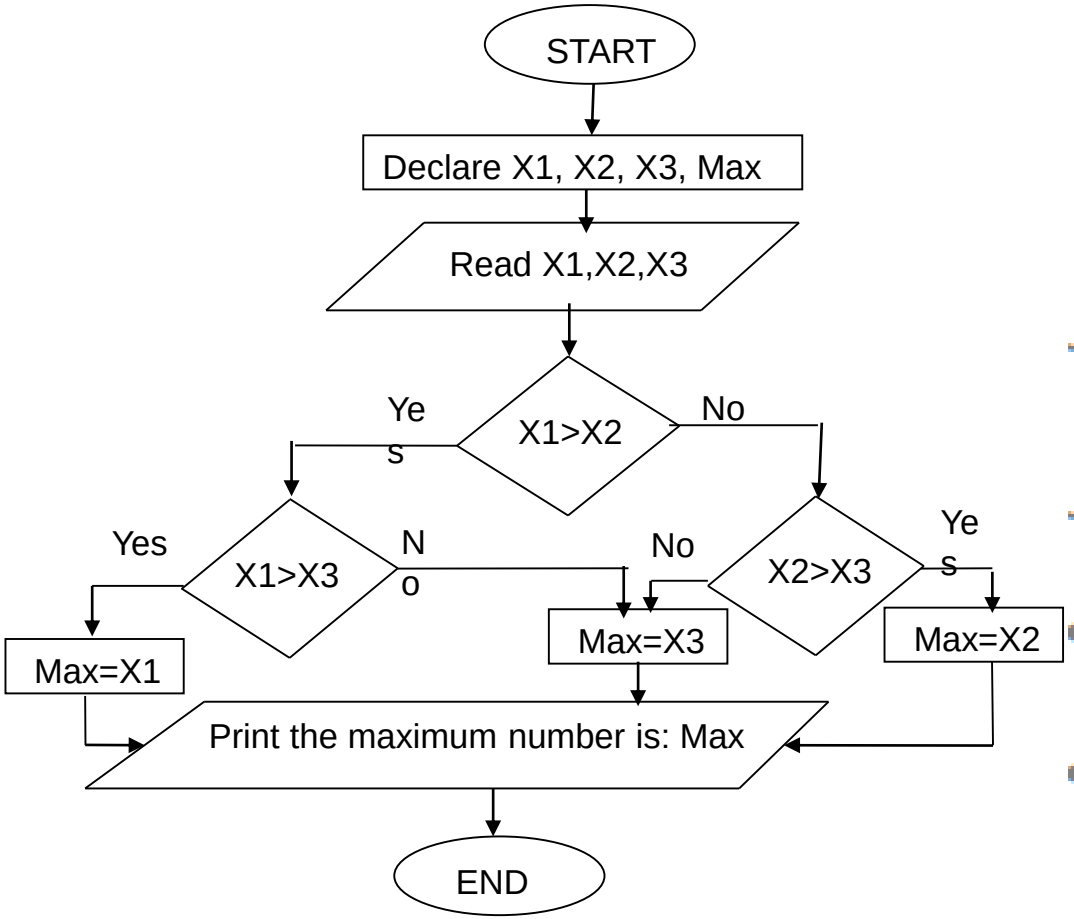
2-**The decision (or conditional) structure** is equivalent to the if/then/else statement in pseudocode and the if function in excel.

Problem 6 :

Draw a flowchart for a program that gets from the user 3 integer numbers, and then returns the maximum one between them.

Pseudocode:

```
Declare  X1, X2, X3, Max
Read X1,X2,X3
IF (X1> X2)
 THEN
   IF (X1>X3)
    THEN
      Max=X1
    ELSE
      Max=X3
  END
ELSE
   IF (X2>X3)
    THEN
      Max=X2
    ELSE
      Max=X3
    ENDIF
ENDIF
Write MAX
```



T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

3-**A repetition structure** represents a part of an algorithm that repeats. This type of structure is also known as a **loop** and has 3 main forms: **FOR**, **WHILE**, and **DO WHILE** structures.

In this course we will focus only on the **for structure,** because as soon as you understand it, you can easily understand the other two structures.

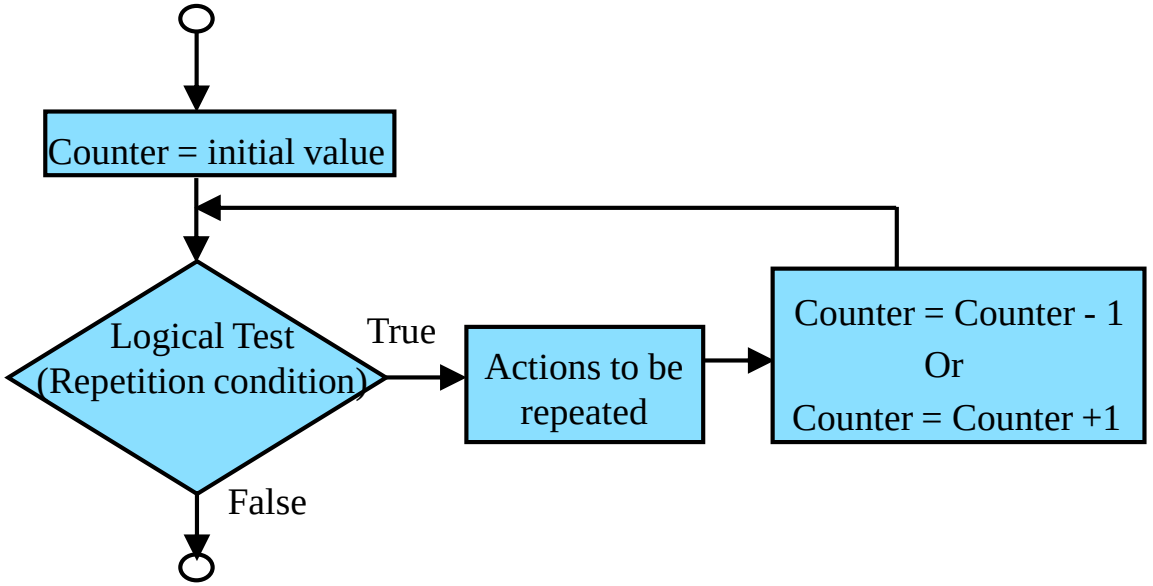
T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

When you want to construct a **FOR structure**, you should find the answers for these questions:

1. What is the maximum number of repetition? Let us keep it in a vaiable named **MaxR**.
2. What is the initial value of the counter? Let us keep it in a vaiable named **Counter**.
3. What is the logical condition (the condition of repetition)?
4. Shoud I increment or drcrement the **Counter** after each iteration? (we have **MaxR** iterations)
5. What are the actions that should be repeated?

~~1. Read~~
~~2. Write~~
~~3. Calculate~~
~~4. Assign~~
~~5. Select~~
6. **Repeat**



T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 7:

Draw a flowchart for a program that reads the grades of five students, finds the average of these grades, and then prints it.

Questions to answer before you start:

1. What is the maximum number of repetition?

T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 7:

Draw a flowchart for a program that reads the grades of five students, finds the average of these grades, and then prints it.

Questions to answer before you start:

1. What is the maximum number of repetition?

   MaxR= 5

2. What is the initial value of the counter?

T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 7:

Draw a flowchart for a program that reads the grades of five students, finds the average of these grades, and then prints it.

Questions to answer before you start:

1. What is the maximum number of repetition?

   MaxR= 5

2. What is the initial value of the counter?

   Counter = 0

3. What is the condition of repetition?

T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 7:

Draw a flowchart for a program that reads the grades of five students, finds the average of these grades, and then prints it.

Questions to answer before you start:

1. What is the maximum number of repetition?

   MaxR= 5

2. What is the initial value of the counter?

   Counter = 0

3. What is the condition of repetition?

   Counter < MaxR

4. Shoud I increment or decrement the **Counter** after each iteration?

T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 7:

Draw a flowchart for a program that reads the grades of five students, finds the average of these grades, and then prints it.

Questions to answer before you start:

1. What is the maximum number of repetition?

   MaxR= 5

2. What is the initial value of the counter?

   Counter = 0

3. What is the condition of repetition?

   Counter < MaxR

4. Shoud I increment or decrement the **Counter** after each iteration?

   Counter = Counter+1

5. What are the actions that should be repeated?

T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 7:

Draw a flowchart for a program that reads the grades of five students, finds the average of these grades, and then prints it.

Questions to answer before you start:

1. What is the maximum number of repetition?

   MaxR= 5

2. What is the initial value of the counter?

   Counter = 0

3. What is the condition of repetition?

   Counter < MaxR

4. Shoud I increment or decrement the **Counter** after each iteration?

   Counter = Counter+1

5. What are the actions that should be repeated?

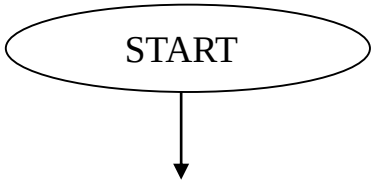
   Read a grade, add a grade, and increment the Counter

T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 7:

Draw a flowchart for a program that reads the grades of five students, finds the average of these grades, and then prints it.



Questions to answer before you start:

1. What is the maximum number of repetition?

   MaxR= 5

2. What is the initial value of the counter?

   Counter = 0

3. What is the condition of repetition?

   Counter < MaxR

4. Shoud I increment or decrement the **Counter** after each iteration?

   Counter = Counter+1

5. What are the actions that should be repeated?

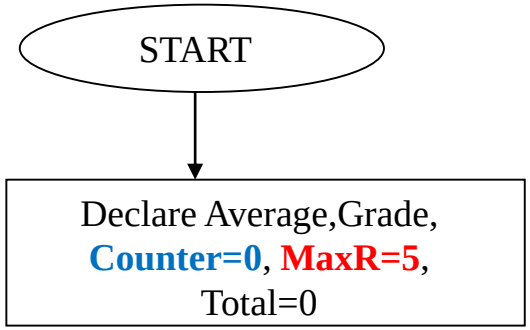
   Read a grade, add a grade, and increment the Counter

T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 7:

Draw a flowchart for a program that reads the grades of five students, finds the average of these grades, and then prints it.



Questions to answer before you start:

1. What is the maximum number of repetition?

   MaxR= 5

2. What is the initial value of the counter?

   Counter = 0

3. What is the condition of repetition?

   Counter < MaxR

4. Shoud I increment or decrement the **Counter** after each iteration?

   Counter = Counter+1

5. What are the actions that should be repeated?

   Read a grade, add a grade, and increment the Counter

T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 7:

Draw a flowchart for a program that reads the grades of five students, finds the average of these grades, and then prints it.

Questions to answer before you start:

1. What is the maximum number of repetition?

   MaxR= 5

2. What is the initial value of the counter?

   Counter = 0

3. What is the condition of repetition?

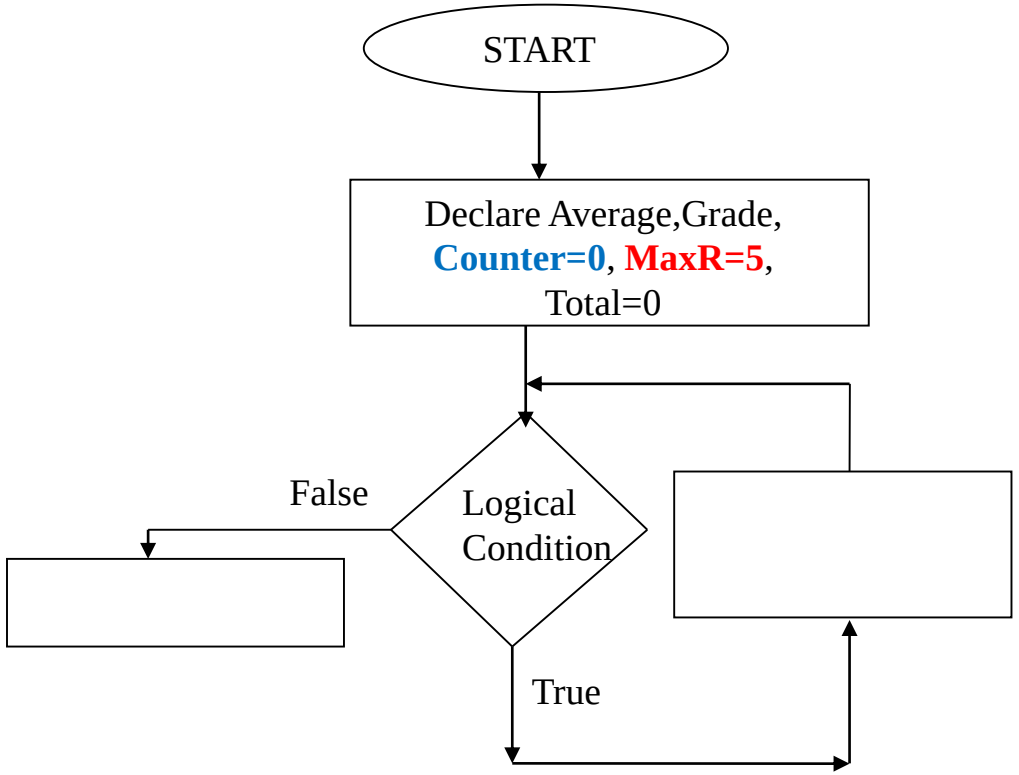
   Counter < MaxR

4. Shoud I increment or decrement the **Counter** after each iteration?

   Counter = Counter+1

5. What are the actions that should be repeated?

   Read a grade, add a grade, and increment the Counter



T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 7:

Draw a flowchart for a program that reads the grades of five students, finds the average of these grades, and then prints it.

Questions to answer before you start:

1. What is the maximum number of repetition?

   MaxR= 5

2. What is the initial value of the counter?

   Counter = 0

3. What is the condition of repetition?

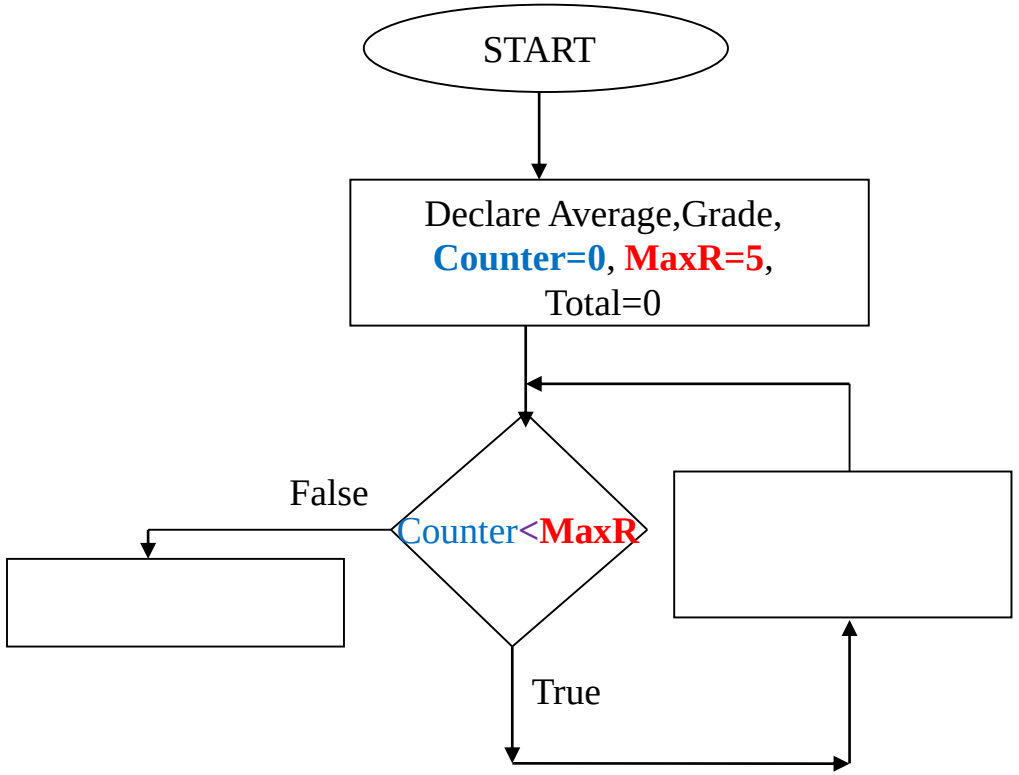
   Counter < MaxR

4. Shoud I increment or decrement the **Counter** after each iteration?

   Counter = Counter+1

5. What are the actions that should be repeated?

   Read a grade, add a grade, and increment the Counter



T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 7:

Draw a flowchart for a program that reads the grades of five students, finds the average of these grades, and then prints it.

Questions to answer before you start:

1. What is the maximum number of repetition?

   MaxR= 5

2. What is the initial value of the counter?

   Counter = 0

3. What is the condition of repetition?

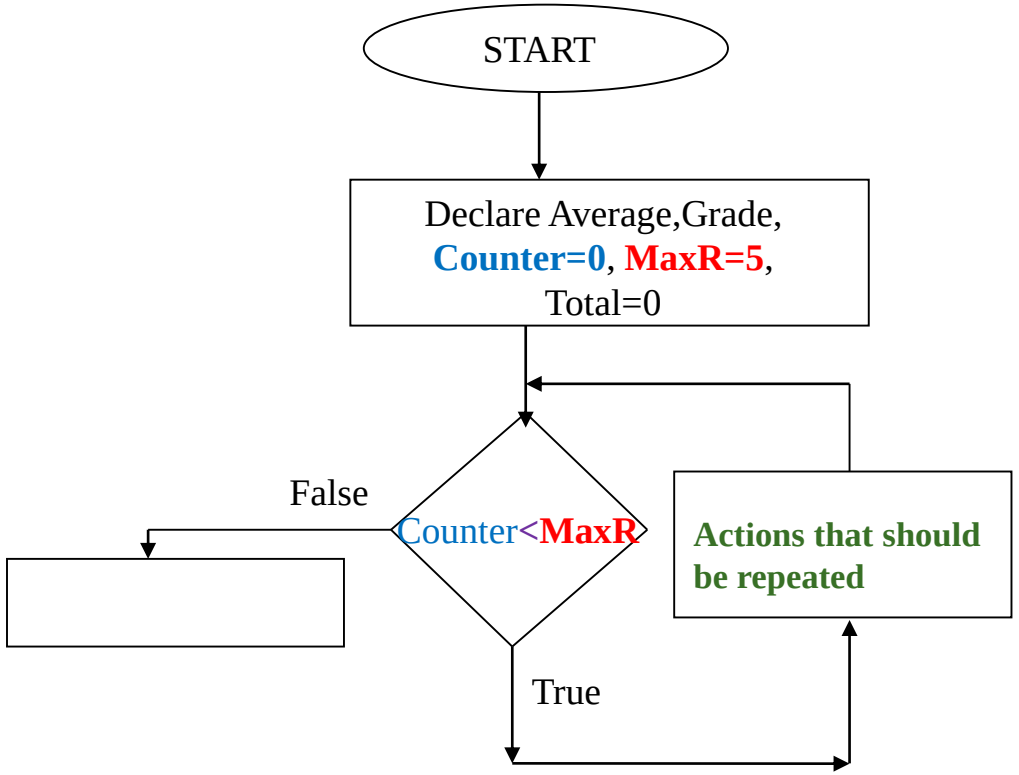
   Counter < MaxR

4. Shoud I increment or decrement the **Counter** after each iteration?

   Counter = Counter+1

5. What are the actions that should be repeated?

   Read a grade, add a grade, and increment the Counter



T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 7:

Draw a flowchart for a program that reads the grades of five students, finds the average of these grades, and then prints it.

Questions to answer before you start:

1. What is the maximum number of repetition?

   MaxR= 5

2. What is the initial value of the counter?

   Counter = 0

3. What is the condition of repetition?

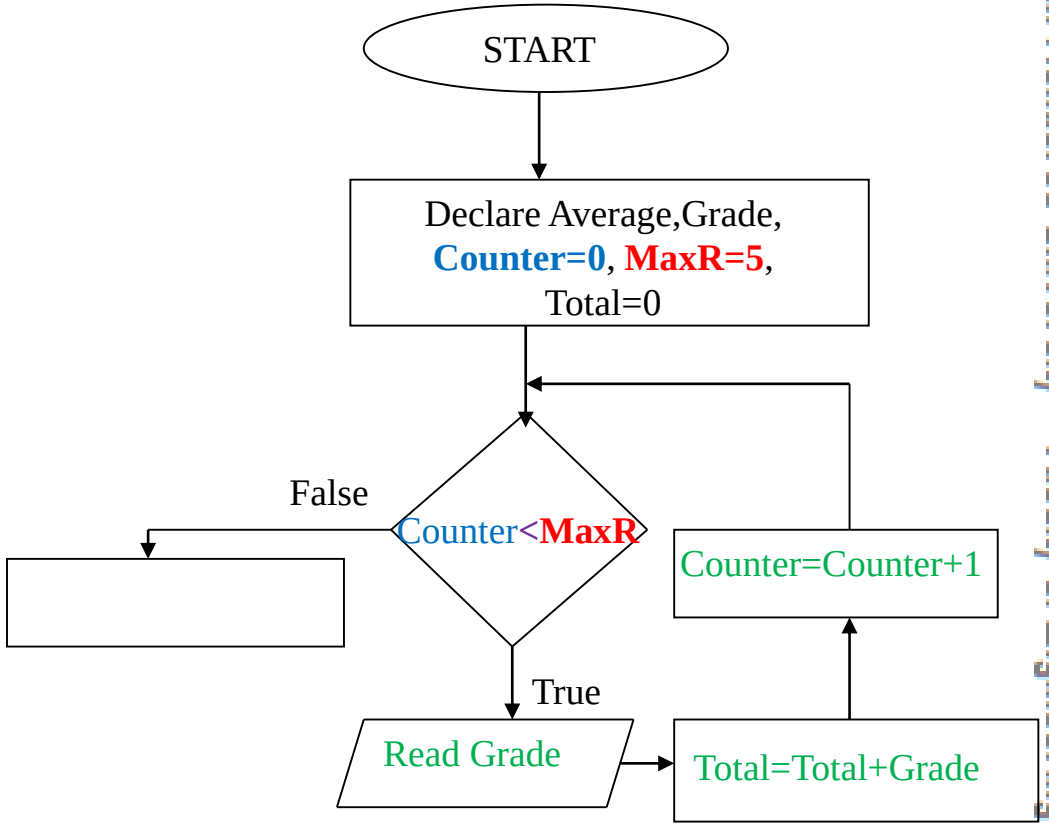
   Counter < MaxR

4. Shoud I increment or decrement the **Counter** after each iteration?

   Counter = Counter+1

5. What are the actions that should be repeated?

   Read a grade, add a grade, and increment the Counter



T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 7:

Draw a flowchart for a program that reads the grades of five students, finds the average of these grades, and then prints it.

Questions to answer before you start:

1. What is the maximum number of repetition?

   MaxR= 5

2. What is the initial value of the counter?

   Counter = 0

3. What is the condition of repetition?

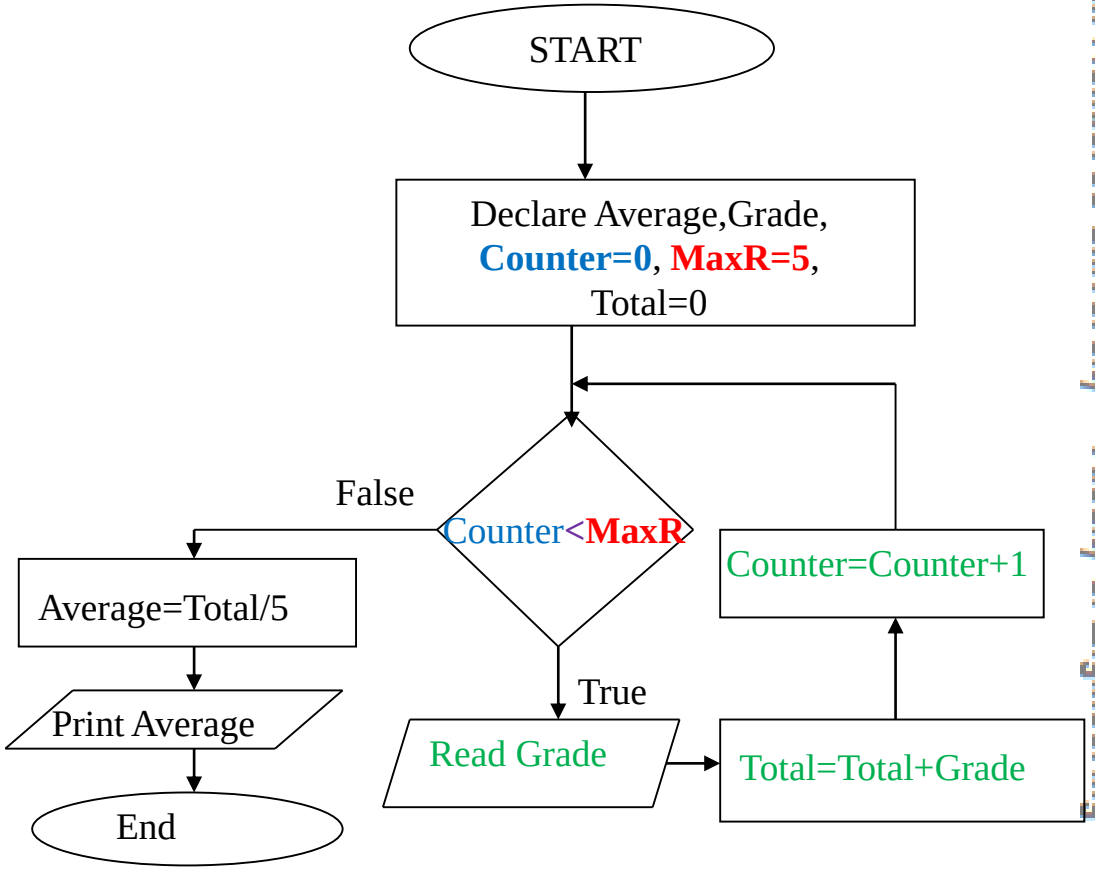
   Counter < MaxR

4. Shoud I increment or decrement the **Counter** after each iteration?

   Counter = Counter+1

5. What are the actions that should be repeated?

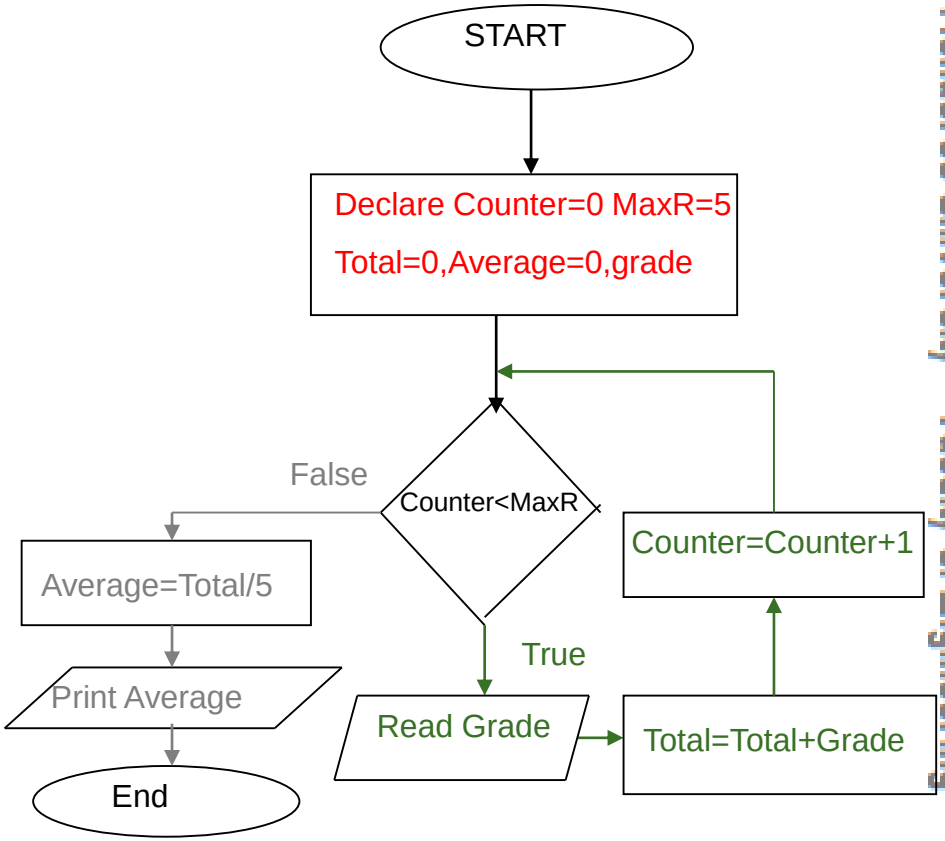
   Read a grade, add a grade, and increment the Counter



T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

A tracing table of the flowchart for the problem 7, assuming that the user has entered the following numbers, 40,30,100,70,60.

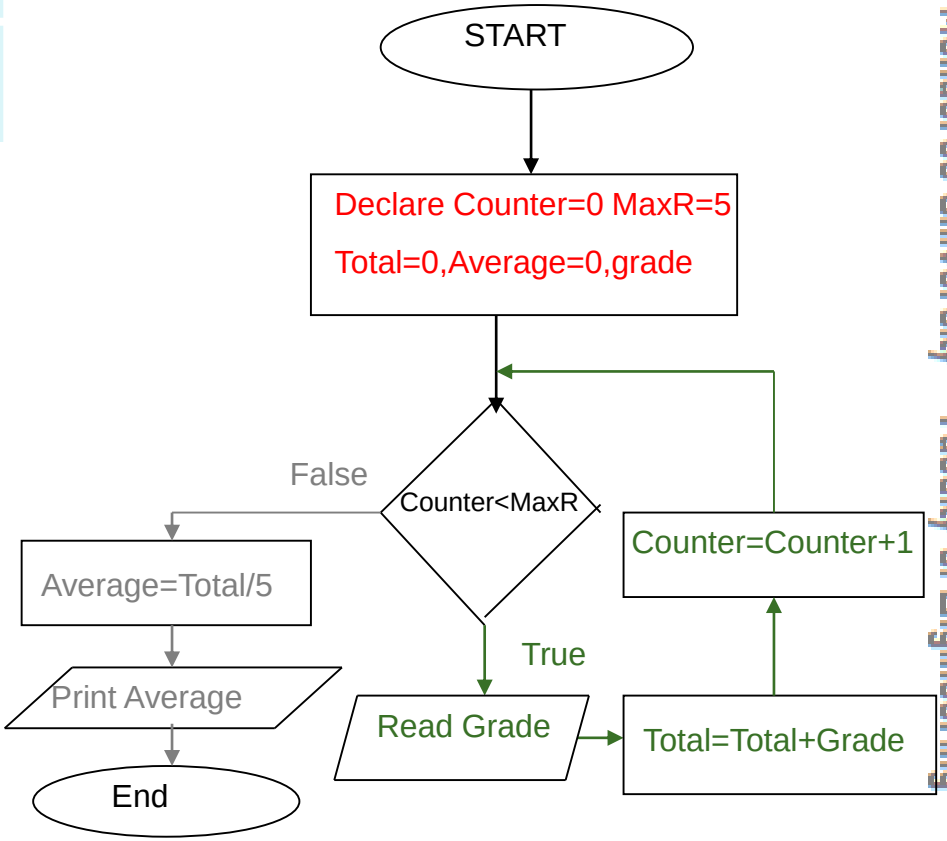


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

A tracing table of the flowchart for the problem 7, assuming that the user has entered the following numbers, 40,30,100,70,60.

	Counter <5	Grade	Total	Average	Counter
Declaration & Initialization:	?	?	0	0	0

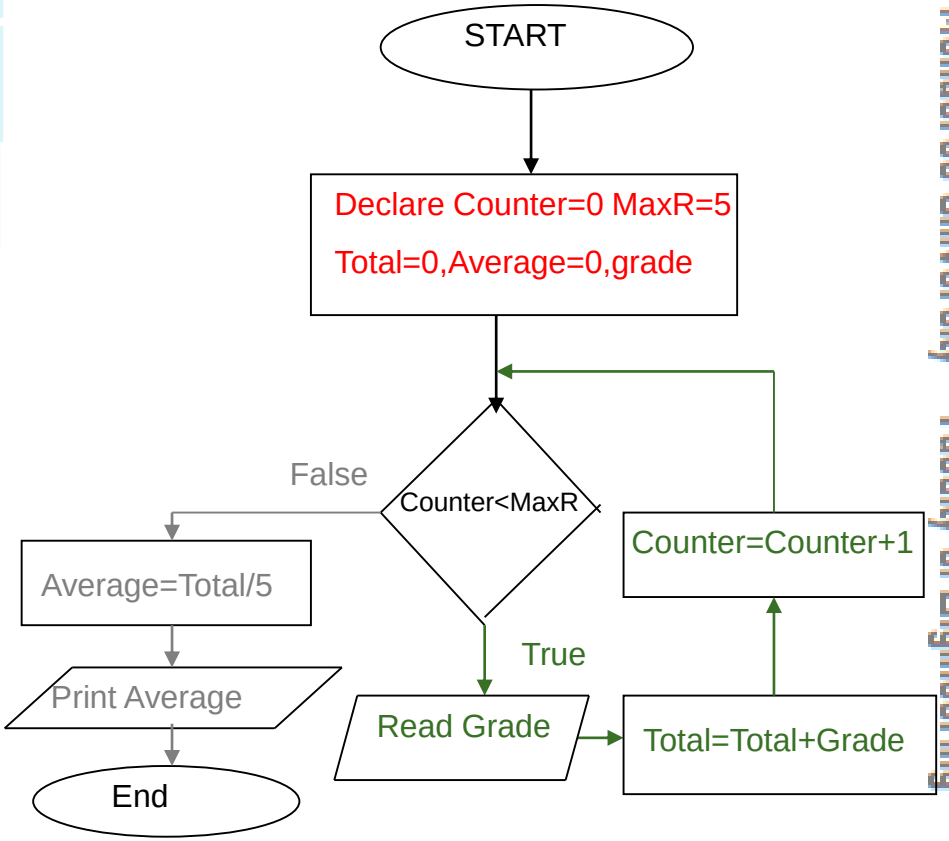


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

A tracing table of the flowchart for the problem 7, assuming that the user has entered the following numbers, 40,30,100,70,60.

	Counter <5	Grade	Total	Average	Counter
Declaration & Initialization:	?	?	0	0	0
Itr 1 of repetition	True	40	40	0	1

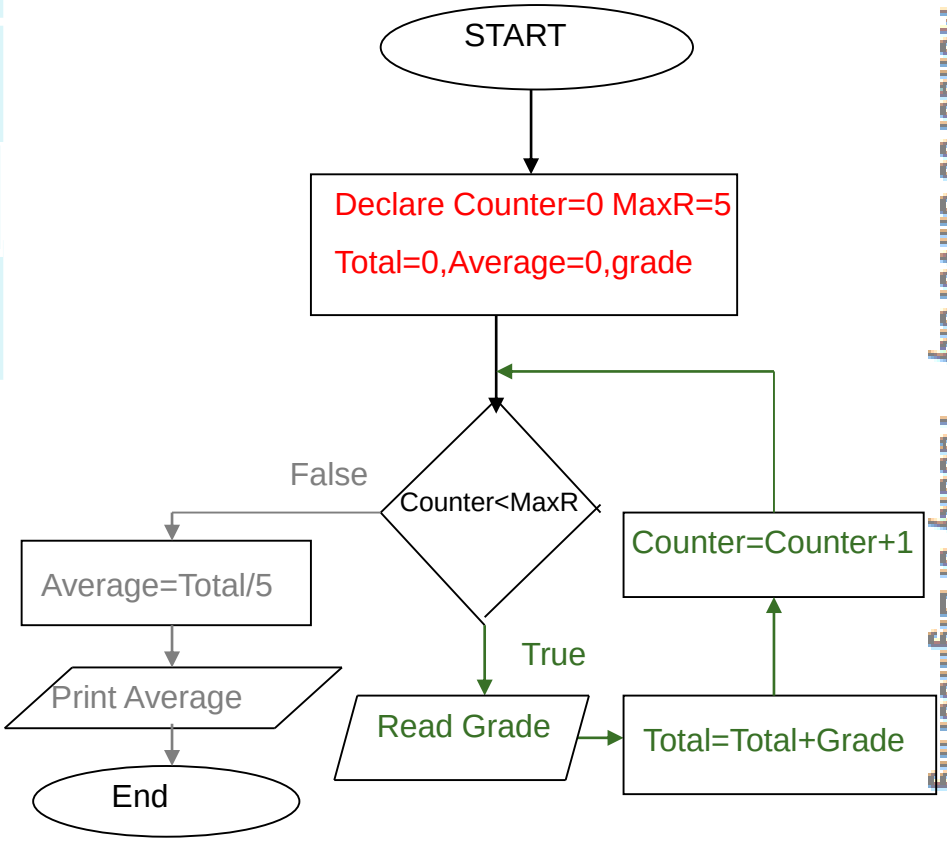


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

A tracing table of the flowchart for the problem 7, assuming that the user has entered the following numbers, 40,30,100,70,60.

	Counter <5	Grade	Total	Average	Counter
Declaration & Initialization:	?	?	0	0	0
Itr 1 of repetition	True	40	40	0	1
Itr 2 of repetition	True	30	70	0	2

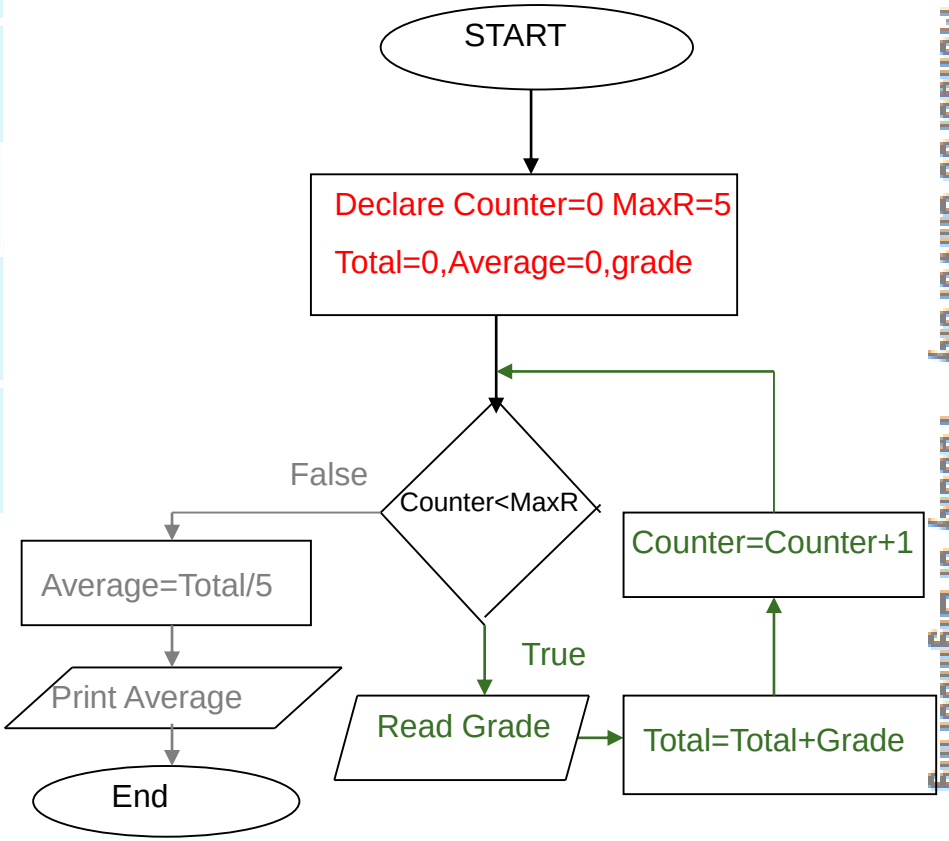


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

A tracing table of the flowchart for the problem 7, assuming that the user has entered the following numbers, 40,30,100,70,60.

	Counter <5	Grade	Total	Average	Counter
Declaration & Initialization:	?	?	0	0	0
Itr 1 of repetition	True	40	40	0	1
Itr 2 of repetition	True	30	70	0	2
Itr 3 of repetition	True	100	170	0	3

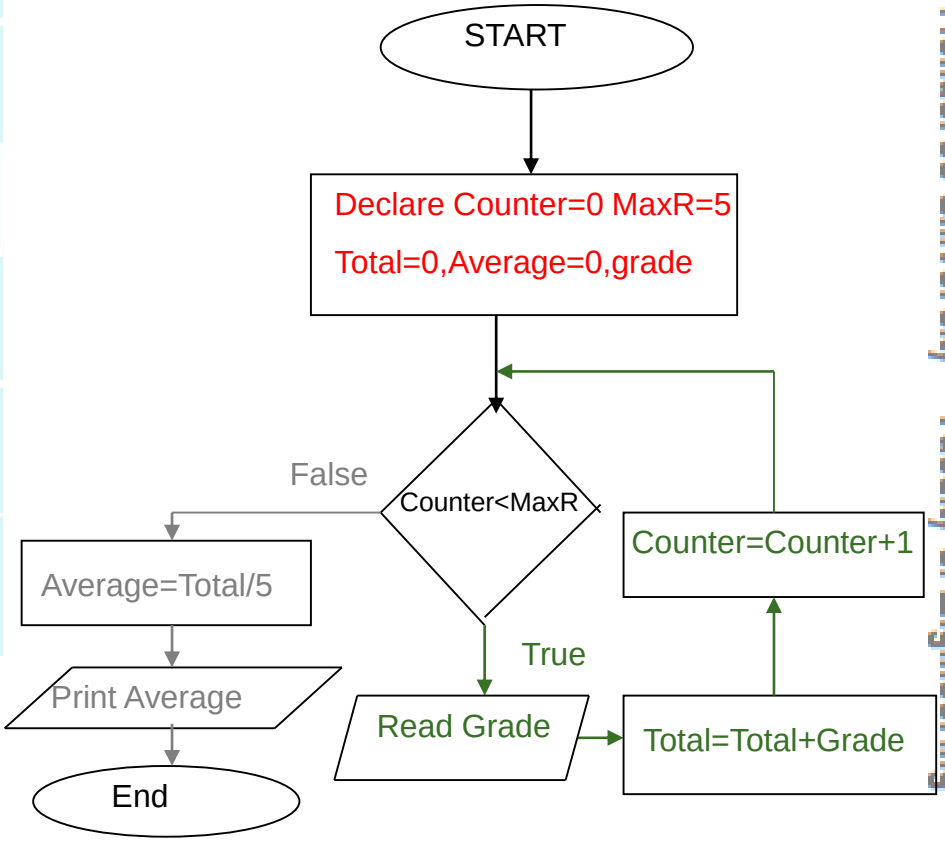


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

A tracing table of the flowchart for the problem 7, assuming that the user has entered the following numbers, 40,30,100,70,60.

	Counter <5	Grade	Total	Average	Counter
Declaration & Initialization:	?	?	0	0	0
Itr 1 of repetition	True	40	40	0	1
Itr 2 of repetition	True	30	70	0	2
Itr 3 of repetition	True	100	170	0	3
Itr 4 of repetition	True	70	240	0	4

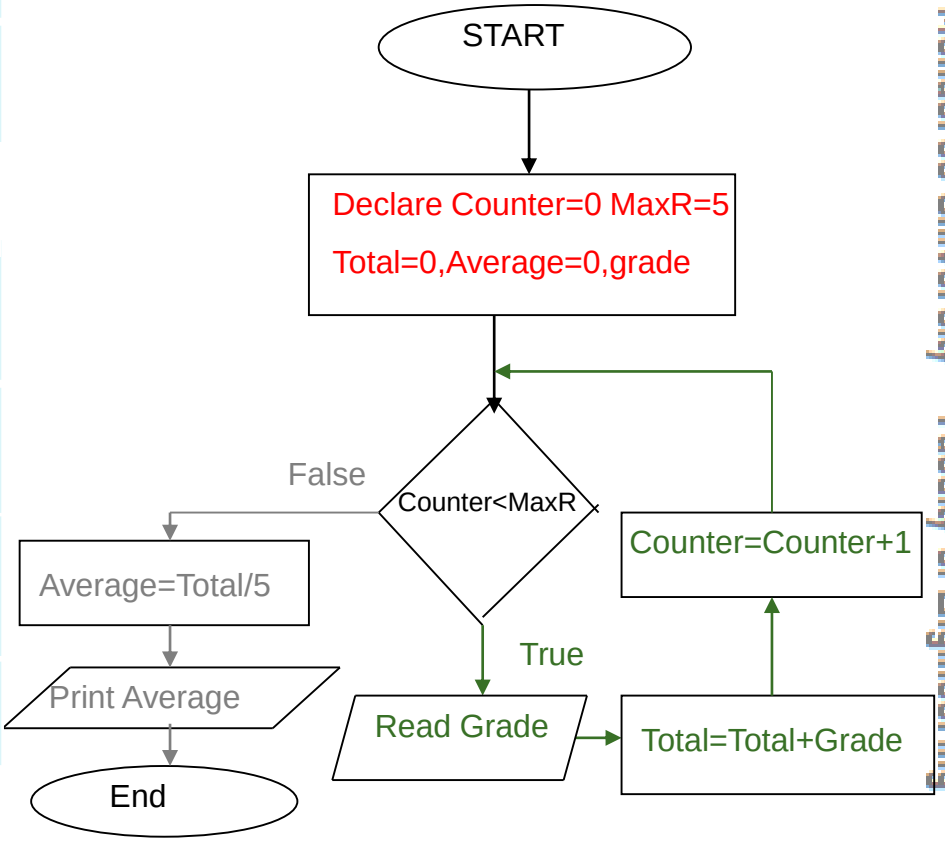


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

A tracing table of the flowchart for the problem 7, assuming that the user has entered the following numbers, 40,30,100,70,60.

	Counter <5	Grade	Total	Average	Counter
Declaration & Initialization:	?	?	0	0	0
Itr 1 of repetition	True	40	40	0	1
Itr 2 of repetition	True	30	70	0	2
Itr 3 of repetition	True	100	170	0	3
Itr 4 of repetition	True	70	240	0	4
Itr 5 of repetition	True	60	300	0	5

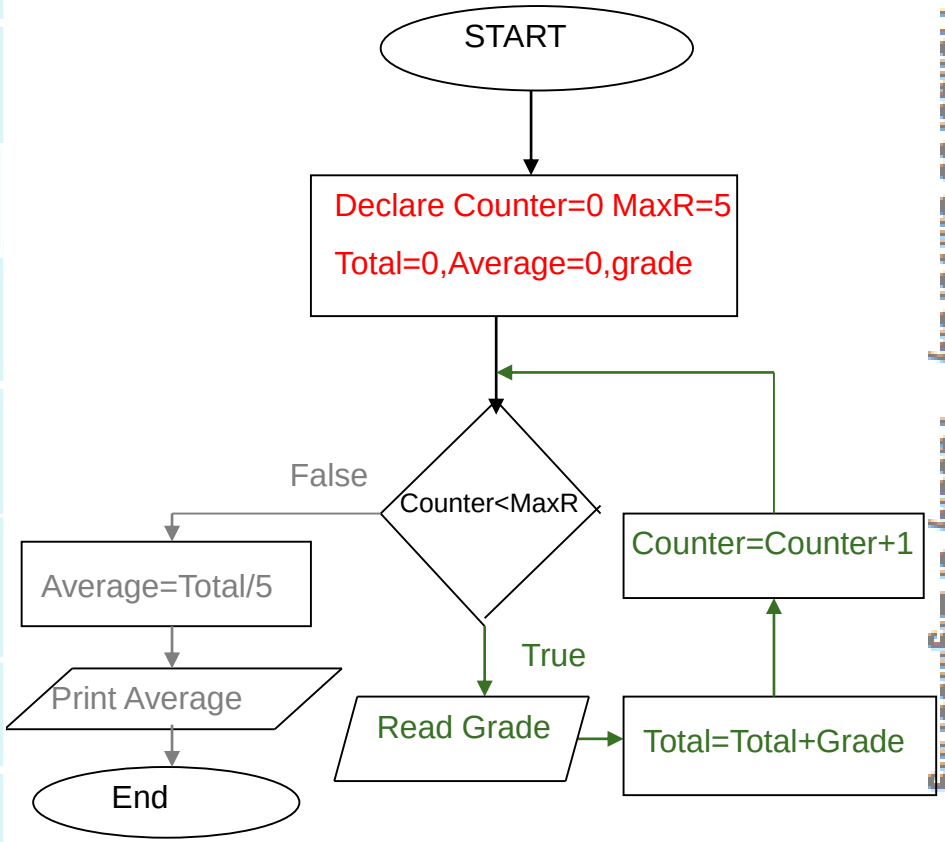


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

A tracing table of the flowchart for the problem 7, assuming that the user has entered the following numbers, 40,30,100,70,60.

	Counter <5	Grade	Total	Average	Counter
Declaration & Initialization:	?	?	0	0	0
Itr 1 of repetition	True	40	40	0	1
Itr 2 of repetition	True	30	70	0	2
Itr 3 of repetition	True	100	170	0	3
Itr 4 of repetition	True	70	240	0	4
Itr 5 of repetition	True	60	300	0	5
End of repetition	false	60	300	60	5



T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 8:

Draw a flowchart for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user. n! = n*(n-1)*(n-2)*…….*1

T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 8:

Draw a flowchart for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user. n! = n*(n-1)*(n-2)*…….*1

Question to answer before you start.

1. What is the maximum number of repetition?

   MaxR= n

2. What is the initial value of the counter?

   Counter = 1

3. What is the condition of repetition?

   Counter < =MaxR

4. Shoud I increment or decrement the **Counter** after each iteration?

   Counter = Counter+1

5. What are the actions that should be repeated?

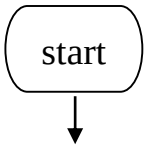
   Multiply with n, decrement n, and increment the Counter

T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 8:

Draw a flowchart for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user. n! = n*(n-1)*(n-2)*…….*1



Question to answer before you start.

1. What is the maximum number of repetition?

   MaxR= n

2. What is the initial value of the counter?

   Counter = 1

3. What is the condition of repetition?

   Counter < =MaxR

4. Shoud I increment or decrement the **Counter** after each iteration?

   Counter = Counter+1

5. What are the actions that should be repeated?

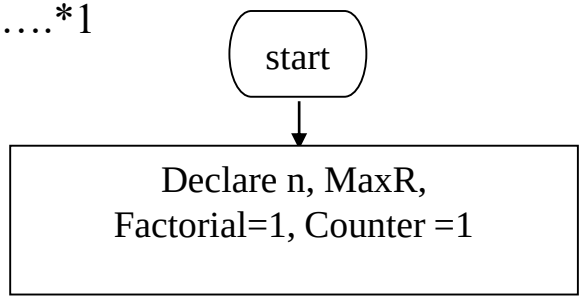
   Multiply with n, decrement n, and increment the Counter

T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 8:

Draw a flowchart for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user. n! = n*(n-1)*(n-2)*…….*1



Question to answer before you start.

1. What is the maximum number of repetition?

   MaxR= n

2. What is the initial value of the counter?

   Counter = 1

3. What is the condition of repetition?

   Counter < =MaxR

4. Shoud I increment or decrement the **Counter** after each iteration?

   Counter = Counter+1

5. What are the actions that should be repeated?

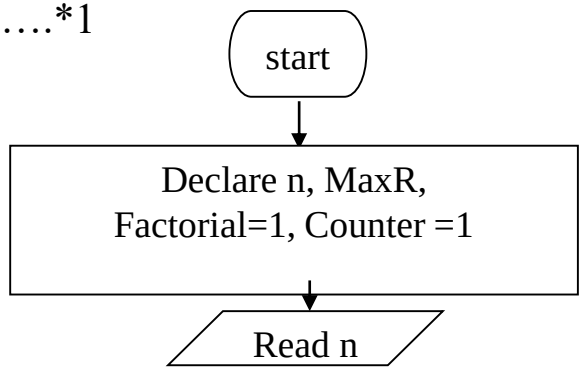
   Multiply with n, decrement n, and increment the Counter

T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 8:

Draw a flowchart for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user. n! = n*(n-1)*(n-2)*…….*1



Question to answer before you start.

1. What is the maximum number of repetition?

   MaxR= n

2. What is the initial value of the counter?

   Counter = 1

3. What is the condition of repetition?

   Counter < =MaxR

4. Shoud I increment or decrement the **Counter** after each iteration?

   Counter = Counter+1

5. What are the actions that should be repeated?

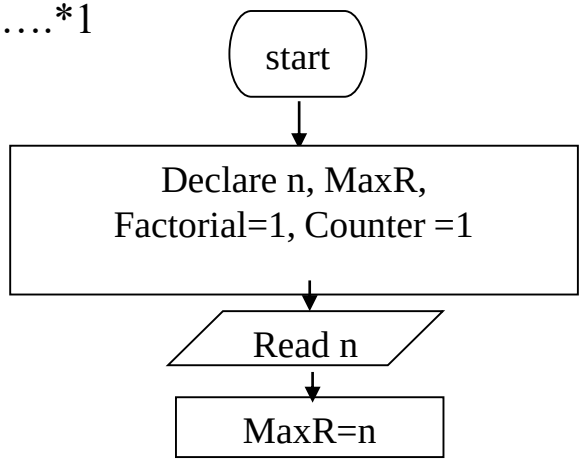
   Multiply with n, decrement n, and increment the Counter

T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 8:

Draw a flowchart for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user. n! = n*(n-1)*(n-2)*…….*1



Question to answer before you start.

1. What is the maximum number of repetition?

   MaxR= n

2. What is the initial value of the counter?

   Counter = 1

3. What is the condition of repetition?

   Counter < =MaxR

4. Shoud I increment or decrement the **Counter** after each iteration?

   Counter = Counter+1

5. What are the actions that should be repeated?

   Multiply with n, decrement n, and increment the Counter

T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 8:

Draw a flowchart for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user. n! = n*(n-1)*(n-2)*…….*1

Question to answer before you start.

1. What is the maximum number of repetition?

   MaxR= n

2. What is the initial value of the counter?

   Counter = 1

3. What is the condition of repetition?

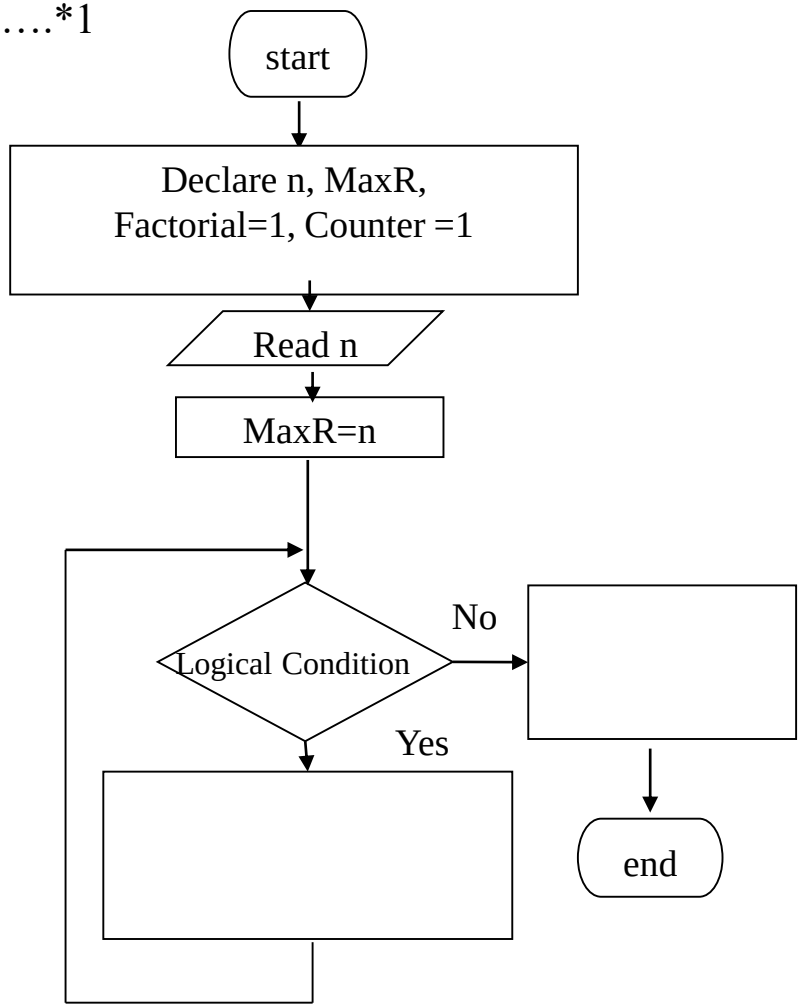
   Counter < =MaxR

4. Shoud I increment or decrement the **Counter** after each iteration?

   Counter = Counter+1

5. What are the actions that should be repeated?

   Multiply with n, decrement n, and increment the Counter



T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 8:

Draw a flowchart for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user. n! = n*(n-1)*(n-2)*…….*1

Question to answer before you start.

1. What is the maximum number of repetition?

   MaxR= n

2. What is the initial value of the counter?

   Counter = 1

3. What is the condition of repetition?

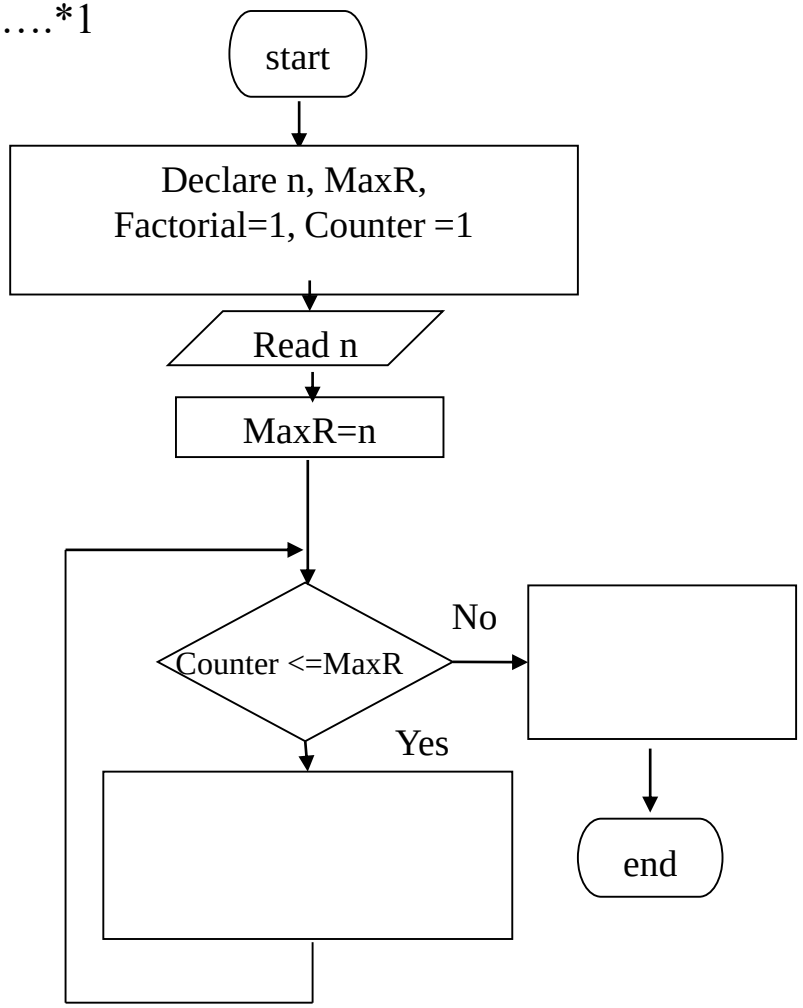
   Counter < =MaxR

4. Shoud I increment or decrement the **Counter** after each iteration?

   Counter = Counter+1

5. What are the actions that should be repeated?

   Multiply with n, decrement n, and increment the Counter



T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 8:

Draw a flowchart for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user. n! = n*(n-1)*(n-2)*…….*1

Question to answer before you start.

1. What is the maximum number of repetition?

   MaxR= n

2. What is the initial value of the counter?

   Counter = 1

3. What is the condition of repetition?

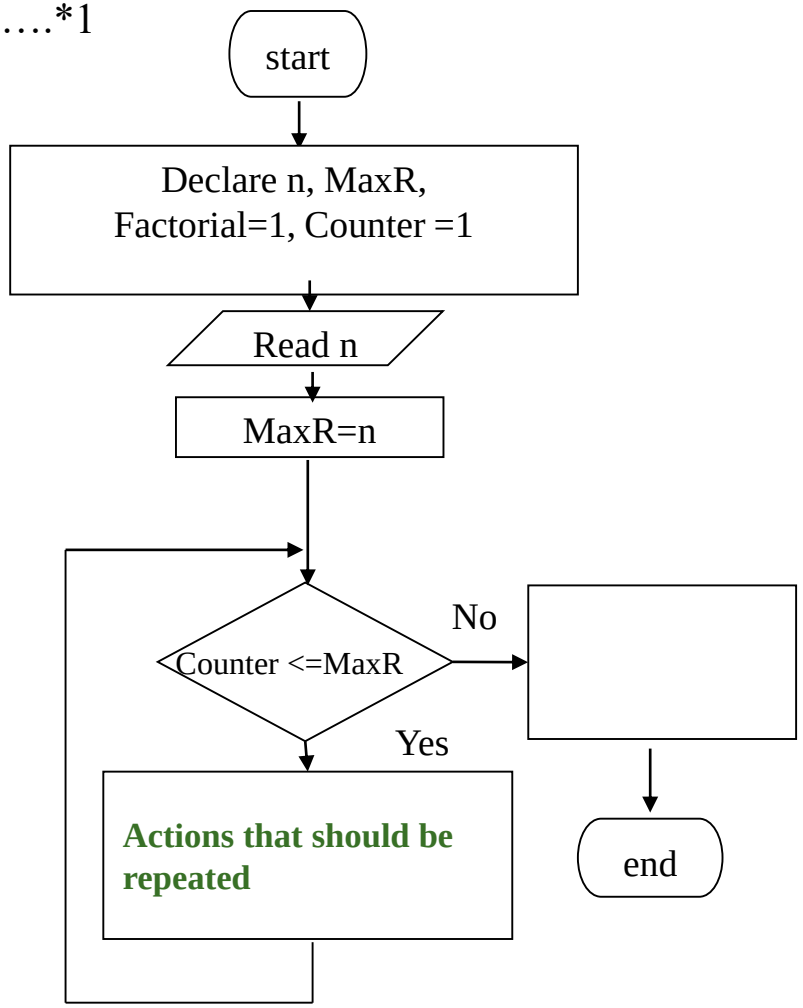
   Counter < =MaxR

4. Shoud I increment or decrement the **Counter** after each iteration?

   Counter = Counter+1

5. What are the actions that should be repeated?

   Multiply with n, decrement n, and increment the Counter



T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 8:

Draw a flowchart for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user. n! = n*(n-1)*(n-2)*…….*1

Question to answer before you start.

1. What is the maximum number of repetition?

   MaxR= n

2. What is the initial value of the counter?

   Counter = 1

3. What is the condition of repetition?

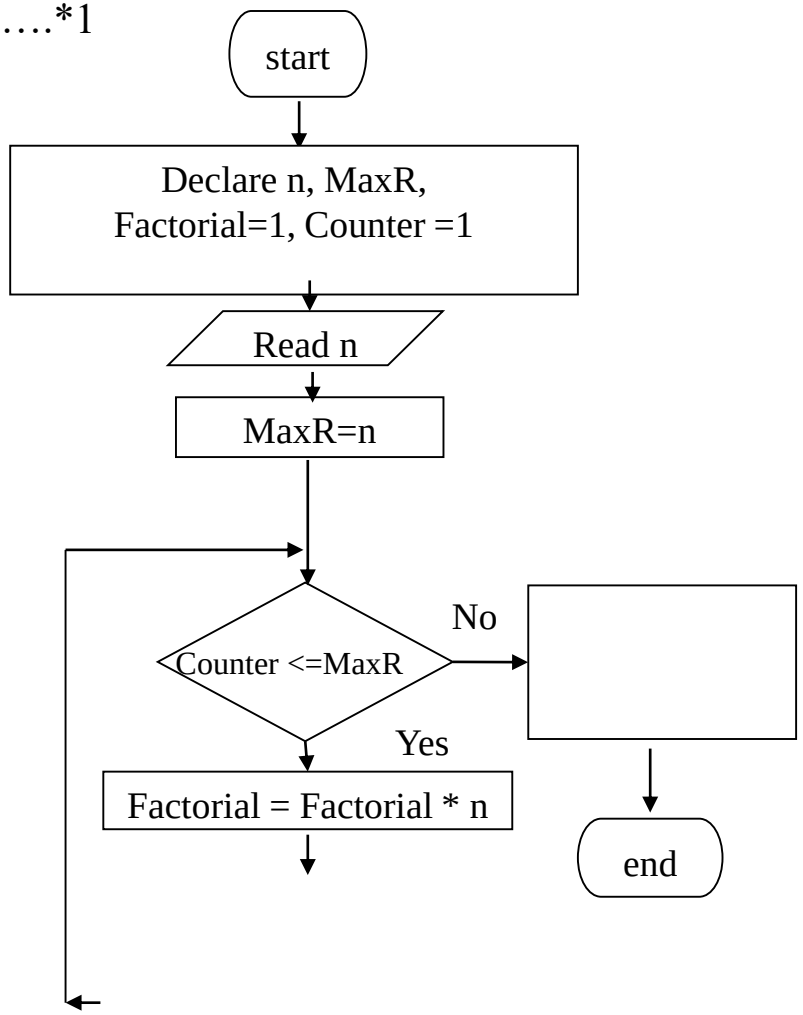
   Counter < =MaxR

4. Shoud I increment or decrement the **Counter** after each iteration?

   Counter = Counter+1

5. What are the actions that should be repeated?

   Multiply with n, decrement n, and increment the Counter



T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 8:

Draw a flowchart for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user. n! = n*(n-1)*(n-2)*…….*1

Question to answer before you start.

1. What is the maximum number of repetition?

   MaxR= n

2. What is the initial value of the counter?

   Counter = 1

3. What is the condition of repetition?

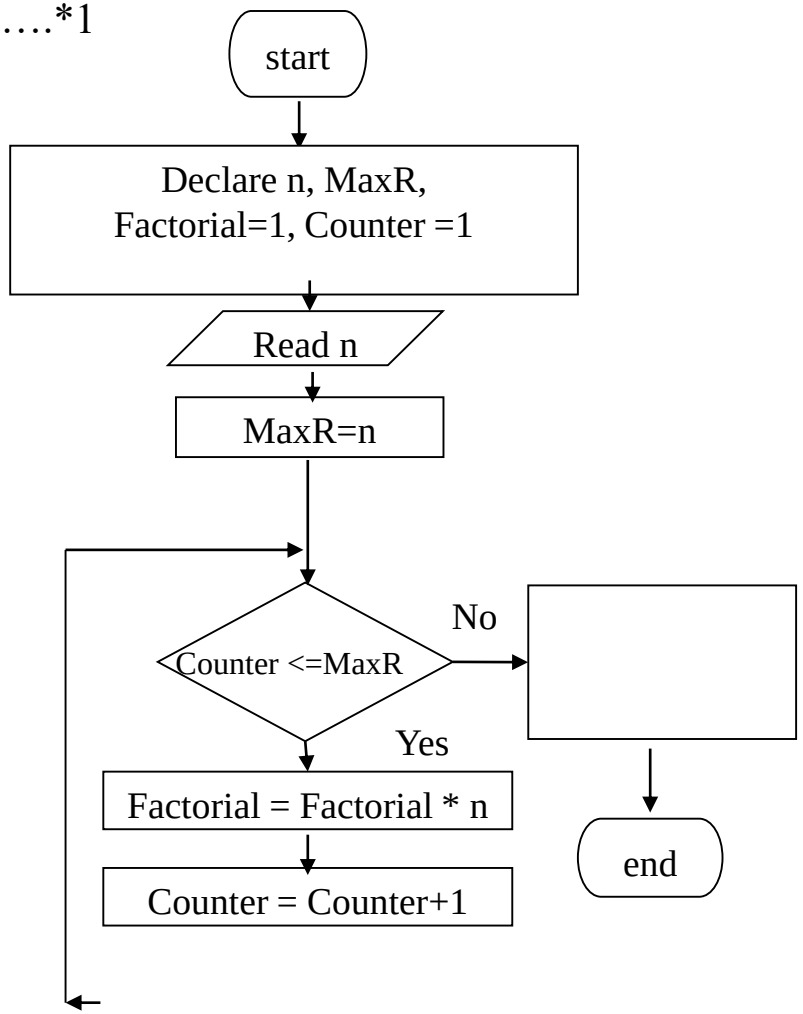
   Counter < =MaxR

4. Shoud I increment or decrement the **Counter** after each iteration?

   Counter = Counter+1

5. What are the actions that should be repeated?

   Multiply with n, decrement n, and increment the Counter



T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 8:

Draw a flowchart for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user. n! = n*(n-1)*(n-2)*…….*1

Question to answer before you start.

1. What is the maximum number of repetition?

   MaxR= n

2. What is the initial value of the counter?

   Counter = 1

3. What is the condition of repetition?

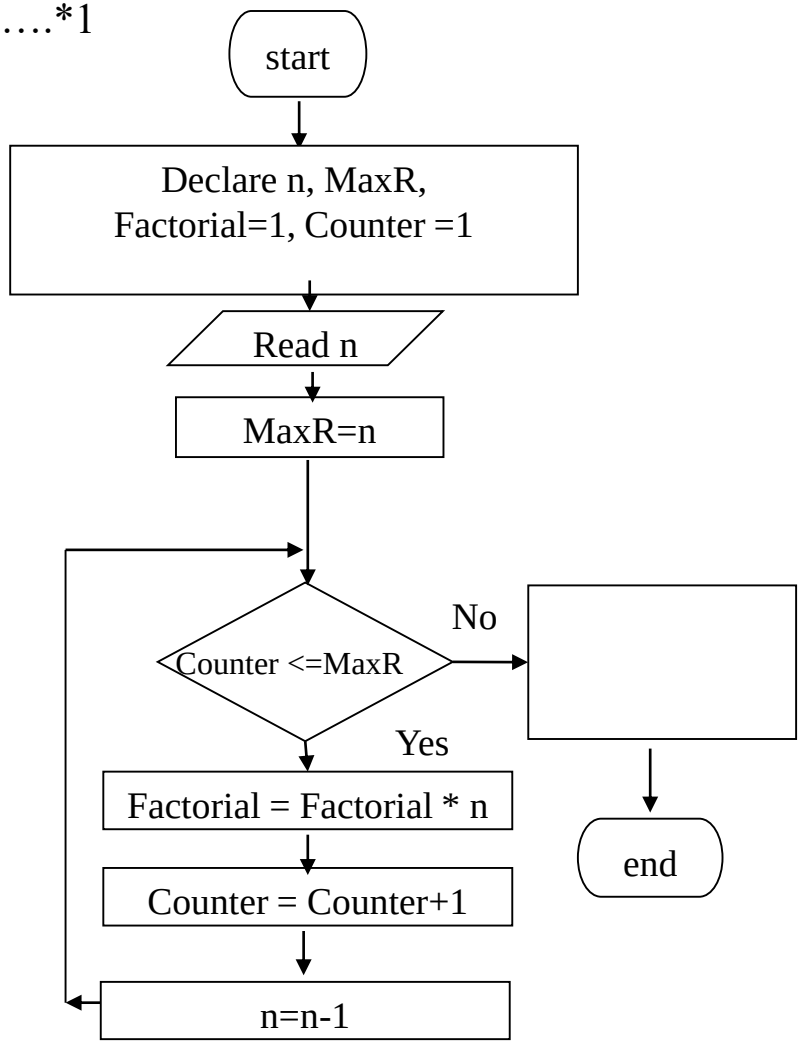
   Counter < =MaxR

4. Shoud I increment or decrement the **Counter** after each iteration?

   Counter = Counter+1

5. What are the actions that should be repeated?

   Multiply with n, decrement n, and increment the Counter



T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 8:

Draw a flowchart for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user. n! = n*(n-1)*(n-2)*…….*1

Question to answer before you start.

1. What is the maximum number of repetition?

   MaxR= n

2. What is the initial value of the counter?

   Counter = 1

3. What is the condition of repetition?

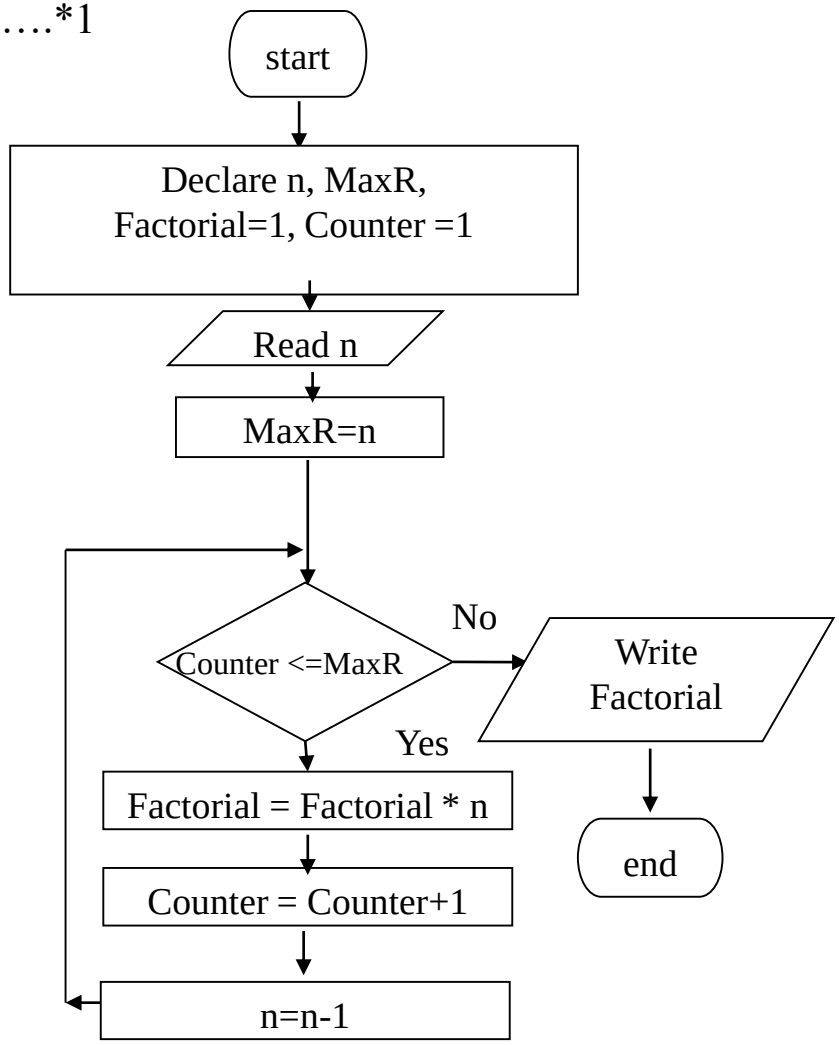
   Counter < =MaxR

4. Shoud I increment or decrement the **Counter** after each iteration?

   Counter = Counter+1

5. What are the actions that should be repeated?

   Multiply with n, decrement n, and increment the Counter

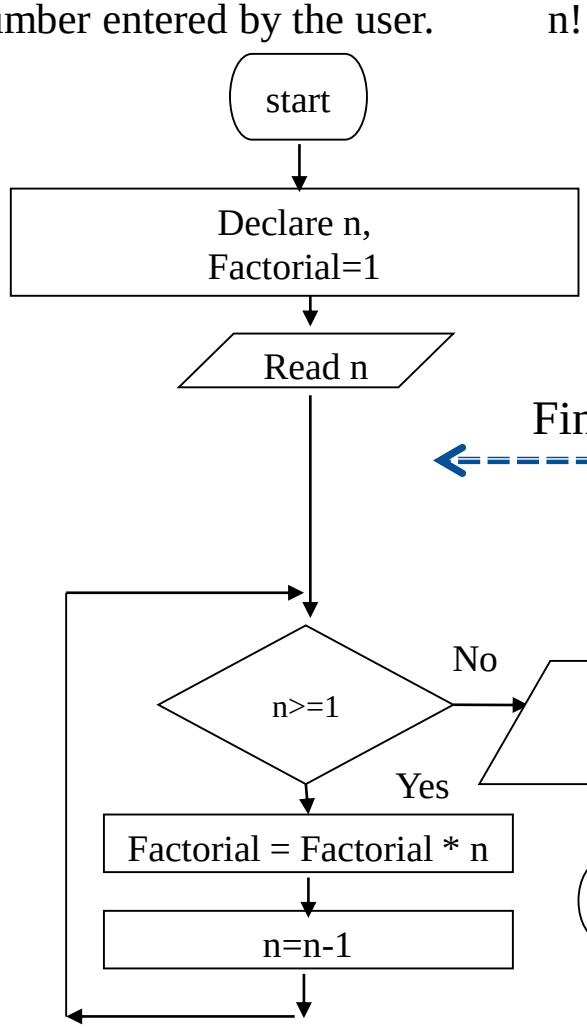


T.C. Adnan Menderes University Faculty of Engineering

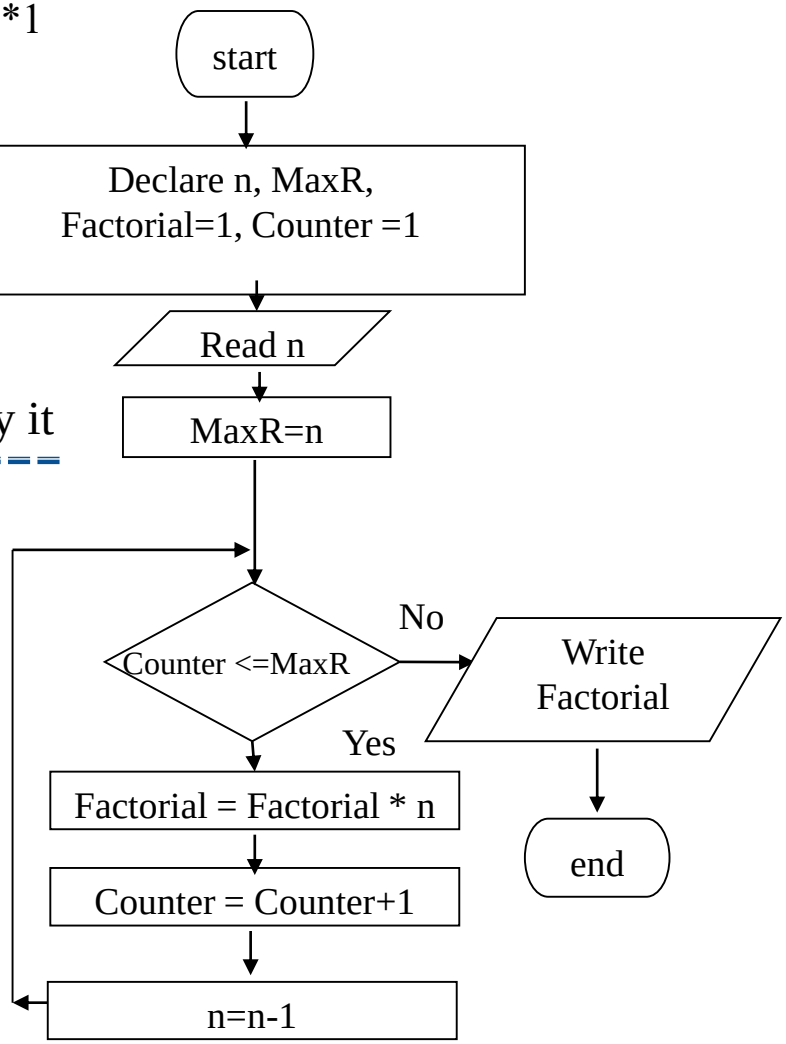
Basic Flowchart Structures

Problem 8:

Draw a flowchart for a program that calculates and prints the factorial (n!) of n. Where n is an integer number entered by the user. n! = n*(n-1)*(n-2)*…….*1



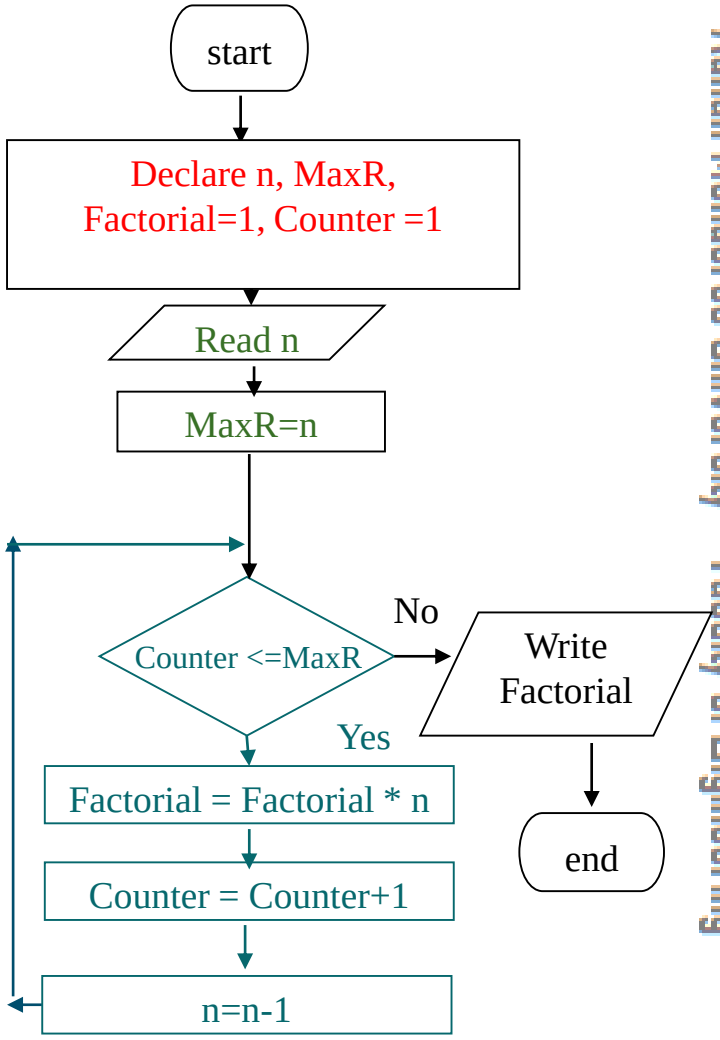
Finally, we can simplify it



T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Tracing of the previous flowchart, assuming that the user has entered the number 3.

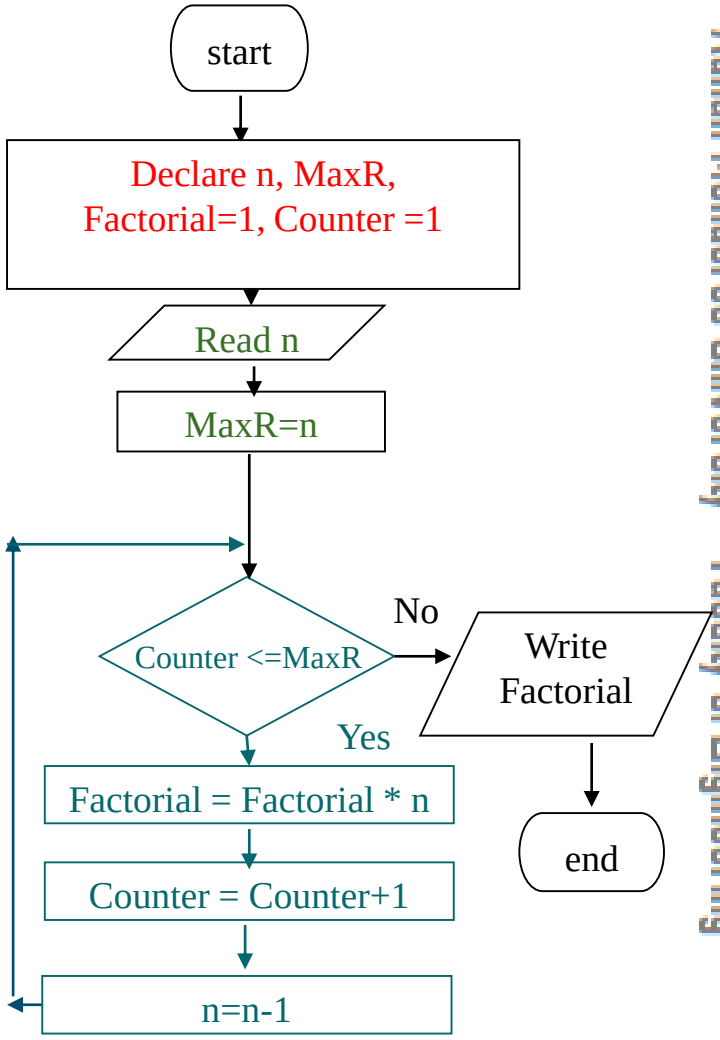


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Tracing of the previous flowchart, assuming that the user has entered the number 3.

	Counter <=MaxR	Factorial	Counter	n	MaxR
Declaration & Initialization:	**?**	**1**	**1**	**?**	**?**

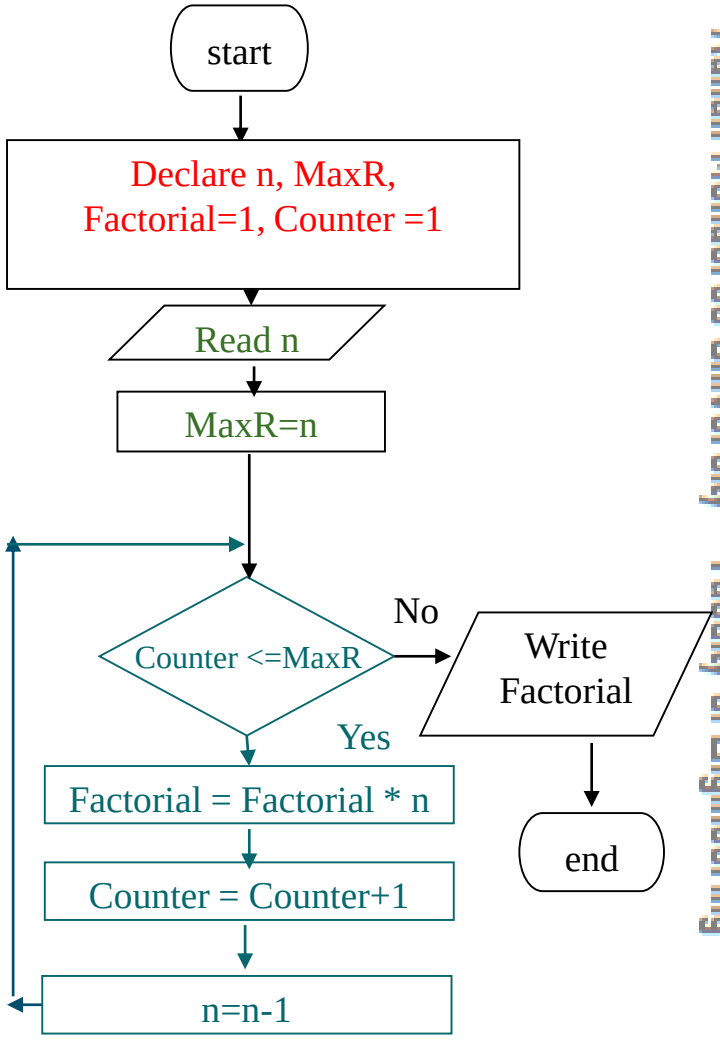


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Tracing of the previous flowchart, assuming that the user has entered the number 3.

	Counter <=MaxR	Factorial	Counter	n	MaxR
Declaration & Initialization:	**?**	**1**	**1**	**?**	**?**
Next 2 processes	**?**	**1**	**1**	**3**	**3**

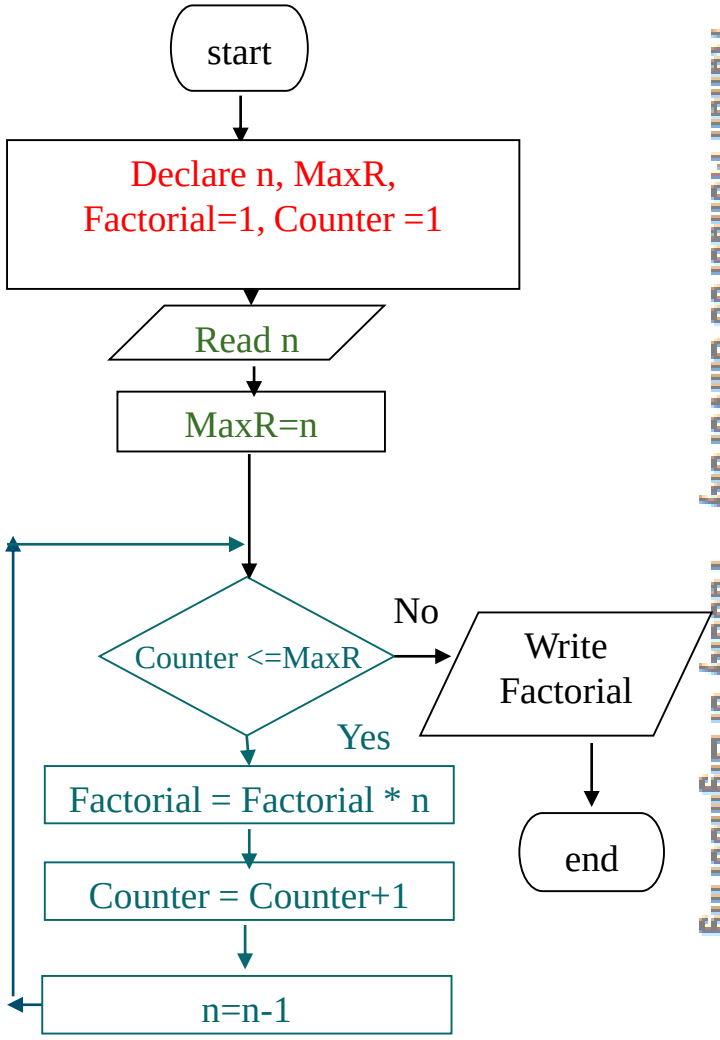


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Tracing of the previous flowchart, assuming that the user has entered the number 3.

	Counter <=MaxR	Factorial	Counter	n	MaxR
Declaration & Initialization:	**?**	**1**	**1**	**?**	**?**
Next 2 processes	**?**	**1**	**1**	**3**	**3**
Itr 1 of repetition	**Yes**	**3**	**2**	**2**	**3**

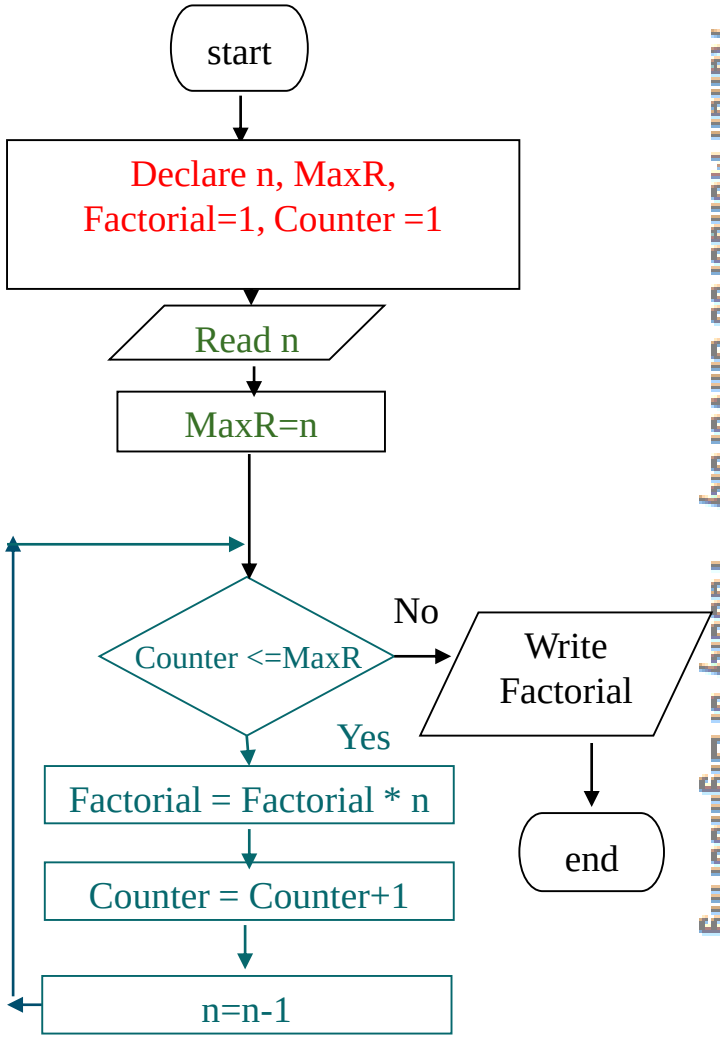


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Tracing of the previous flowchart, assuming that the user has entered the number 3.

	Counter <=MaxR	Factorial	Counter	n	MaxR
Declaration & Initialization:	?	1	1	?	?
Next 2 processes	?	1	1	3	3
Itr 1 of repetition	Yes	3	2	2	3
Itr 2 of repetition	Yes	6	3	1	3

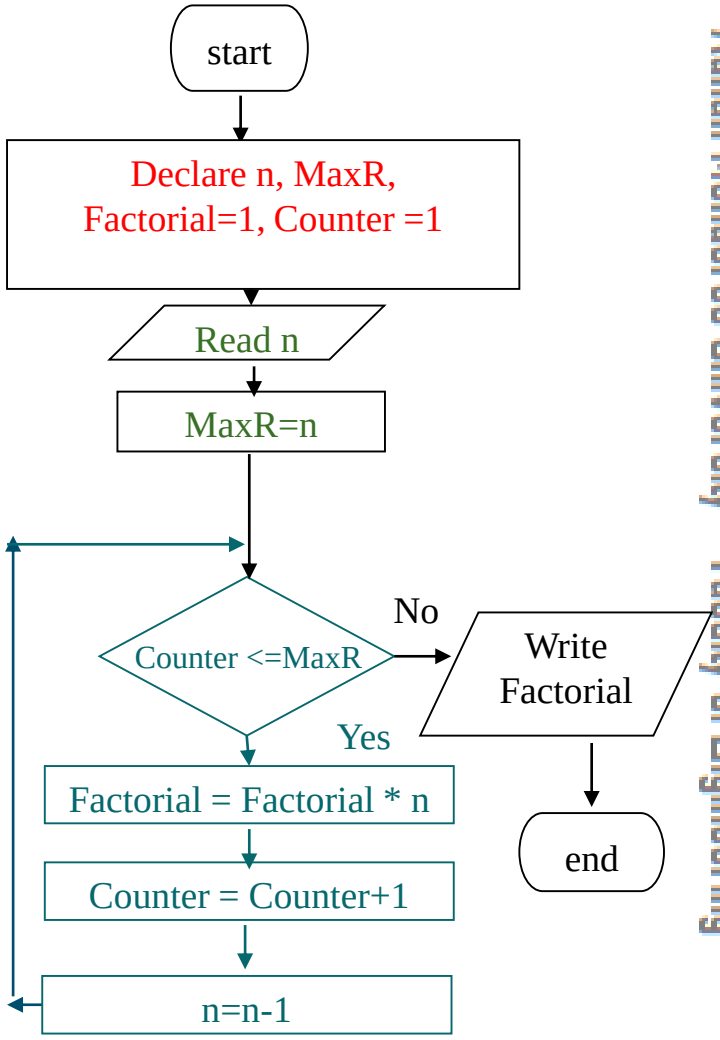


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Tracing of the previous flowchart, assuming that the user has entered the number 3.

	Counter <=MaxR	Factorial	Counter	n	MaxR
Declaration & Initialization:	?	1	1	?	?
Next 2 processes	?	1	1	3	3
Itr 1 of repetition	Yes	3	2	2	3
Itr 2 of repetition	Yes	6	3	1	3
Itr 3 of repetition	Yes	6	4	0	3

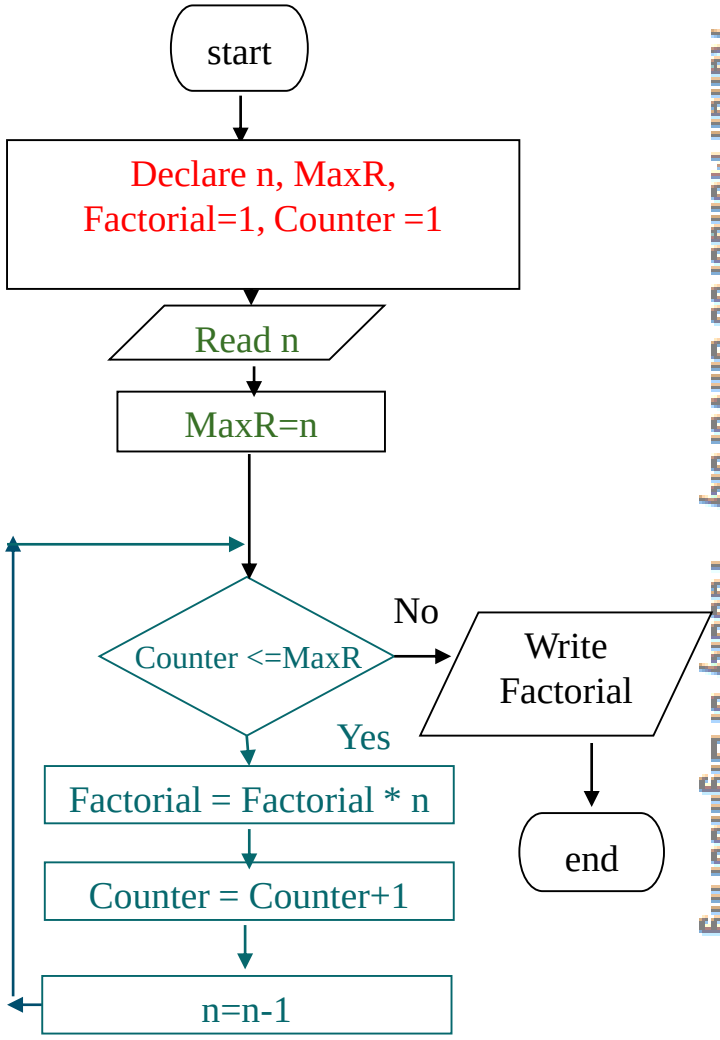


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Tracing of the previous flowchart, assuming that the user has entered the number 3.

	Counter <=MaxR	Factorial	Counter	n	MaxR
Declaration & Initialization:	?	1	1	?	?
Next 2 processes	?	1	1	3	3
Itr 1 of repetition	Yes	3	2	2	3
Itr 2 of repetition	Yes	6	3	1	3
Itr 3 of repetition	Yes	6	4	0	3
End of repetition	No	6	4	0	3



T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 9: Write a pseudocode and draw a flowchart for a program that finds and prints the largest number of N numbers (N can be any number). Read numbers one by one.

Psuedocode

```
Declare counter = 1, N, CuNum, Max
Read N, CuNum
Max = CuNum
FOR Counter=1:N-1
Read CuNum
IF (CuNum >Max)
  THEN
     Max= CuNum
ENDEF
Counter=Counter+1
ENDFOR
Write Max
```

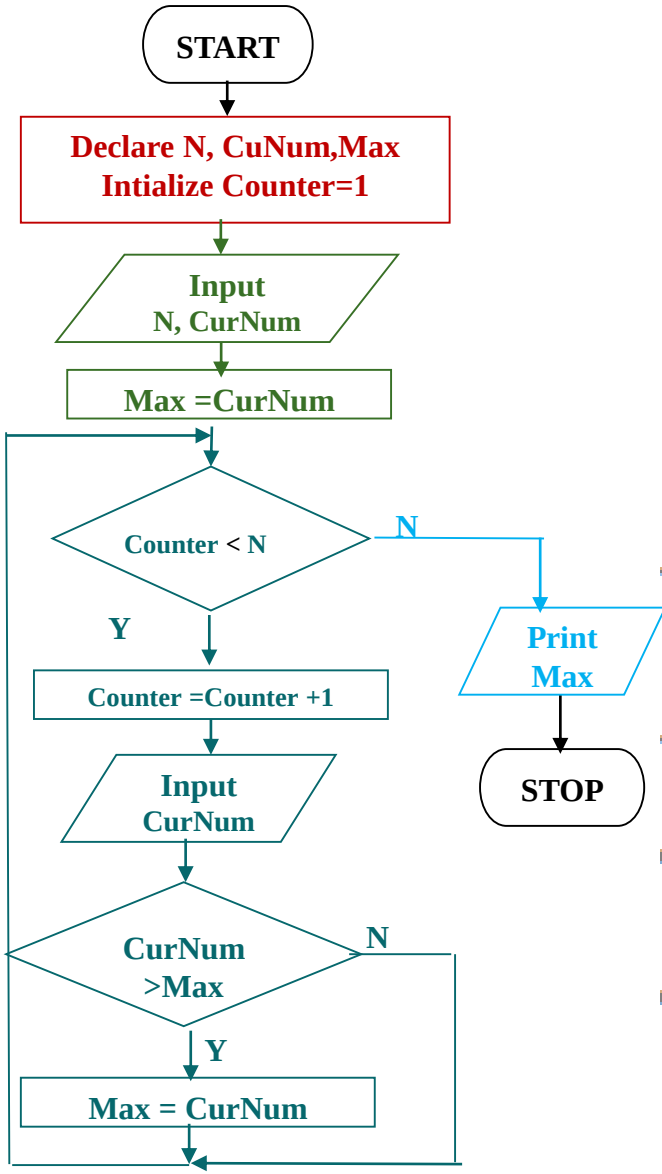
T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Problem 9: Write a pseudocode and draw a flowchart for a program that finds and prints the largest number of N numbers (N can be any number). Read numbers one by one.

Psuedocode

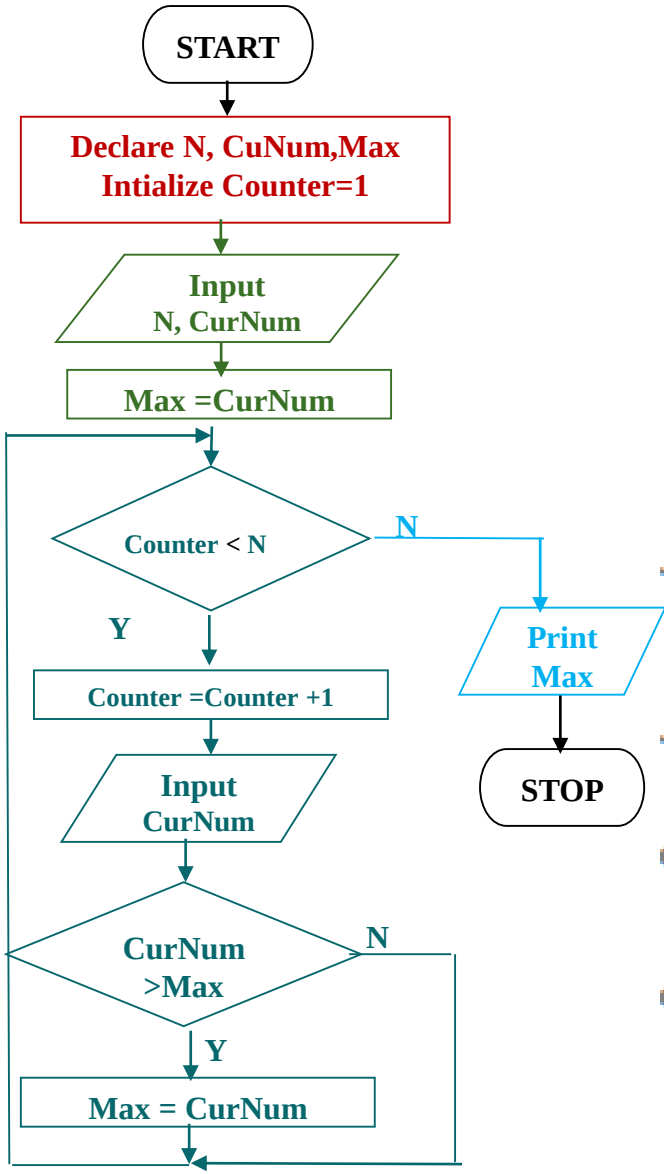
```
Declare counter = 1, N, CuNum, Max
Read N, CuNum
Max = CuNum
FOR Counter=1:N-1
Read CuNum
IF (CuNum >Max)
  THEN
     Max= CuNum
ENDEF
Counter=Counter+1
ENDFOR
Write Max
```



T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Tracing of the previous flowchart, assuming that the user has entered the following numbers: 5 for the number N, and then the following numbers one by one (1, 4, 2, 6, 8)

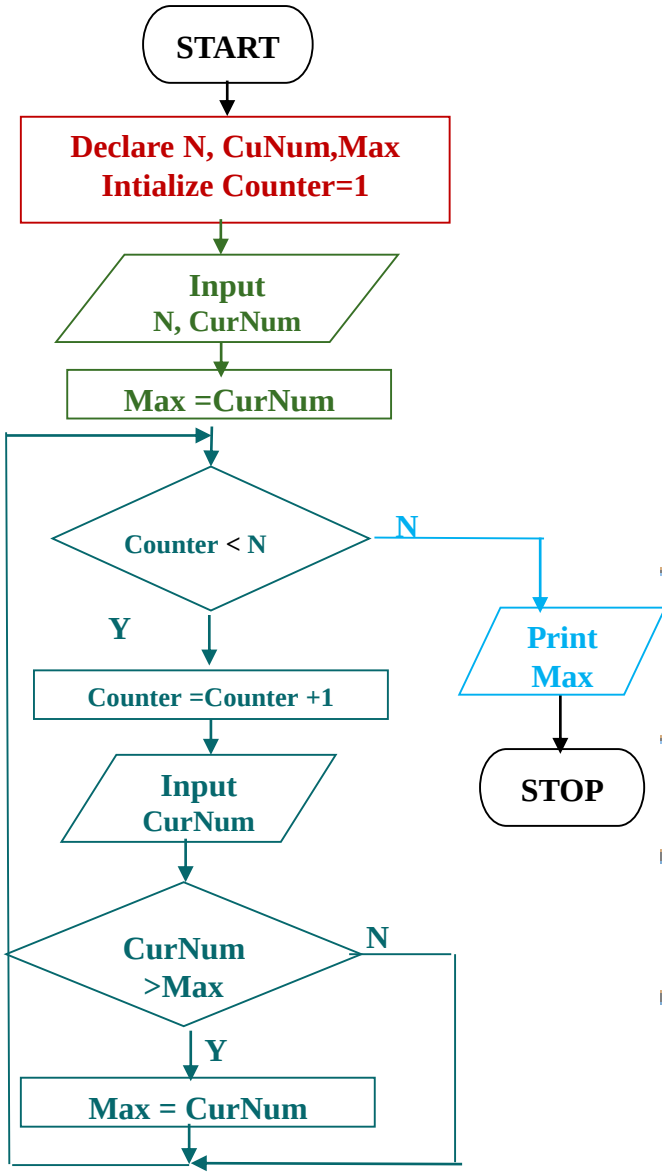


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Tracing of the previous flowchart, assuming that the user has entered the following numbers: 5 for the number N, and then the following numbers one by one (1, 4, 2, 6, 8)

	N	Counter< N	Counter	CuNum	CuNum>Max	Max
Declaration & Initialization:	?	?	1	?	?	?

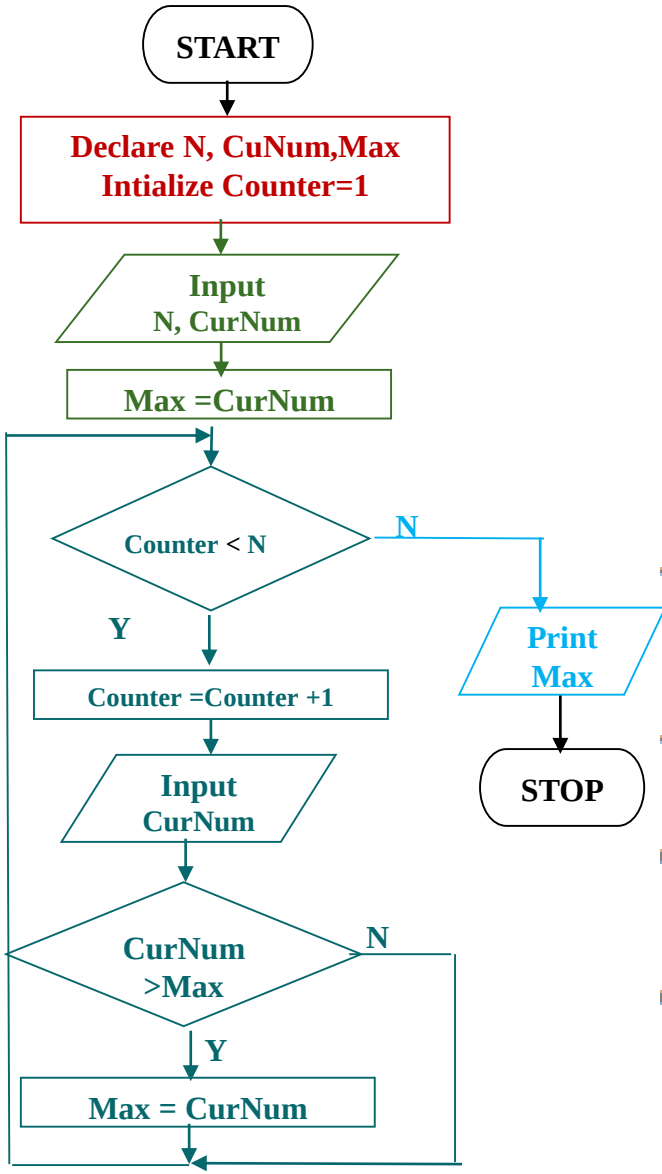


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Tracing of the previous flowchart, assuming that the user has entered the following numbers: 5 for the number N, and then the following numbers one by one (1, 4, 2, 6, 8)

	N	Counter< N	Counter	CuNum	CuNum>Max	Max
Declaration & Initialization:	?	?	1	?	?	?
Next 2 processes	5	?	1	1	?	1

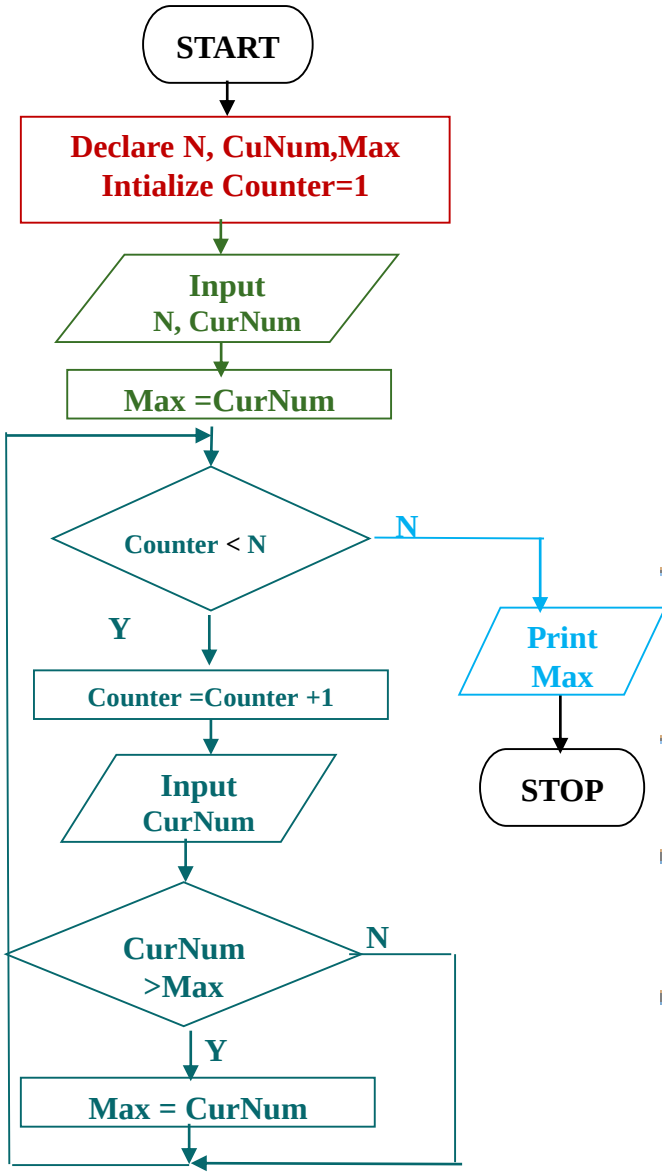


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Tracing of the previous flowchart, assuming that the user has entered the following numbers: 5 for the number N, and then the following numbers one by one (1, 4, 2, 6, 8)

	N	Counter< N	Counter	CuNum	CuNum>Max	Max
Declaration & Initialization:	?	?	1	?	?	?
Next 2 processes	5	?	1	1	?	1
Itr 1 of repetition	5	Y	2	4	Y	4

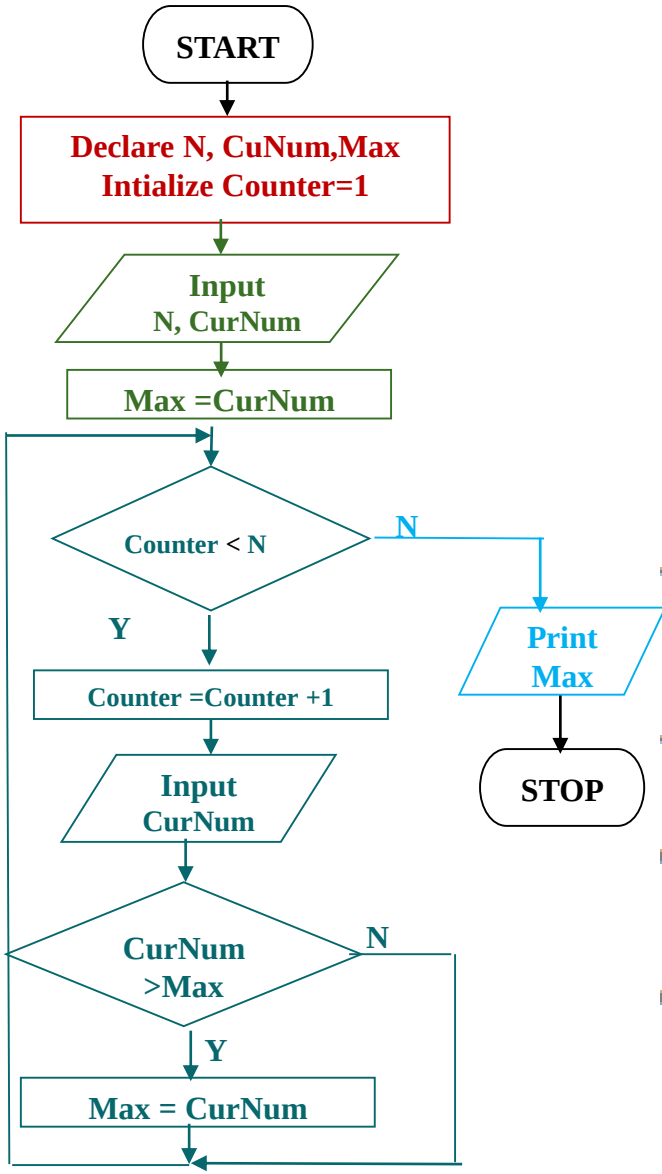


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Tracing of the previous flowchart, assuming that the user has entered the following numbers: 5 for the number N, and then the following numbers one by one (1, 4, 2, 6, 8)

	N	Counter< N	Counter	CuNum	CuNum>Max	Max
Declaration & Initialization:	?	?	1	?	?	?
Next 2 processes	5	?	1	1	?	1
Itr 1 of repetition	5	Y	2	4	Y	4
Itr 2 of repetition	5	Y	3	2	N	4

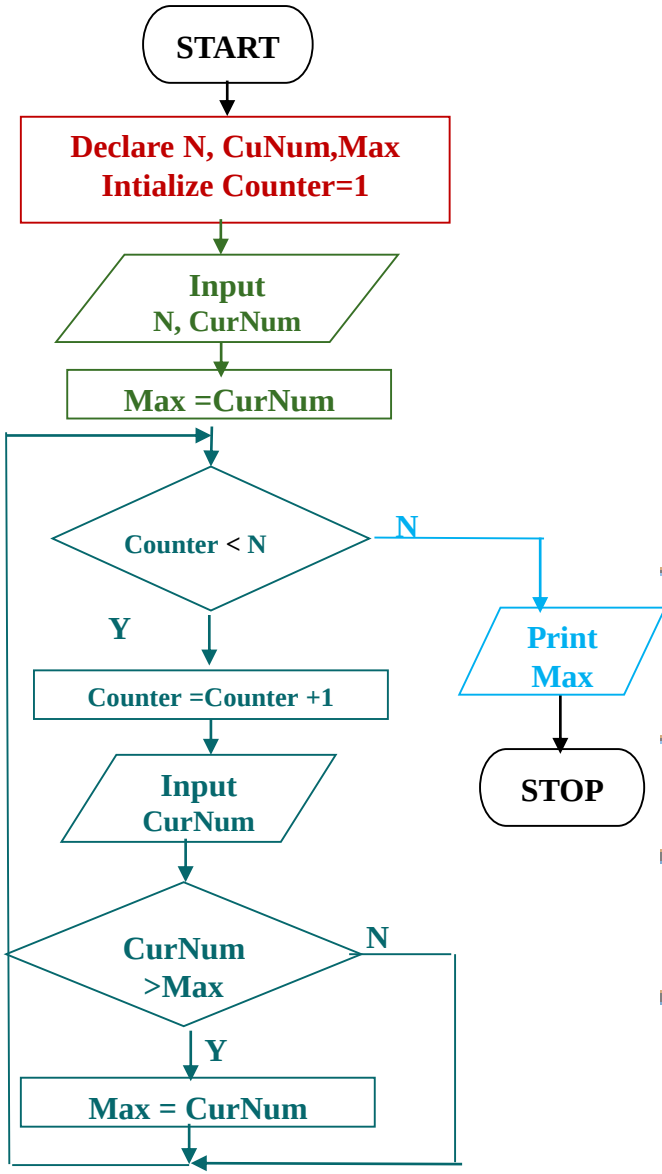


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Tracing of the previous flowchart, assuming that the user has entered the following numbers: 5 for the number N, and then the following numbers one by one (1, 4, 2, 6, 8)

	N	Counter< N	Counter	CuNum	CuNum>Max	Max
Declaration & Initialization:	?	?	1	?	?	?
Next 2 processes	5	?	1	1	?	1
Itr 1 of repetition	5	Y	2	4	Y	4
Itr 2 of repetition	5	Y	3	2	N	4
Itr 3 of repetition	5	Y	4	6	Y	6

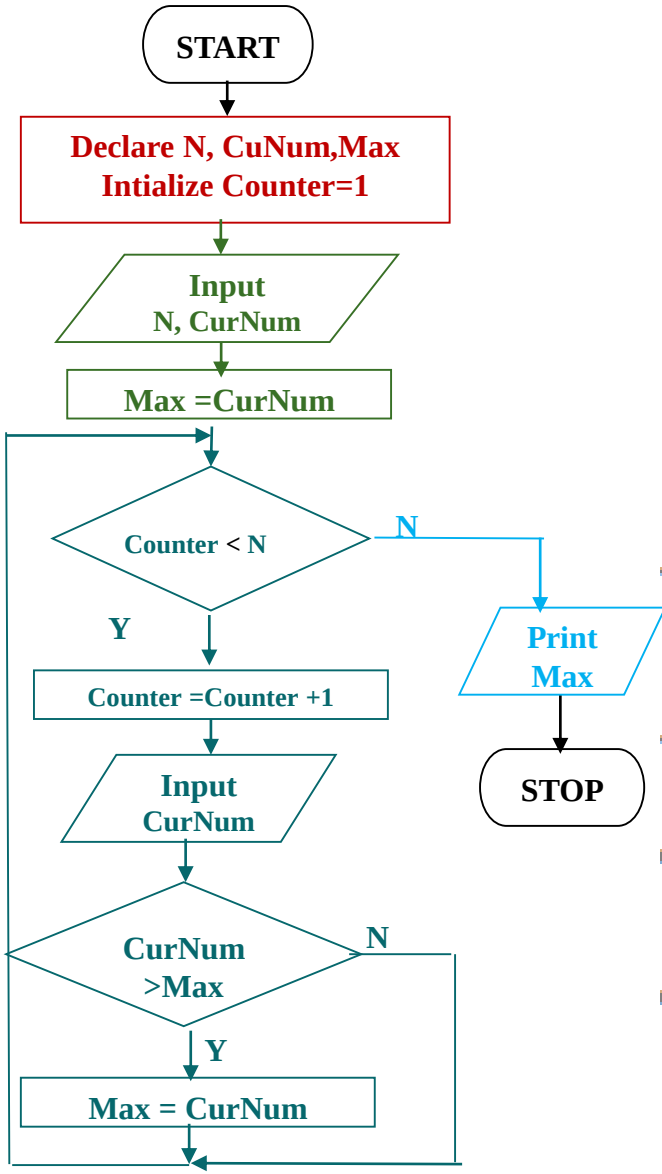


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Tracing of the previous flowchart, assuming that the user has entered the following numbers: 5 for the number N, and then the following numbers one by one (1, 4, 2, 6, 8)

	N	Counter< N	Counter	CuNum	CuNum>Max	Max
Declaration & Initialization:	?	?	1	?	?	?
Next 2 processes	5	?	1	1	?	1
Itr 1 of repetition	5	Y	2	4	Y	4
Itr 2 of repetition	5	Y	3	2	N	4
Itr 3 of repetition	5	Y	4	6	Y	6
Itr 4 of repetition	5	Y	5	8	Y	8

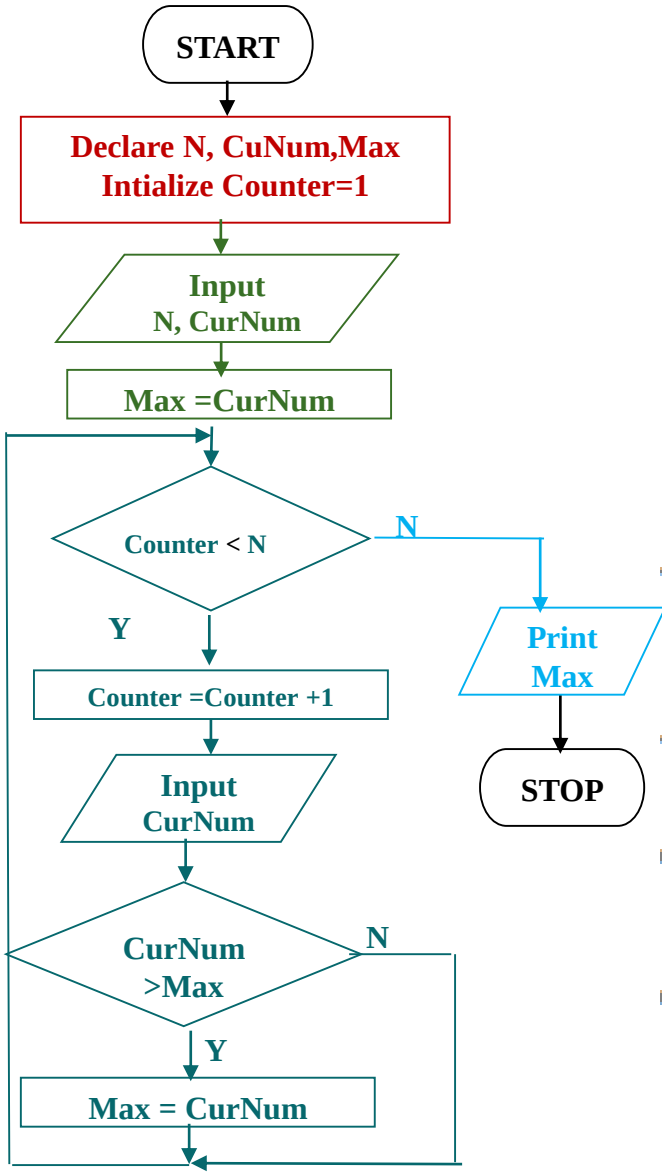


T.C. Adnan Menderes University Faculty of Engineering

Basic Flowchart Structures

Tracing of the previous flowchart, assuming that the user has entered the following numbers: 5 for the number N, and then the following numbers one by one (1, 4, 2, 6, 8)

	N	Counter< N	Counter	CuNum	CuNum>Max	Max
Declaration & Initialization:	?	?	1	?	?	?
Next 2 processes	5	?	1	1	?	1
Itr 1 of repetition	5	Y	2	4	Y	4
Itr 2 of repetition	5	Y	3	2	N	4
Itr 3 of repetition	5	Y	4	6	Y	6
Itr 4 of repetition	5	Y	5	8	Y	8
End of repetition	5	N	5	8	Y	8



T.C. Adnan Menderes University Faculty of Engineering

References

- https://www.site.uottawa.ca/~ivan/07.alg_1.ppt
- Prog 0101 fundamentals of programming- chapter 3 algorithms-FTMS Global College- Kuala Lumpur, Malaysia.

T.C. Adnan Menderes University Faculty of Engineering