CSE111 Introduction to Computer Applications

Lecture 14

Introduction to Matlab - Part 2

Prepared by Asst. Prof. Dr. Mohamed KURDI

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

Summary of the Last Week

- Introducing Matlab Desktop Panels.
  - Entering Commands and Expressions.
  - Performing Arithmetic Operations.
  - Managing a Work Session.
- Working on Matrices and Arrays.
  - Array Creation.
  - Array Operations.
  - Concatenation.
  - Complex Numbers.
  - Array Indexing.
- Workspace Variables.
- Character Strings.
- Calling Functions.
- 2-D Plots
- 3-D Plots
- Subplots

T.C. Adnan Menderes University Faculty of Engineering

Learning Objectives

- Understand and apply how to create, save, and run scripts.
- Understand and apply how to use help and documentation.
- Understand the language fundamentals.
  - Data types.
  - Conditional operators.
  - Input/display functions.
  - Understand and implement the basic programming structures.
    - Sequence structure.
    - Conditional structure.
    - Repetition structure.
- Understand and apply how to convert flowcharts and pseudocode into Matlab code.
- Understand how to write and invoke user-defined functions.

T.C. Adnan Menderes University Faculty of Engineering

Outlines

- Programming and Scripts
  - Sample Script
  - Help and Documentation
- Language Fundamentals
  - Data Types
  - Conditional Operators
  - Input/Disp Functions
  - Programming Structures
    - Sequence Structure
    - Conditional Structure
    - Repetition Structure
- Scripts vs. User-Defined Functions

T.C. Adnan Menderes University Faculty of Engineering

Programming and Scripts

Sample Script

- The simplest type of a Matlab program is called a **script**.
- A **script** is **a file** with a **.m** extension that contains multiple sequential lines of Matlab commands and function calls.

T.C. Adnan Menderes University Faculty of Engineering

Programming and Scripts

Sample Script

- The simplest type of a Matlab program is called a **script**.
- A **script** is **a file** with a **.m** extension that contains multiple sequential lines of Matlab commands and function calls.
- To create a new **script file** named **plotrand**, type at the command line the command **edit** followed by the name of the **script file**, as follows.

  edit plotrand

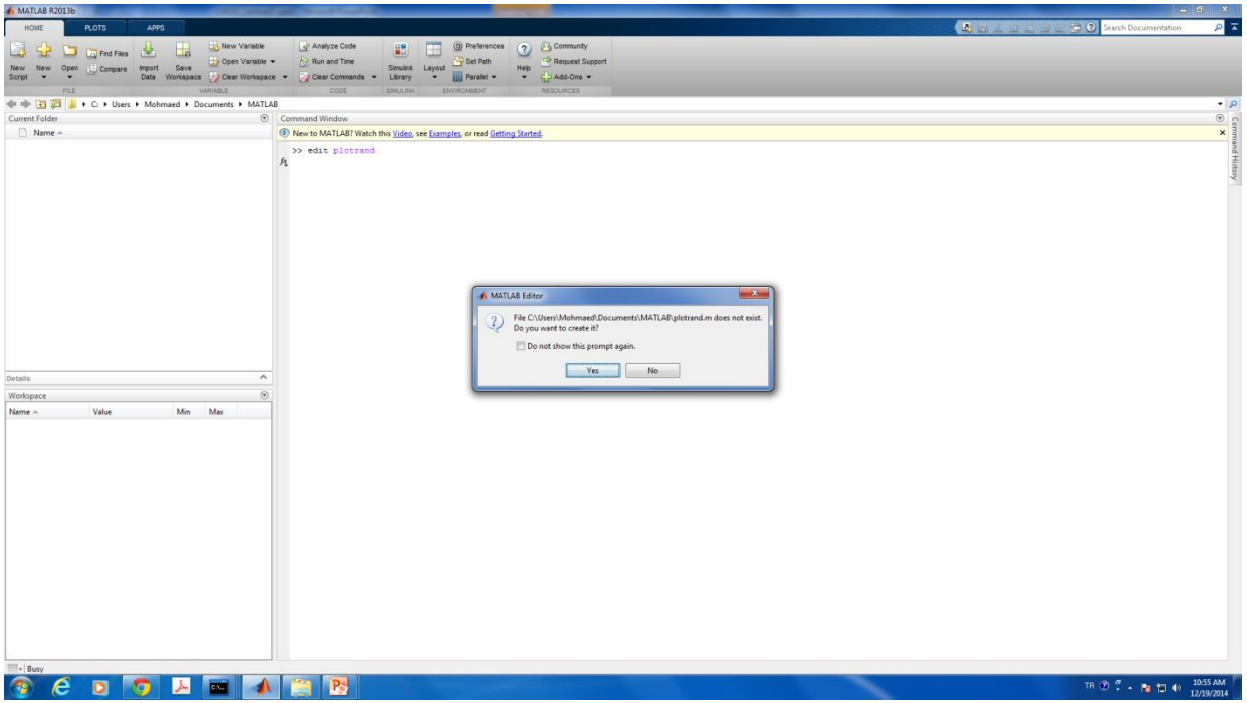
T.C. Adnan Menderes University Faculty of Engineering

Programming and Scripts

Sample Script

- The simplest type of a Matlab program is called a **script**.
- A **script** is **a file** with a **.m** extension that contains multiple sequential lines of Matlab commands and function calls.
- To create a new **script file** named **plotrand**, type at the command line the command **edit** followed by the name of the **script file**, as follows.

  edit plotrand



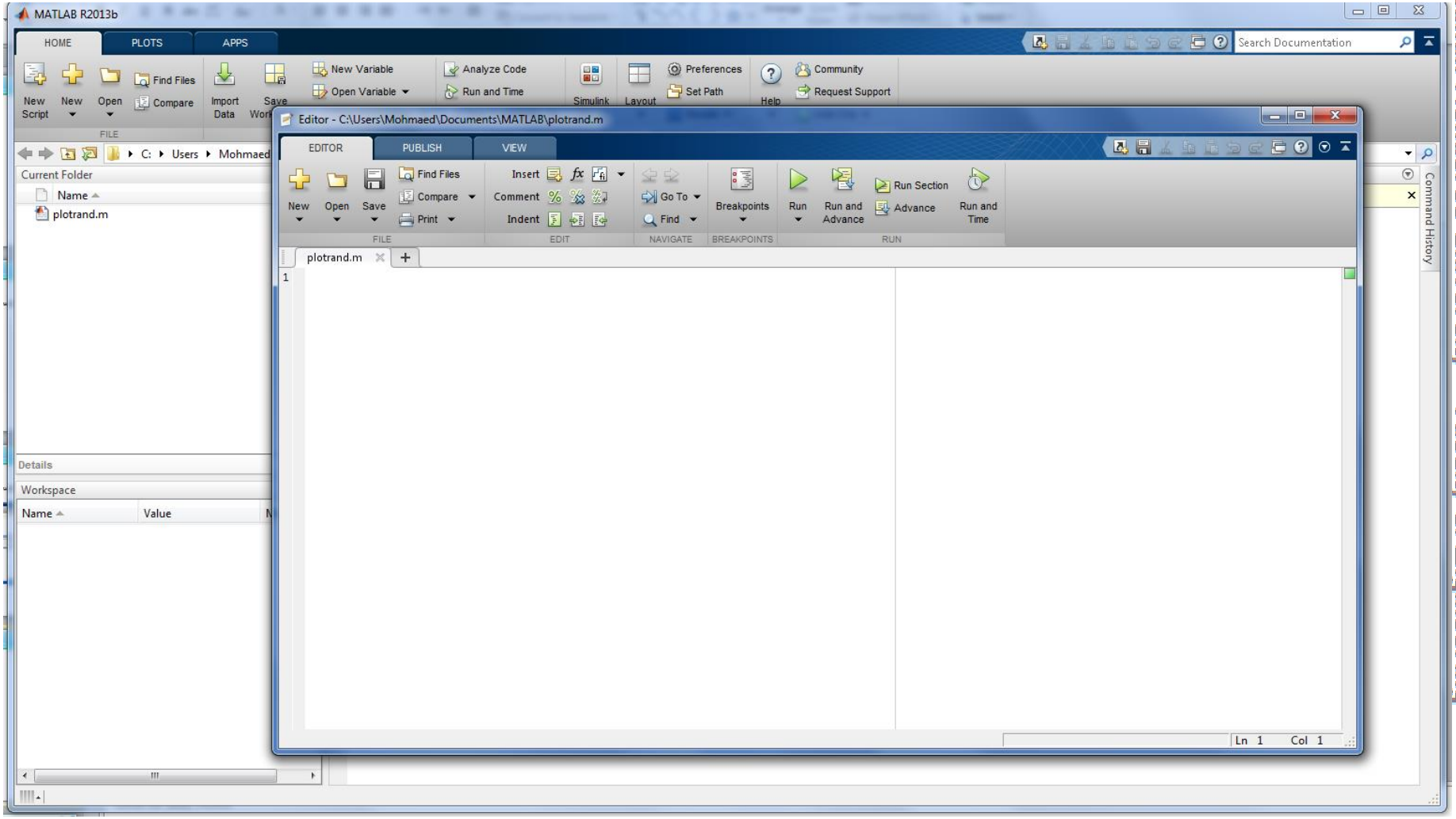
- **Matlab Editor** dialoge box will appear, asking you if you want to create a new file, select **Yes**.

T.C. Adnan Menderes University Faculty of Engineering

Programming and Scripts

Sample Script

- This opens a blank file named **plotrand.m**, that is stored at the Current Folder.



T.C. Adnan Menderes University Faculty of Engineering

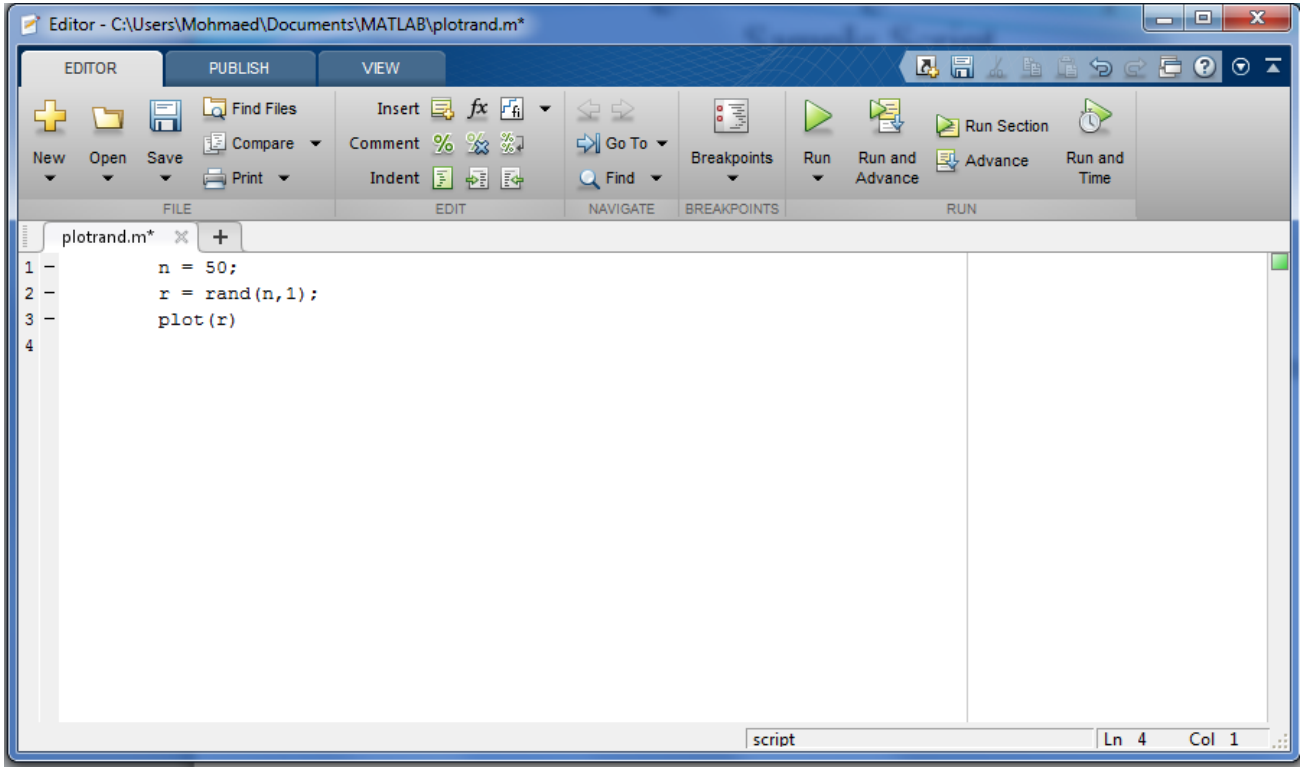
Programming and Scripts

Sample Script

- Enter some code that plots a vector of random data:

```
n = 50;
r = rand(n,1);
plot(r)
```

- Note: **rand (M,N)** returns an M-by-N matrix containing pseudorandom values drawn from the standard uniform distribution on the open interval (0,1)

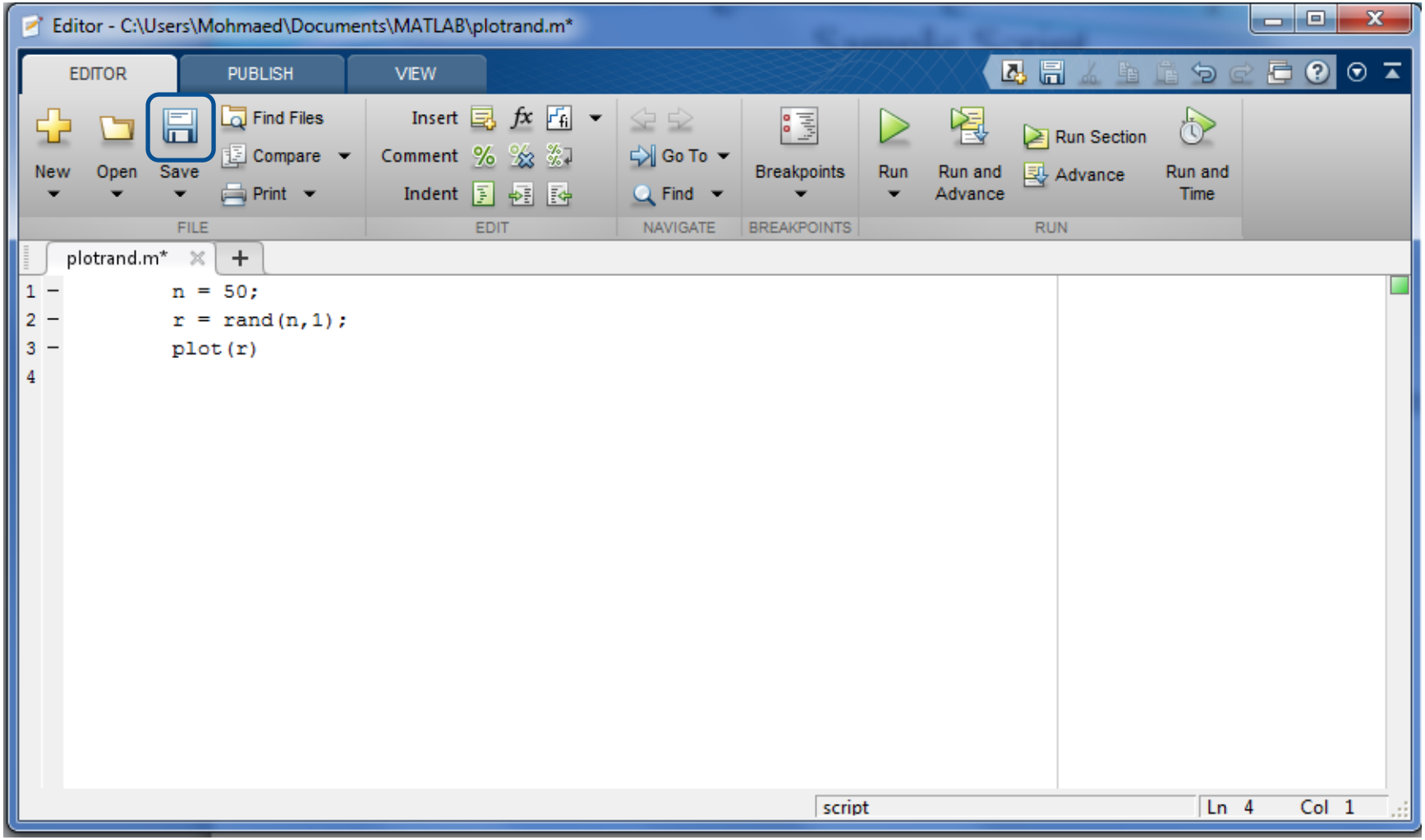


T.C. Adnan Menderes University Faculty of Engineering

Programming and Scripts

Sample Script

- Click **Save**

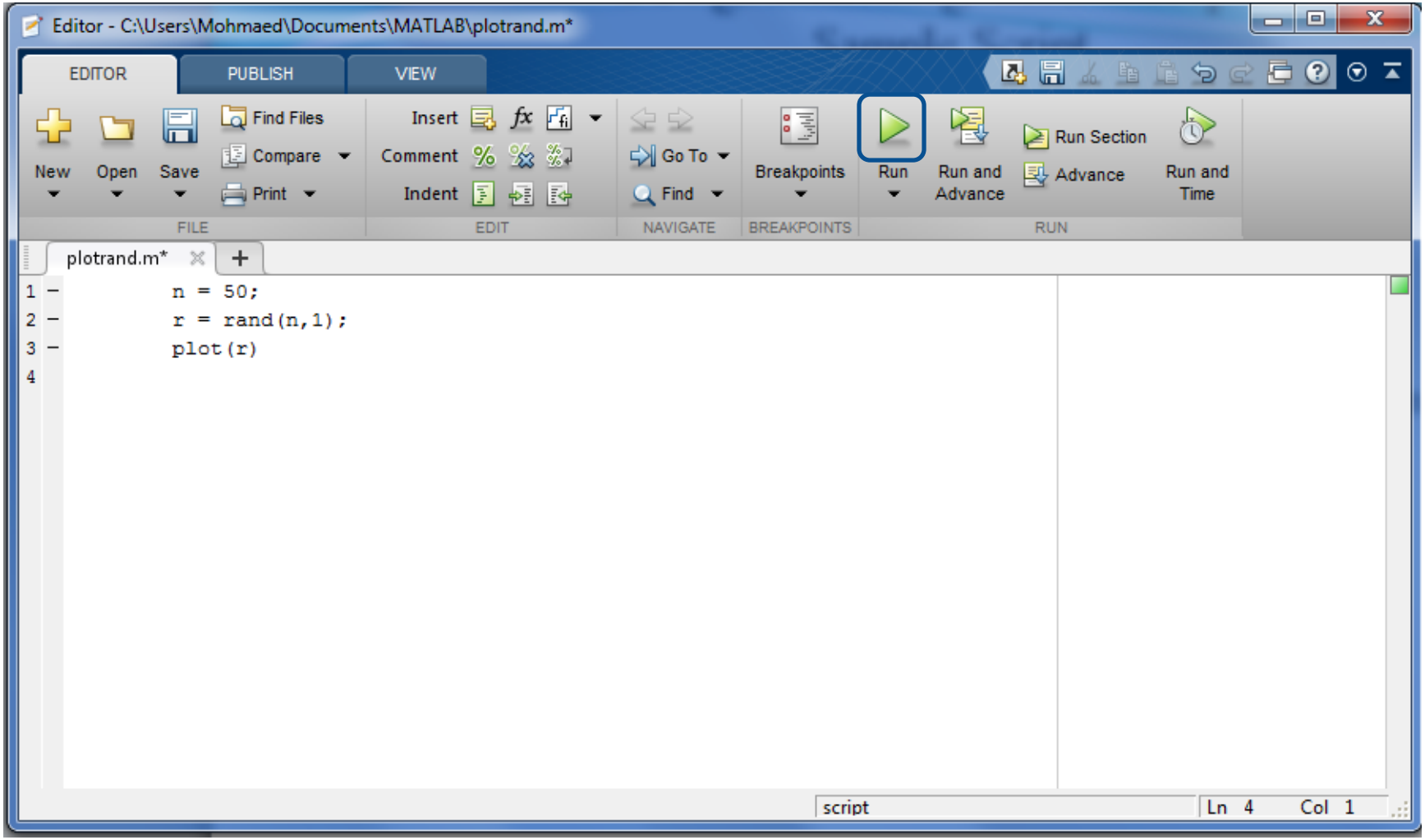


T.C. Adnan Menderes University Faculty of Engineering

Programming and Scripts

Sample Script

- To run the script file plotrand, click **Save and Run**, or type its name (plotrand) at the command line.

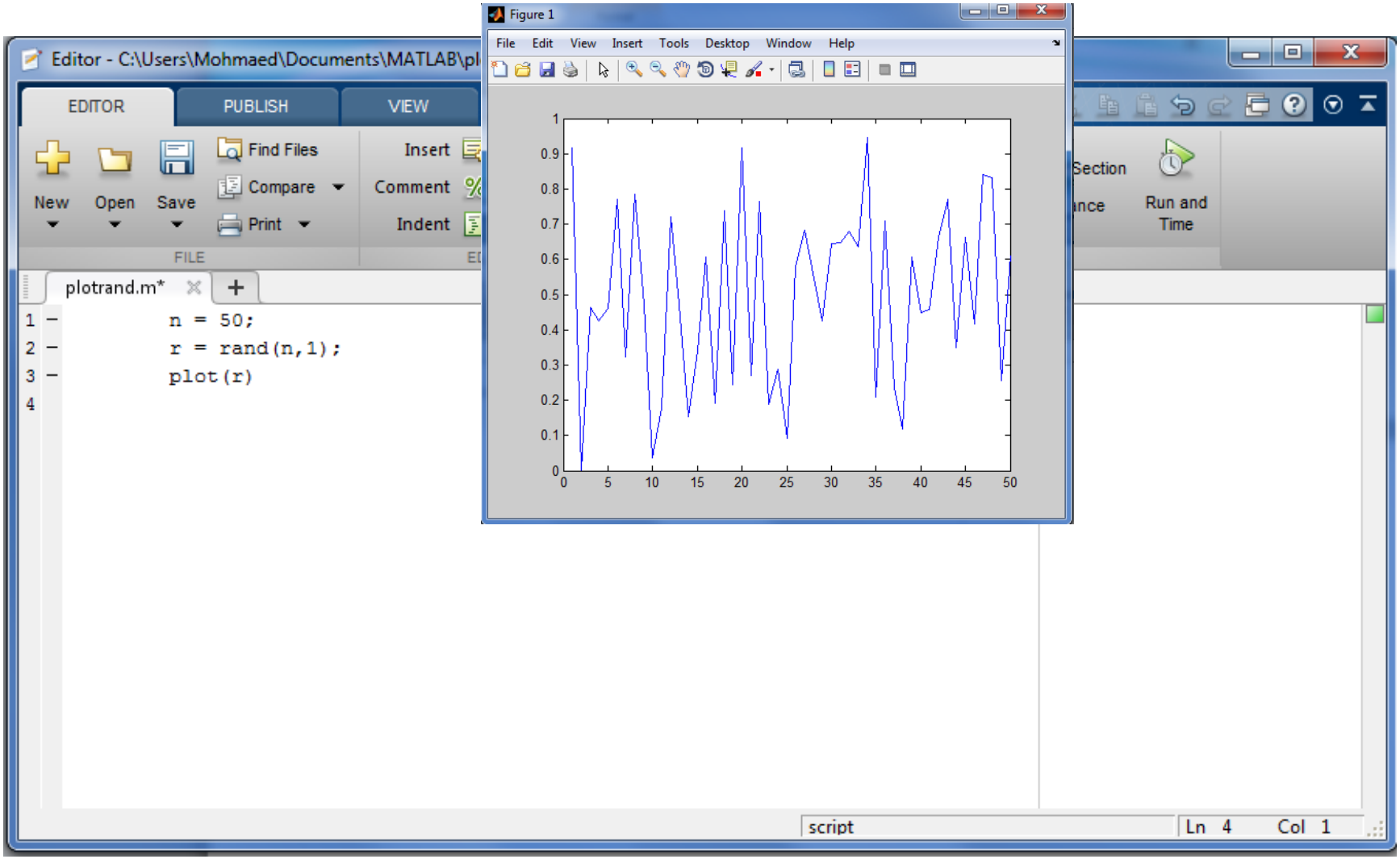


T.C. Adnan Menderes University Faculty of Engineering

Programming and Scripts

Sample Script

- The output window will appear on the screen.



T.C. Adnan Menderes University Faculty of Engineering

Programming and Scripts

Sample Script

- Next, add code that draws a horizontal line on the plot at the mean and adds a title to it:

```
m = mean(r);
hold on
plot([0,n],[m,m])
hold off
title('Mean of Random Uniform Data')
```

- Notes:
  - If you want to run the script file by typing its name at the command line, you should save it first.
  - Whenever you write a code, it is a good practice to add comments that describe the code.
  - Comments allow others to understand your code, and can refresh your memory when you return to it later.
  - Add comments using the percent **(%)** symbol.

T.C. Adnan Menderes University Faculty of Engineering

Programming and Scripts

Sample Script

- After adding comments, the final shape of the file will become like this:

```
n = 50;  % define a variable that holds the upper limit of the random values
r = rand(n,1); % Generate random data from a uniform distribution in the range [1,n]
plot(r) % Plot the random data
m = mean(r); % and calculate the mean
hold on
plot([0,n],[m,m]) % Draw a line from (0,m) to (n,m)
hold off  % keep the current figure
title('Mean of Random Uniform Data') % add a title to the plot
```

T.C. Adnan Menderes University Faculty of Engineering

Programming and Scripts

Help and Documentation

- All Matlab functions have supporting documentation that includes examples and describes the function inputs, outputs, and calling syntax.

T.C. Adnan Menderes University Faculty of Engineering

Programming and Scripts

Help and Documentation

- All Matlab functions have supporting documentation that includes examples and describes the function inputs, outputs, and calling syntax.
- There are several ways to access this information from the command line:

1. Open the function documentation in a separate window using the doc command.

   doc mean

T.C. Adnan Menderes University Faculty of Engineering

Programming and Scripts

Help and Documentation

- All Matlab functions have supporting documentation that includes examples and describes the function inputs, outputs, and calling syntax.
- There are several ways to access this information from the command line:

1. Open the function documentation in a separate window using the doc command.

   doc mean

2. Display function hints (the syntax portion of the function documentation) in the command window by pausing after you type the open parentheses for the function input arguments.

   mean(

T.C. Adnan Menderes University Faculty of Engineering

Programming and Scripts

Help and Documentation

- All Matlab functions have supporting documentation that includes examples and describes the function inputs, outputs, and calling syntax.
- There are several ways to access this information from the command line:

1. Open the function documentation in a separate window using the doc command.

   doc mean

2. Display function hints (the syntax portion of the function documentation) in the command window by pausing after you type the open parentheses for the function input arguments.

   mean(

3. View an abbreviated text version of the function documentation in the command window using the help command.

   help mean

T.C. Adnan Menderes University Faculty of Engineering

Programming and Scripts

Help and Documentation

- All Matlab functions have supporting documentation that includes examples and describes the function inputs, outputs, and calling syntax.
- There are several ways to access this information from the command line:

1. Open the function documentation in a separate window using the doc command.

   doc mean

2. Display function hints (the syntax portion of the function documentation) in the command window by pausing after you type the open parentheses for the function input arguments.

   mean(

3. View an abbreviated text version of the function documentation in the command window using the help command.

   help mean

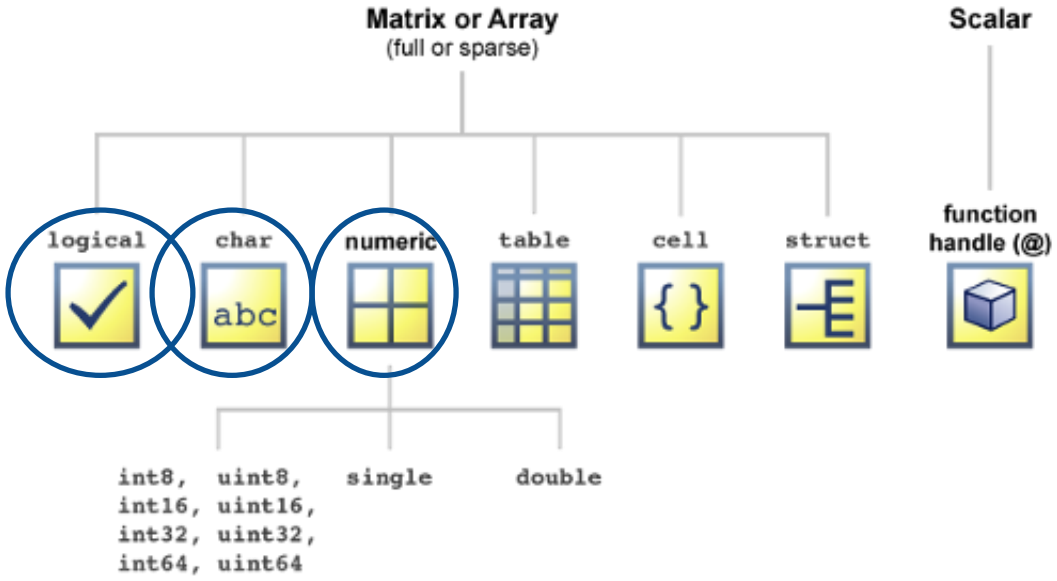
4. Access the complete product documentation by clicking the help icon.

T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

Data Types

All of the fundamental Matlab classes are shown in the diagram below:



T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

Data Types

double, single	Floating-Point Numbers	• Required for fractional numeric data. • Double and Single precision. • Use realmin and realmax to show range of values. • Default numeric type in Matlab.
int8, uint8, int16, uint16, int32, uint32, int64, uint64	Integers	• Use for signed and unsigned whole(integer) numbers. • More efficient use of memory. • Use intmin and intmax to show range of values. • Choose from 4 sizes (8, 16, 32, and 64 bits).
char	Characters and Strings	• Data type for text. • For multiple strings, use cell arrays.
logical	Logical Operations	• Use in conditions (relational operation) for testing. • Can have one of two values: true or false • Also useful in array indexing.

T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

Conditional Operators

- Conditional operators are usually used for comparing operands, using operators like "less than" and "not equal to" .
- The following table provides a summary:

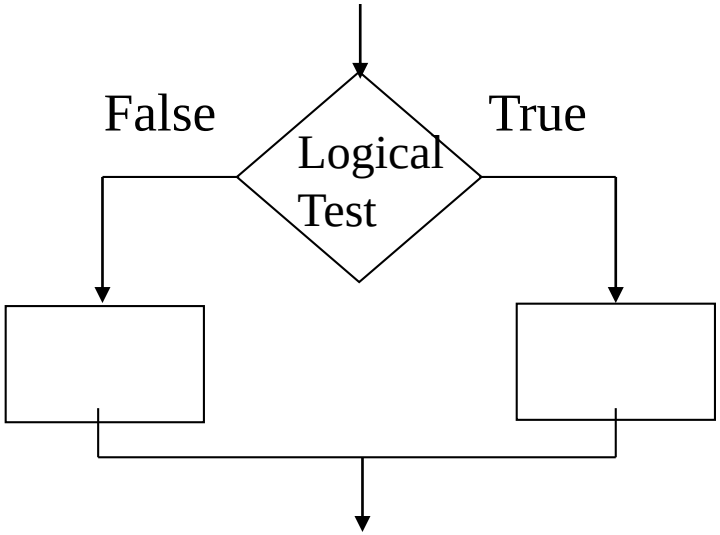
Operator	Description
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Equal to
~=	Not equal to

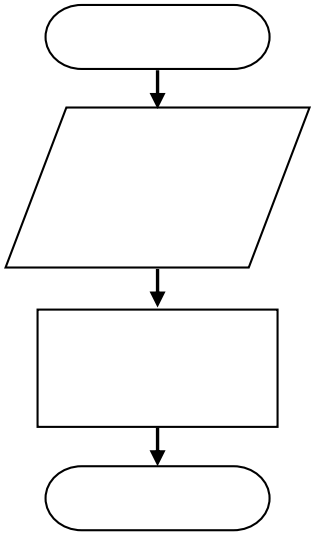
T.C. Adnan Menderes University Faculty of Engineering

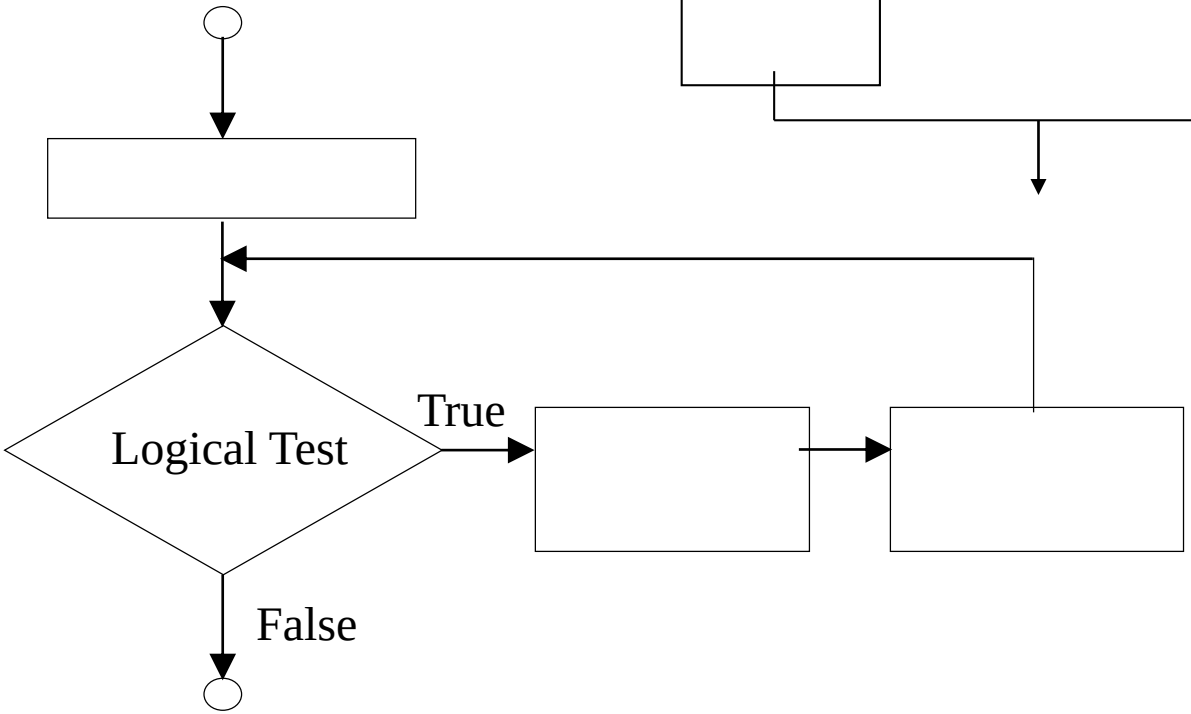
Language Fundamentals

Programming Structures

The three basic types of programming structures in pseudocode and flowcharts have their own implementations methods in Matlab.







T.C. Adnan Menderes University Faculty of Engineering

Converting Flowcharts into Matlab Commands

Matlab	Flowchart
input()	

T.C. Adnan Menderes University Faculty of Engineering

Converting Flowcharts into Matlab Commands

Matlab	Flowchart
disp()	

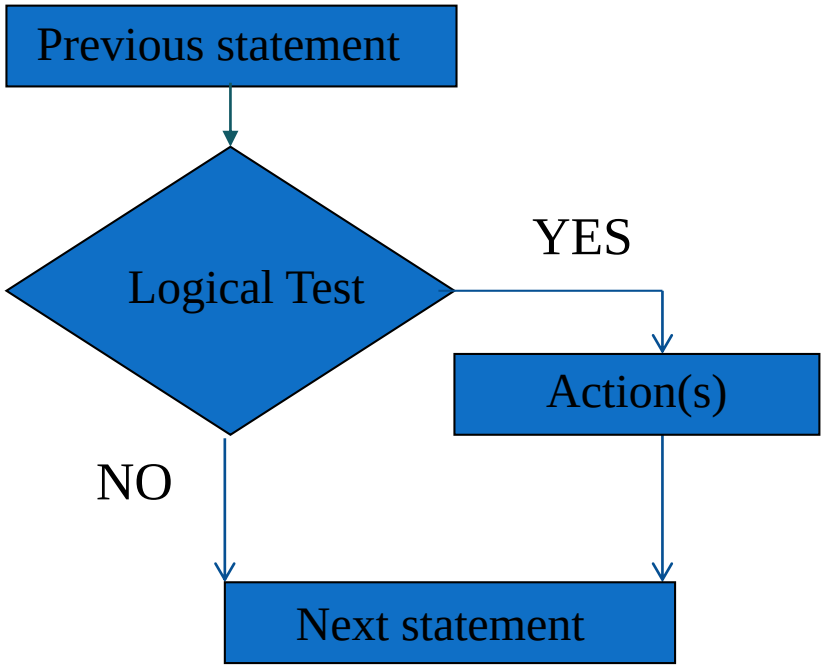
T.C. Adnan Menderes University Faculty of Engineering

Converting Flowcharts into Matlab Commands

Matlab	Flowchart
=,+,/,*,-	=,+,/,*,-

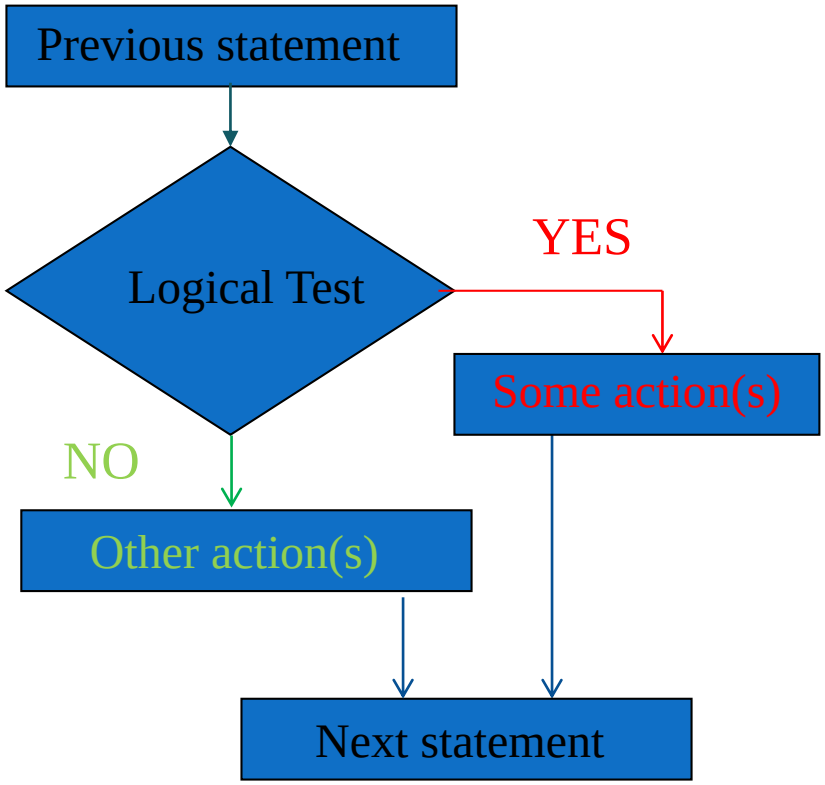
T.C. Adnan Menderes University Faculty of Engineering

Converting Flowcharts into Matlab Commands

Matlab	Flowchart
Previous statement If Logical Test Action(s) end Next statement	

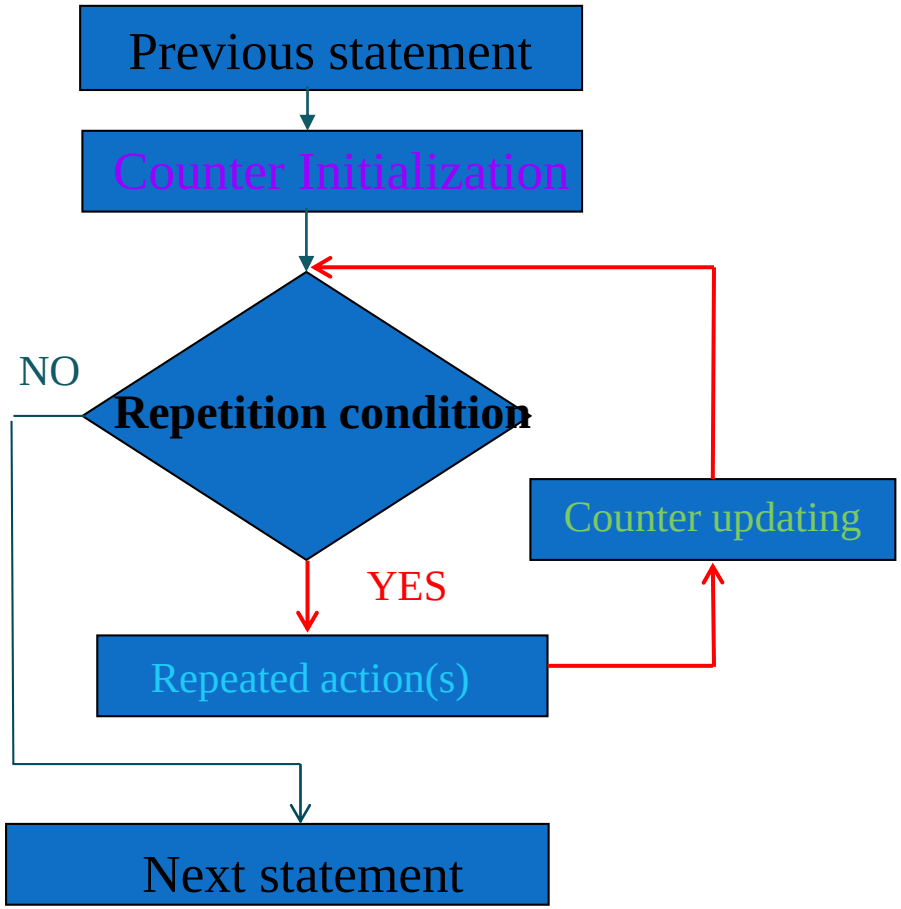
T.C. Adnan Menderes University Faculty of Engineering

Converting Flowcharts into Matlab Commands

Matlab	Flowchart
Previous statement if Logical Test Some action(s) else Other action(s) end Next statement	

T.C. Adnan Menderes University Faculty of Engineering

Converting Flowcharts into Matlab Commands

Matlab	Flowchart
Previous statement Counter Initialization while (**Repetition condition**) Repeated action(s) Counter updating end Next statement	

T.C. Adnan Menderes University Faculty of Engineering

Converting Flowcharts into Matlab Commands

Matlab

Previous statement

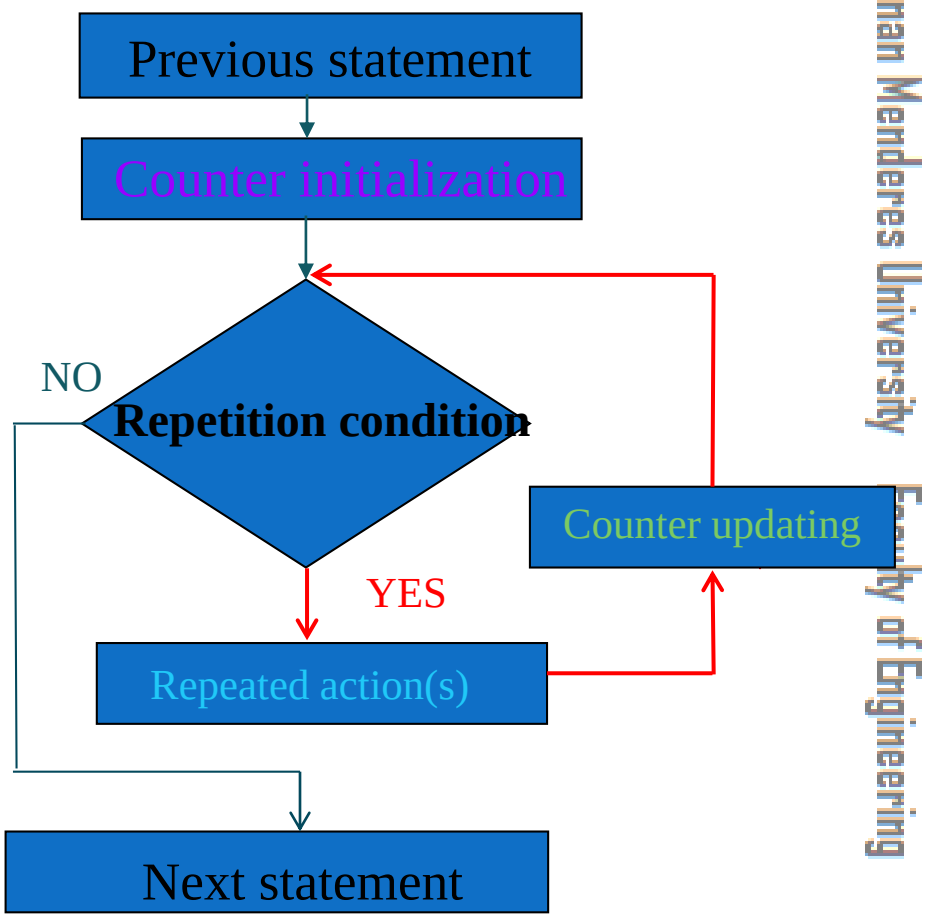
for Counter =intial value: step size:final value

Repeated action(s)

end

Next statement

Flowchart



T.C. Adnan Menderes University Faculty of Engineering

Converting Flowcharts into Matlab Commands

Matlab

Previous statement

for Counter =intial value: step size:final value

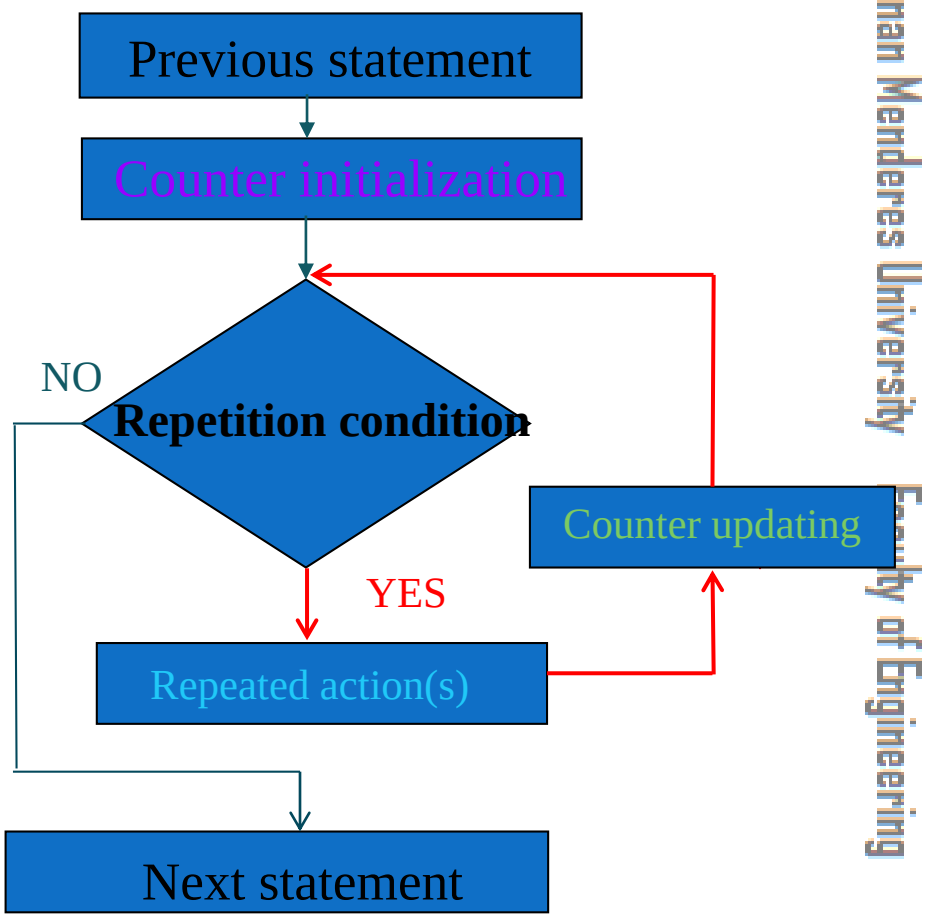
  Repeated action(s)

end

Next statement

- The step size is the amount on which counter should be incremented or decremented after each iteration.
  i.e. Counter = Counter - step size
  or, Counter = Counter + step size
- The Repetition condition is **Counter <=final value** and it is hidden from the user

Flowchart



T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Input Function

- **Syntax 1: result=input(prompt)**
  - **Description**:
    - Requests user input by displaying the prompt string on the screen, then waits for input from the keyboard, evaluates any expressions in the input, and then save the result to the specified variable.
    - If you press the **Enter** key without entering anything, then input returns an empty matrix.
    - If you enter an invalid expression at the prompt, then Matlab displays the relevant error message, and then redisplays the prompt.
  - **Example**:

    Prompt= 'how old are you: ';

    Age=input(Prompt)

T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Input Function

- **Syntax 1: result=input(prompt)**
  - **Description**:
    - Requests user input by displaying the prompt string on the screen, then waits for input from the keyboard, evaluates any expressions in the input, and then save the result to the specified variable.
    - If you press the **Enter** key without entering anything, then input returns an empty matrix.
    - If you enter an invalid expression at the prompt, then Matlab displays the relevant error message, and then redisplays the prompt.
  - **Example**:

    Prompt= 'how old are you: ';

    Age=input(Prompt)

- **Syntax 2: str = input(prompt,'s')**
  - **Description:**

    returns the entered text as a string, without evaluating expressions.
  - **Example:**

    Prompt='Do you like programming ';

    Feedback=input(Prompt, 's')

T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Disp Function

- **Syntax : disp(X)**
  - **Description:**
    - Displays the contents of X without printing the variable name, disp does not display empty variables.
  - **Examples:**
    - A = [15 150];

      disp(A)

T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Disp Function

- **Syntax : disp(X)**
  - **Description:**
    - Displays the contents of X without printing the variable name, disp does not display empty variables.
  - **Examples:**
    - A = [15 150];

      disp(A)
    - B = 'Hello World.';

      disp(B)

T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Disp Function

- **Syntax : disp(X)**
  - **Description:**
    - Displays the contents of X without printing the variable name, disp does not display empty variables.
  - **Examples:**
    - A = [15 150];

      disp(A)
    - B = 'Hello World.';

      disp(B)
    - name = 'Alice';

      Age = 12;

      C = [name,' will be ',num2str(Age),' this year.'];

      disp(C)

T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

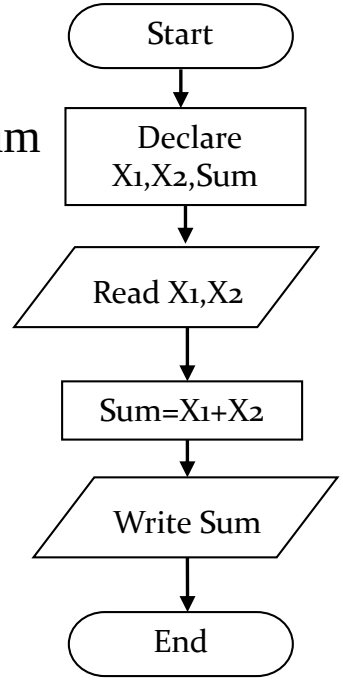
The Sequence Structure

T.C. Adnan Menderes University Faculty of Engineering

Problem1:
Write a Matlab code 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



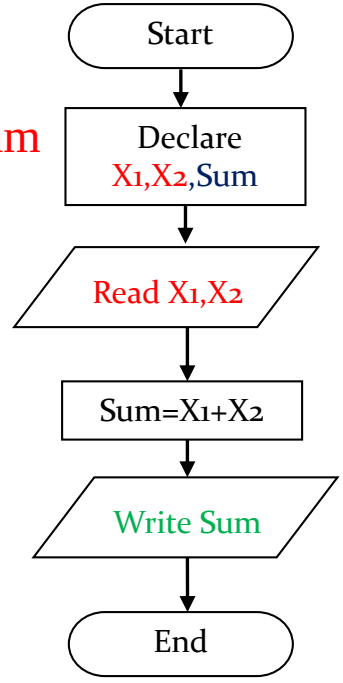
Language Fundamentals

The Sequence Structure

Problem1:
Write a Matlab code 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



Matlab code

```
X1=input ('Please enter the first ineger number: ');
X2=input ('Please enter the second ineger number: ');
Sum = X1+X2;
Message=['The result of adding ', num2str(X1), ' and ',
num2str(X2), ' is: ', num2str(Sum)];
disp (Message);
```

T.C. Adnan Menderes University Faculty of Engineering

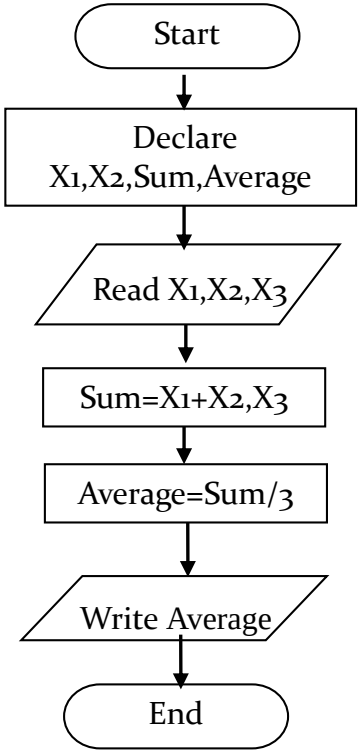
Language Fundamentals

The Sequence Structure

Problem 2:
Write a Matlab code 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

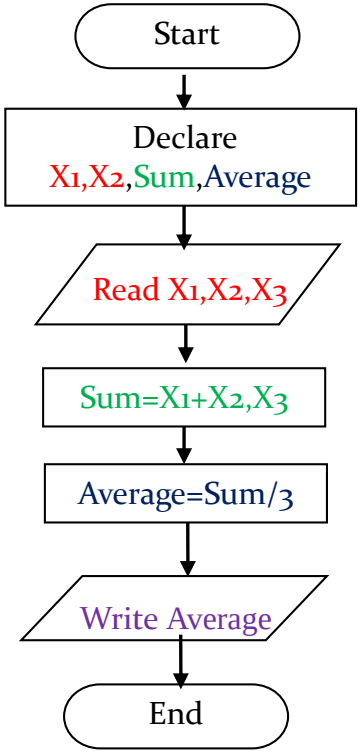
Language Fundamentals

The Sequence Structure

Problem 2:
Write a Matlab code 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



Matlab code

```
X1=input ('Please input the first number: ');
X2 =input ('Please input the second number: ');
X3=input ('Please input the third number: ');
Sum=X1+X2+X3;
Average= Sum/3;
Message=['The average of ', num2str(X1), ',',
num2str(X2),',and ',num2str(X3),' is:
',num2str(Average)];
disp (Message);
```

T.C. Adnan Menderes University Faculty of Engineering

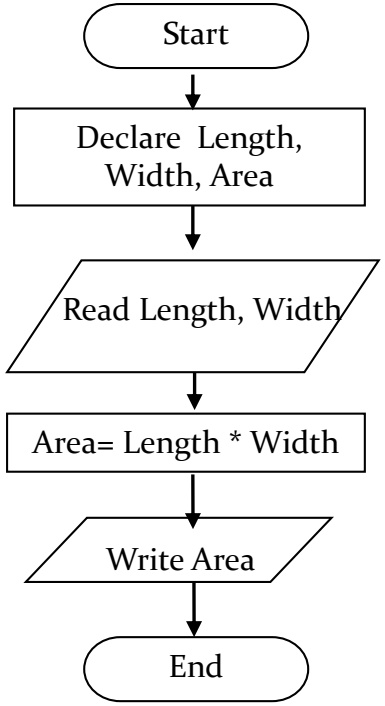
Language Fundamentals

The Sequence Structure

Problem 3:
Write a Matlab code for a program that calcultes 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

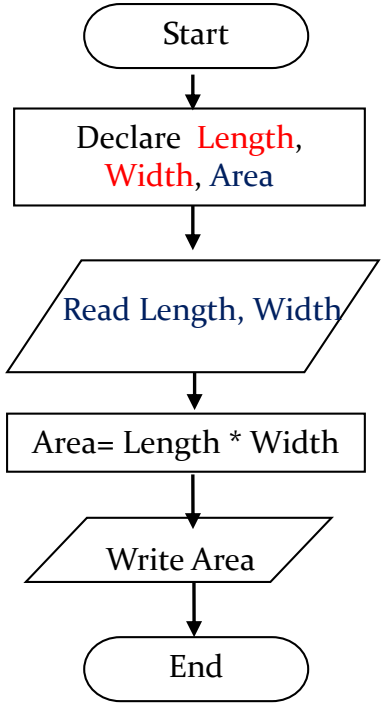
Language Fundamentals

The Sequence Structure

Problem 3:
Write a Matlab code for a program that calcultes the area of an rectangle.

Pseudocode:

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



Matlab code

```
Length =input ('Please input the length : ');
Width =input ('Please input the width : ');
Area= Length * Width;
Message=['The area is ', num2str(Area)];
disp (Message);
```

T.C. Adnan Menderes University Faculty of Engineering

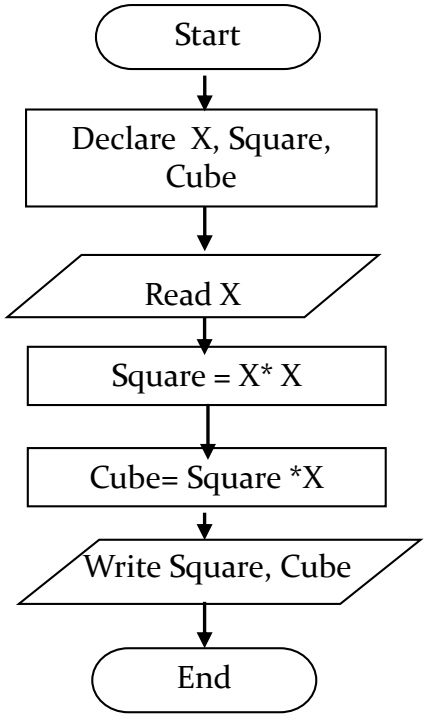
Language Fundamentals

The Sequence Structure

Problem 4:
Write a Matlab code 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

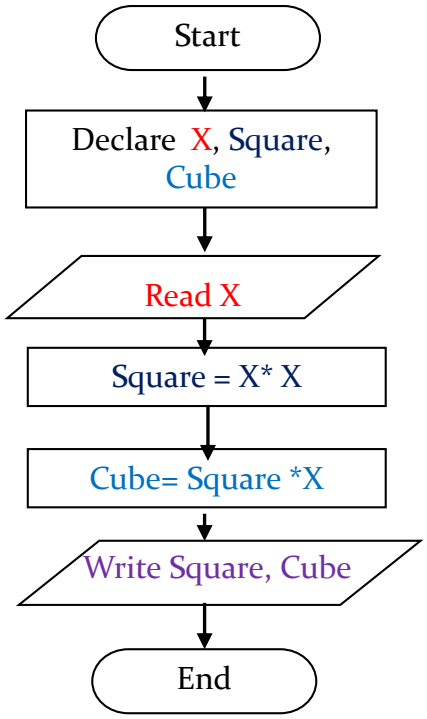
Language Fundamentals

The Sequence Structure

Problem 4:
Write a Matlab code 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



Matlab code

```
X=input ('Please input an integer number: ');
Square = X* X;
Cube= Square *X;
Message1=['The Square of  ', num2str(X), ' is: ', num2str(Square )];
Message2=['The Cube of  ', num2str(X), ' is: ', num2str(Cube )];
disp (Message1);
disp (Message2);
```

T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Conditional Structure

- Matlab offers many types of conditional structures.
- We will only cover the **if – then – else** structure as it is the most common.
  - **Syntax:**

```
if  logical test
    commands to execute if test is true
else
    commands to execute if test is false
end
```

T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Conditional Structure

- Matlab offers many types of conditional structures.
- We will only cover the **if – then – else** structure as it is the most common.
  - **Syntax:**

```
if  logical test
    commands to execute if test is true
else
    commands to execute if test is false
end
```

  - **Example**:

```
AbsenceHours=20;
if AbsenceHours>15
  disp ('fail')
else
  disp('Not yet')
end
```

T.C. Adnan Menderes University Faculty of Engineering

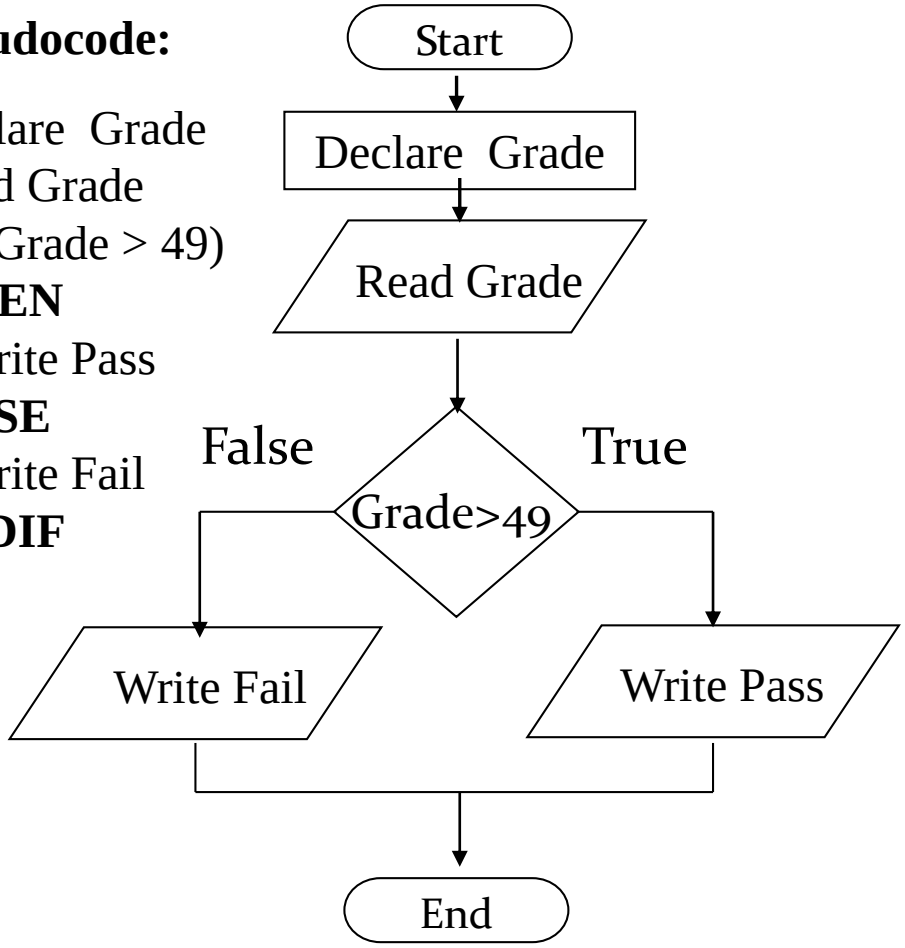
Language Fundamentals

The Conditional Structure

Problem 5:
Write a Matlab code 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

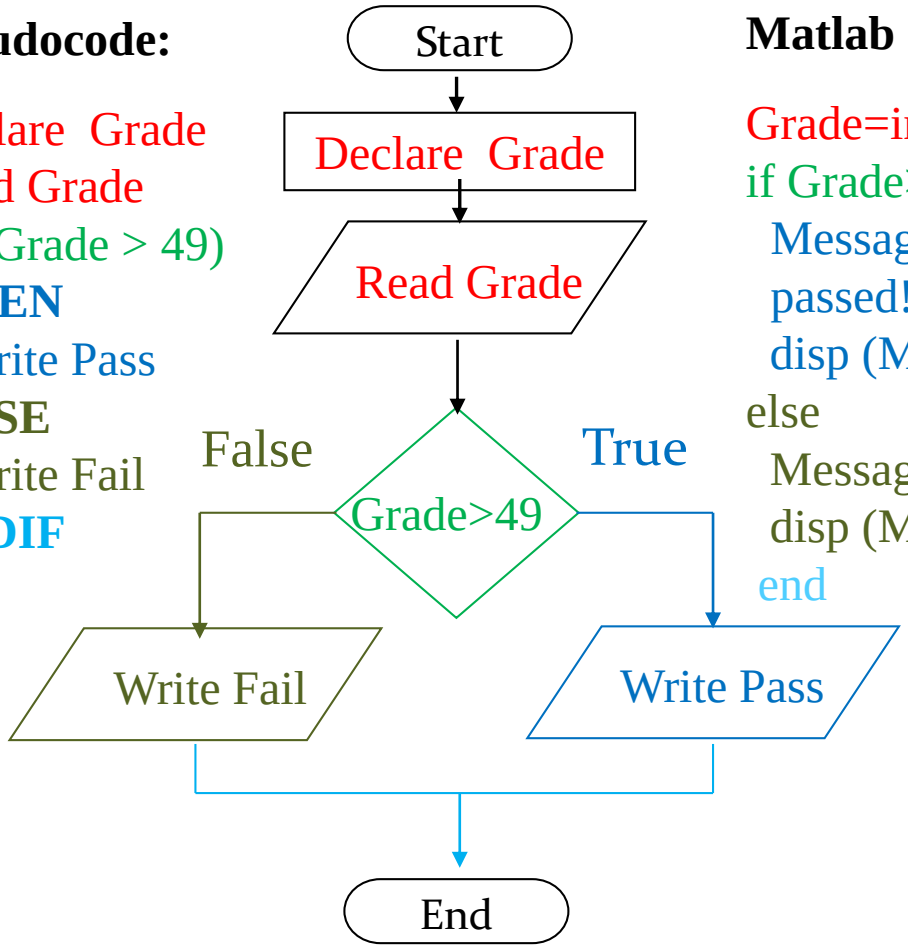
Language Fundamentals

The Conditional Structure

Problem 5:
Write a Matlab code 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
```



Matlab code:

```
Grade=input ('Please input a student's grade : ');
if Grade>49
  Message=['Congratulations, the student has
  passed!'];
  disp (Message);
else
  Message=['We are sorry, the student has failed'];
  disp (Message);
end
```

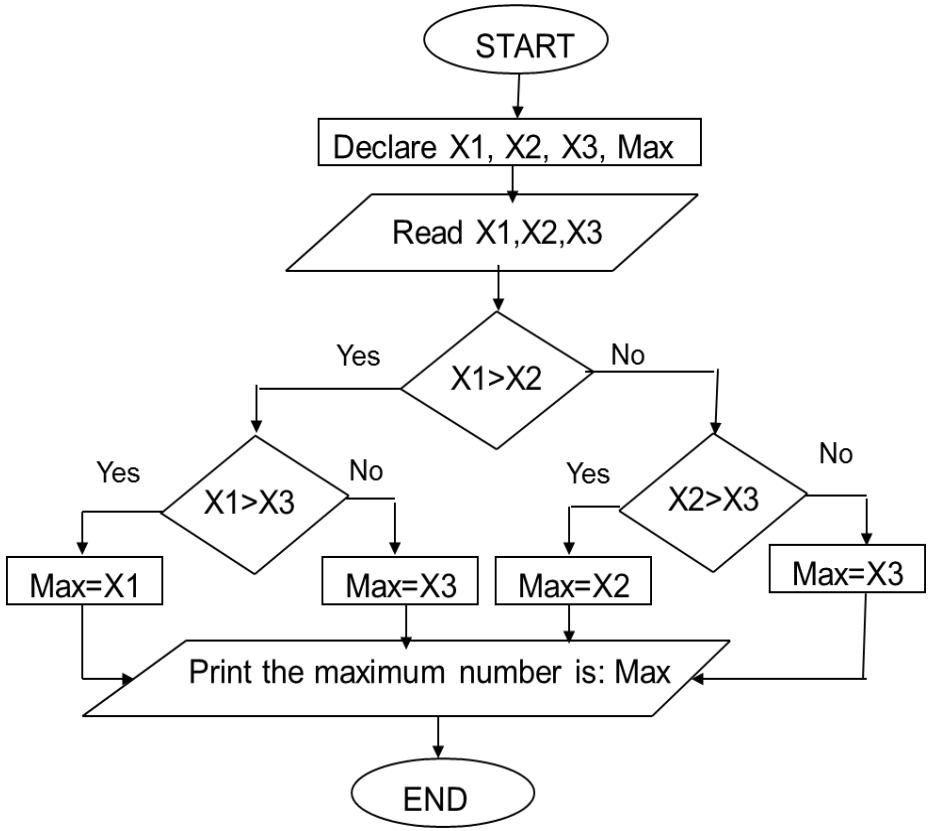
T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Conditional Structure

Problem 6:

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



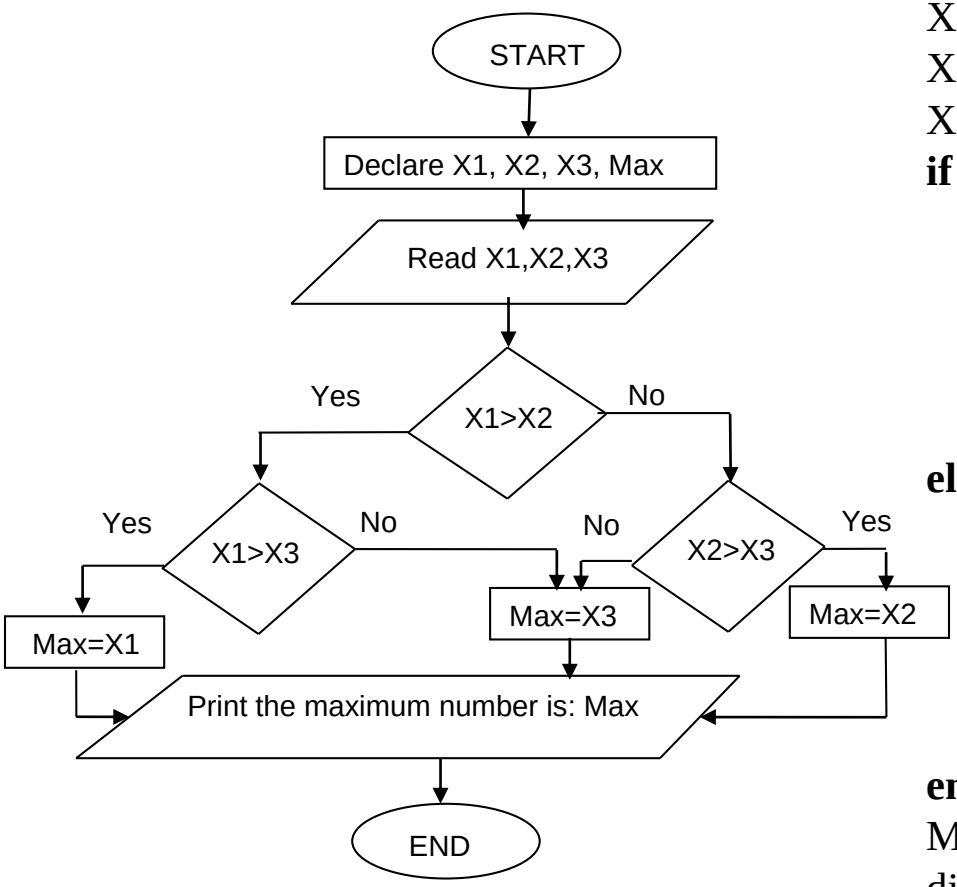
T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Conditional Structure

Problem 6:

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



Matlab code:

```
X1=input ('Please input the first number: ');
X2=input ('Please input the second number: ');
X3=input ('Please input the third number: ');
if X1>X2
  if X1>X3
    Max=X1;
  else
    Max=X3;
  end
else
  if X2>X3
    Max=X2;
  else
    Max=X3;
  end
end
Message=['The maximum number is: ', num2str(Max)];
disp (Message);
```

T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Repetition Structure

- Matlab offers many types of loops structures.
- We will only cover the **for Structure** as it is the most common.
  - **Syntax**:

```
for counter=start:step:end
    commands
end
```

Note: if you omit step from the for statement, Matlab will assume that step=1.

T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Repetition Structure

- Matlab offers many types of loops structures.
- We will only cover the **for Structure** as it is the most common.
  - **Syntax**:

```
for counter=start:step:end
    commands
end
```

Note: if you omit step from the for statement, Matlab will assume that step=1.

- **Example 1**: write a Matlab code for a program that prints the even numbers that exist in the range [0,100].
  - **Solution1:**

```
for counter=0:2:100
    disp(num2str(counter));
end
```

T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Repetition Structure

- Matlab offers many types of loops structures.
- We will only cover the **for Structure** as it is the most common.
  - **Syntax**:

```
for counter=start:step:end
    commands
end
```

Note: if you omit step from the for statement, Matlab will assume that step=1.

- **Example 1**: write a Matlab code for a program that prints the even numbers that exist in the range [0,100].
  - **Solution1:**

```
for counter=0:2:100
    disp(num2str(counter));
end
```

  - **Solution2:**

```
for counter=0:1:100
    if  mod(counter , 2) ==0
        disp(num2str(counter));
    end
end
```

T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Repetition Structure

- Matlab offers many types of loops structures.
- We will only cover the **for Structure** as it is the most common.
  - **Syntax**:

```
for counter=start:step:end
    commands
end
```

Note: if you omit step from the for statement, Matlab will assume that step=1.

- **Example 1**: write a Matlab code for a program that prints the even numbers that exist in the range [0,100].
  - **Solution1:**

```
for counter=0:2:100
    disp(num2str(counter));
end
```

  - **Solution2:**

```
for counter=0:1:100
    if  mod(counter , 2) ==0
        disp(num2str(counter));
    end
end
```

  - **Solution3:**

```
for counter=100:-1:0
    if  mod(counter , 2) ==0
        disp(num2str(counter));
    end
end
```

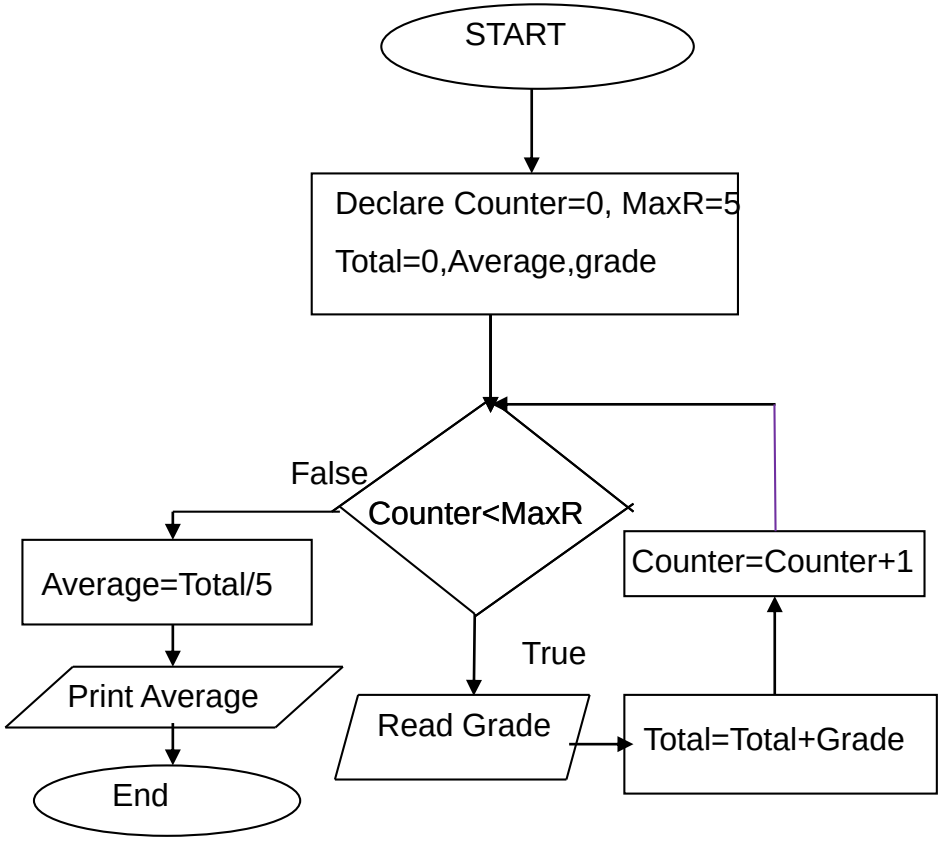
T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Repetition Structure

Problem 7:

Write a Matlab for a program that reads the grades of 5 students, finds the average of these grades, and then prints it.



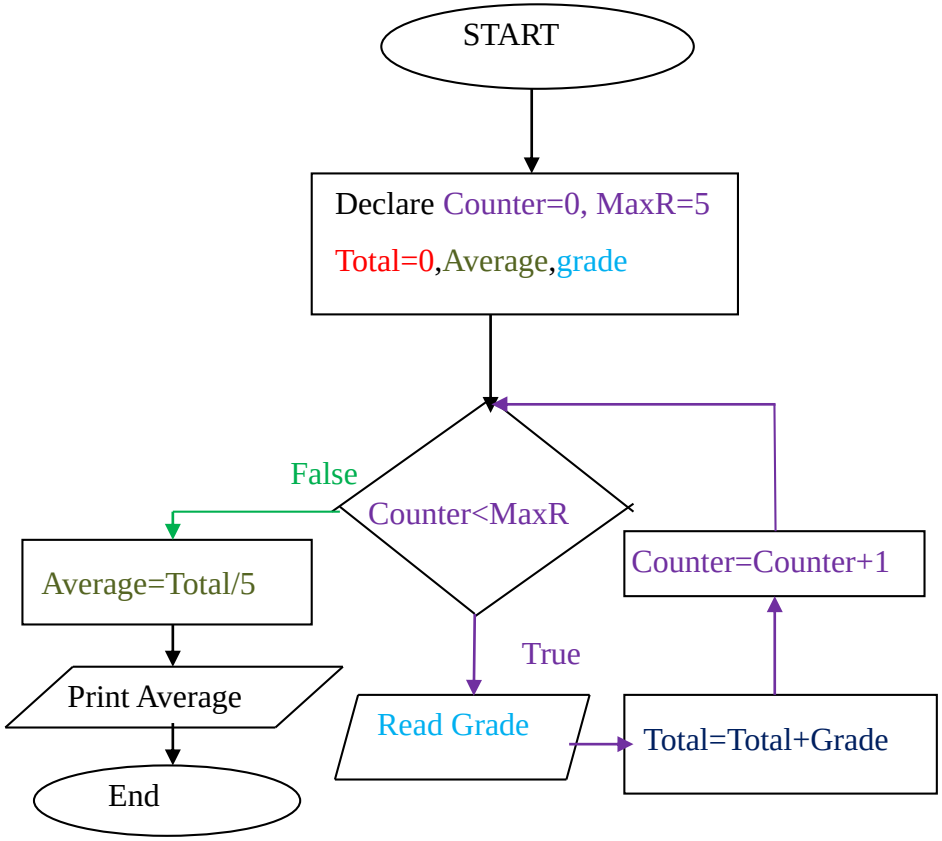
T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Repetition Structure

Problem 7:

Write a Matlab for a program that reads the grades of 5 students, finds the average of these grades, and then prints it.



Matlab code:

```
Total=0;
for Counter=1:5
  Grade=input ('Please input a student's grade: ');
  Total= Total+ Grade;
end
Average= Total/5;
Message=['The average : ', num2str(Average)];
disp (Message);
```

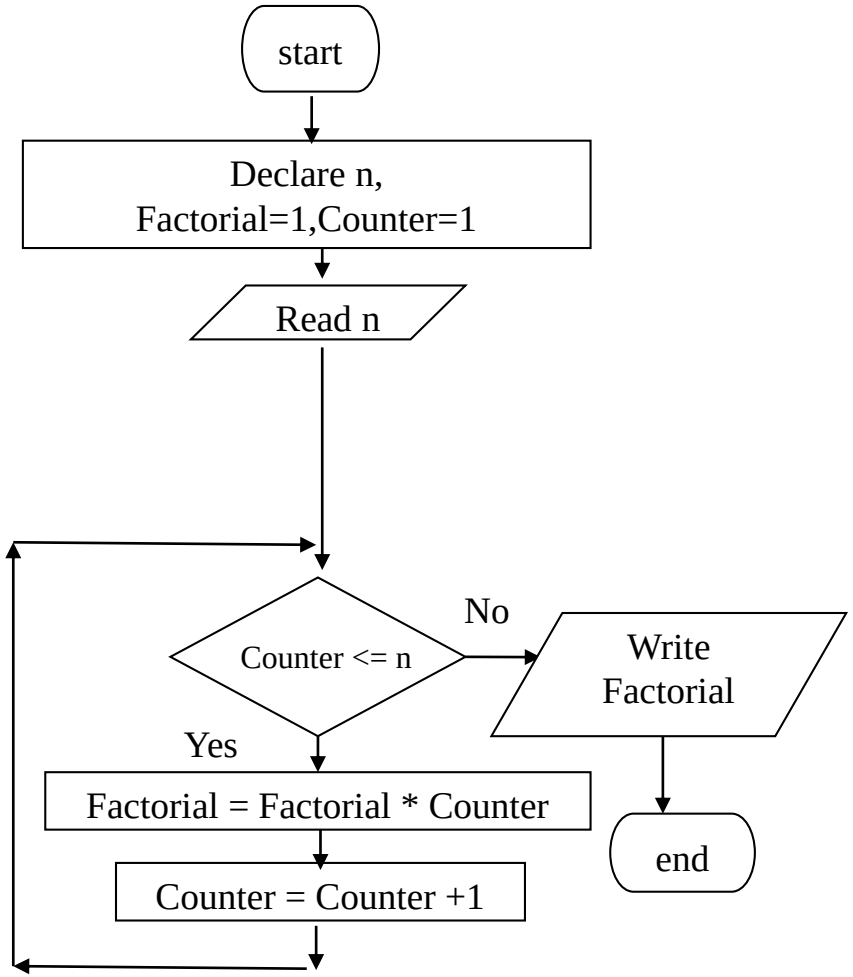
T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Repetition Structure

Problem 8:

Write a Matlab code 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



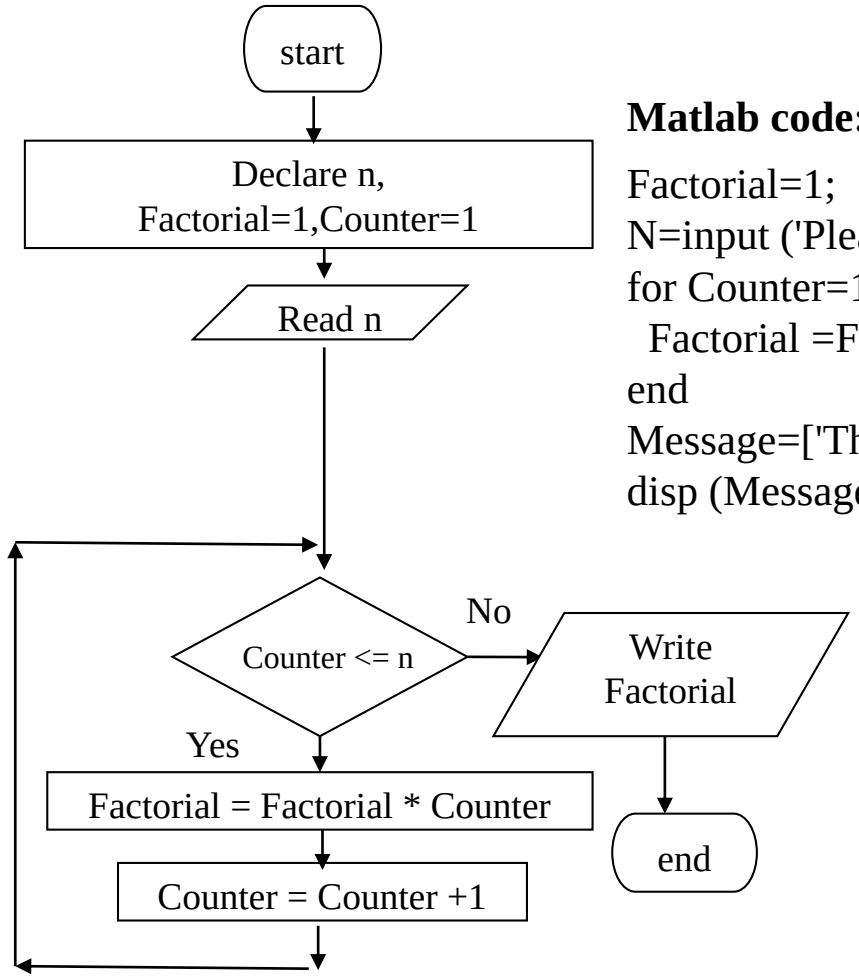
T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Repetition Structure

Problem 8:

Write a Matlab code 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



Matlab code:

```
Factorial=1;
N=input ('Please input an integer number: ');
for Counter=1:N
  Factorial =Factorial *Counter;
end
Message=['The factorial of:  ', num2str(N), 'is:  ' ,num2str(Factorial )];
disp (Message);
```

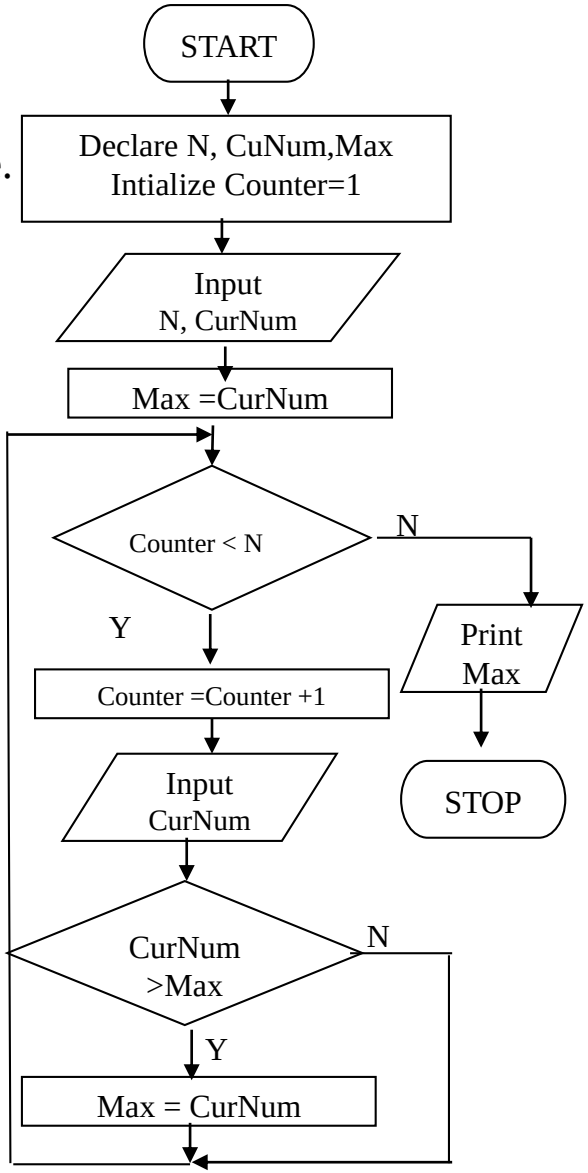
T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Repetition Structure

Problem 9:

Write a Matlab code for a program that finds and prints the largest number of N numbers (N can be any number). Read numbers one by one.



T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

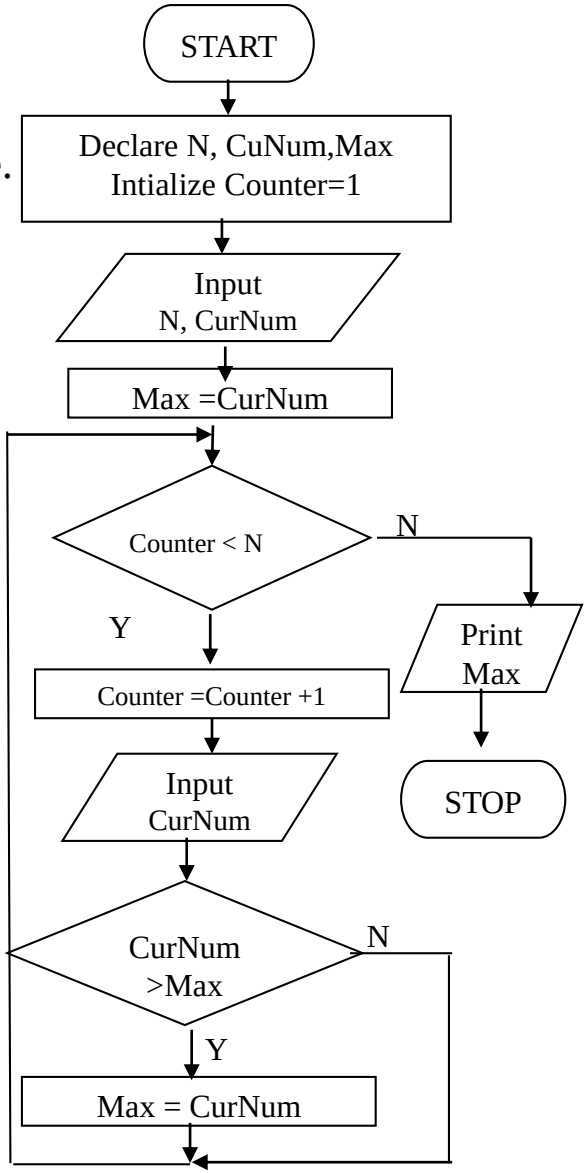
The Repetition Structure

Problem 9:

Write a Matlab code for a program that finds and prints the largest number of N numbers (N can be any number). Read numbers one by one.

Matlab code:

```
N=input ('Please enter the number of numbers you are going to enter: ');
CurNum=input ('Please enter a number: ');
Max=CurNum;
for Counter=1:N-1
  CurNum=input ('Please enter a number: ');
  if CurNum> Max
    Max= CurNum;
  end
end
Message=['The maximum number is  ', num2str(Max)];
disp (Message);
```



T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Repetition Structure

Problem 10:

Draw a flowchart and write a Matlab code for a program that reads two integer numbers N1&N2 from the user, then calculates and prints the $N1^{N2}$ (N1 raised to the power N2)..

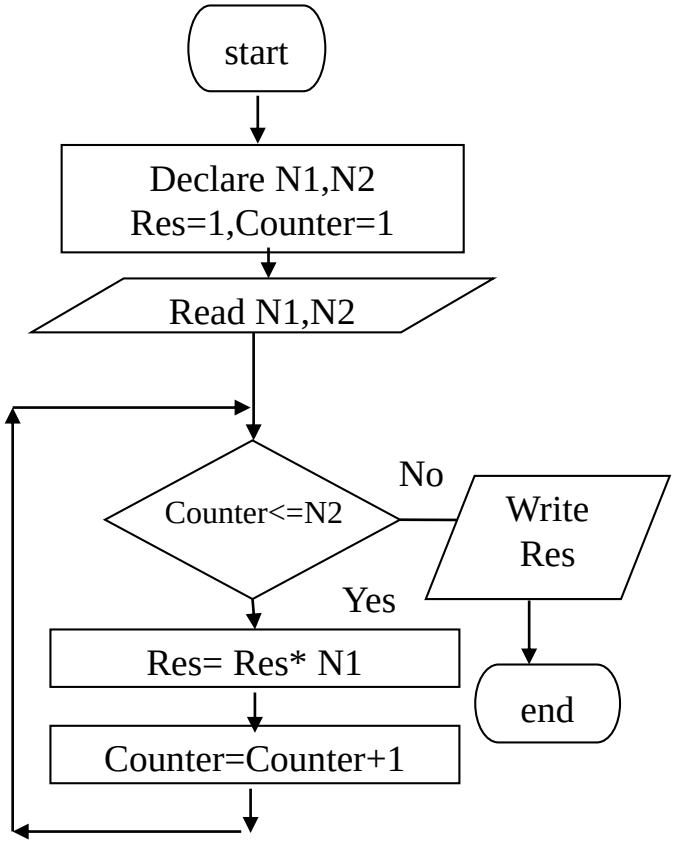
T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Repetition Structure

Problem 10:

Draw a flowchart and write a Matlab code for a program that reads two integer numbers N1&N2 from the user, then calculates and prints the $N1^{N2}$ (N1 raised to the power N2)..



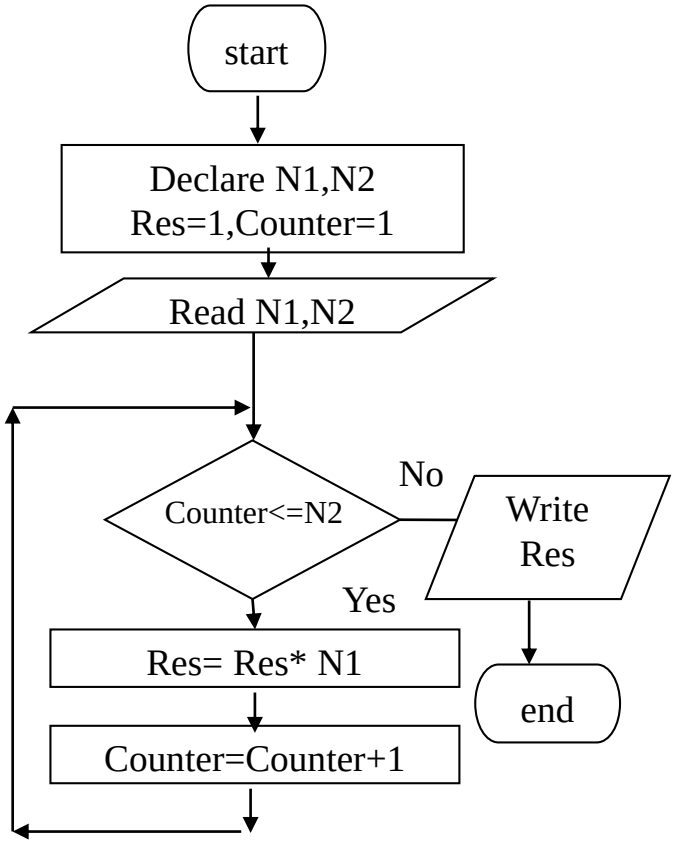
T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Repetition Structure

Problem 10:

Draw a flowchart and write a Matlab code for a program that reads two integer numbers N1&N2 from the user, then calculates and prints the $N1^{N2}$ (N1 raised to the power N2)..



Matlab code:

```
Res=1;
N1=input ('Please enter the base: ');
N2=input ('Please enter the exponent: ');
for Counter=1:N2
  Res=Res*N1;
end
Message=['The result is  ', num2str(Res)];
disp (Message);
```

T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Repetition Structure

Problem 11:

Draw a flowchart and write a Matlab code 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.

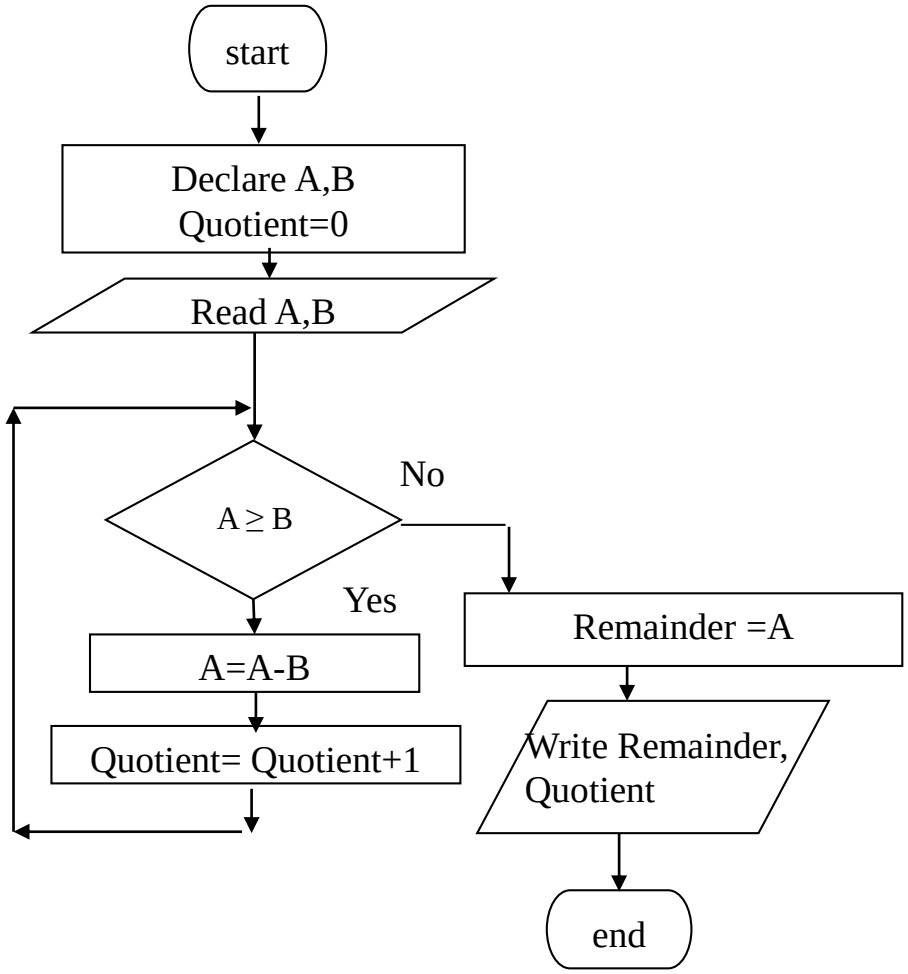
T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Repetition Structure

Problem 11:

Draw a flowchart and write a Matlab code 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.



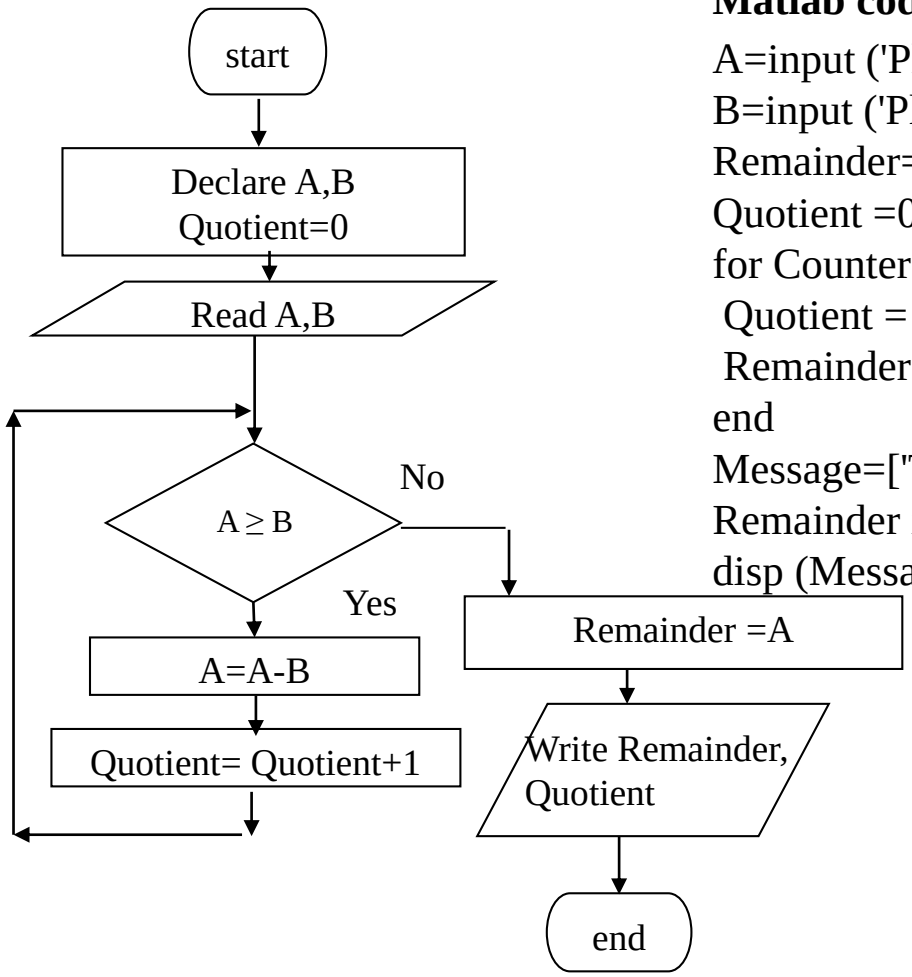
T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Repetition Structure

Problem 11:

Draw a flowchart and write a Matlab code 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.



Matlab code:

```
A=input ('Please enter the first nonnegative integer number: ');
B=input ('Please enter the second nonnegative integer number: ');
Remainder=A;
Quotient =0;
for Counter=A:-B:B
 Quotient = Quotient +1;
 Remainder=Remainder-B;
end
Message=['The Quotient is  ', num2str(Quotient), '  The
Remainder is  ', num2str(Remainder)];
disp (Message);
```

T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Repetition Structure

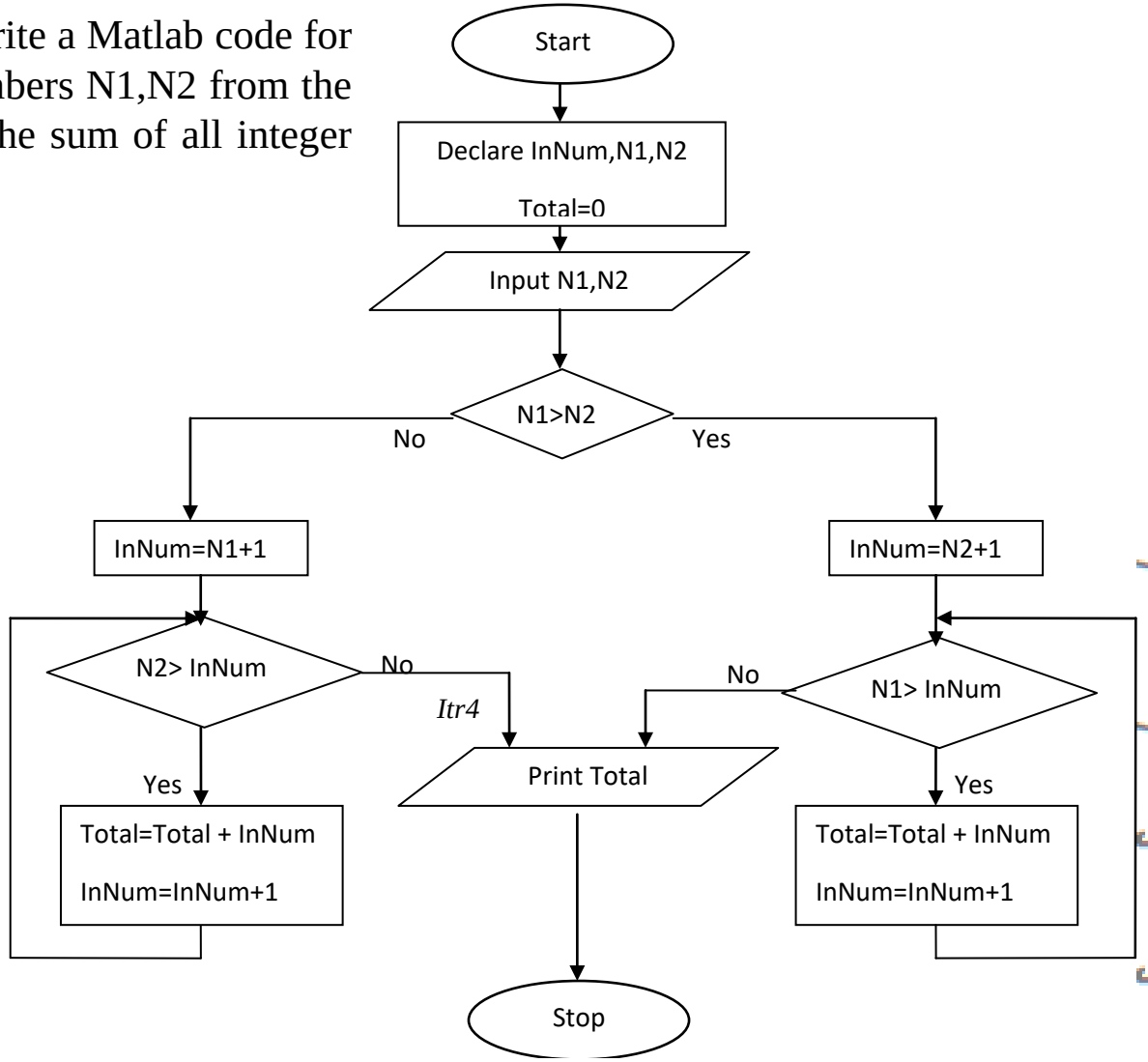
Problem 12: Draw a flowchart and write a Matlab code for a program that reads two integer numbers N1,N2 from the user, and then calculates and prints the sum of all integer numbers that exist between them.

T.C. Adnan Menderes University Faculty of Engineering

Language Fundamentals

The Repetition Structure

Problem 12: Draw a flowchart and write a Matlab code for a program that reads two integer numbers N1,N2 from the user, and then calculates and prints the sum of all integer numbers that exist between them.



T.C. Adnan Menderes University Faculty of Engineering

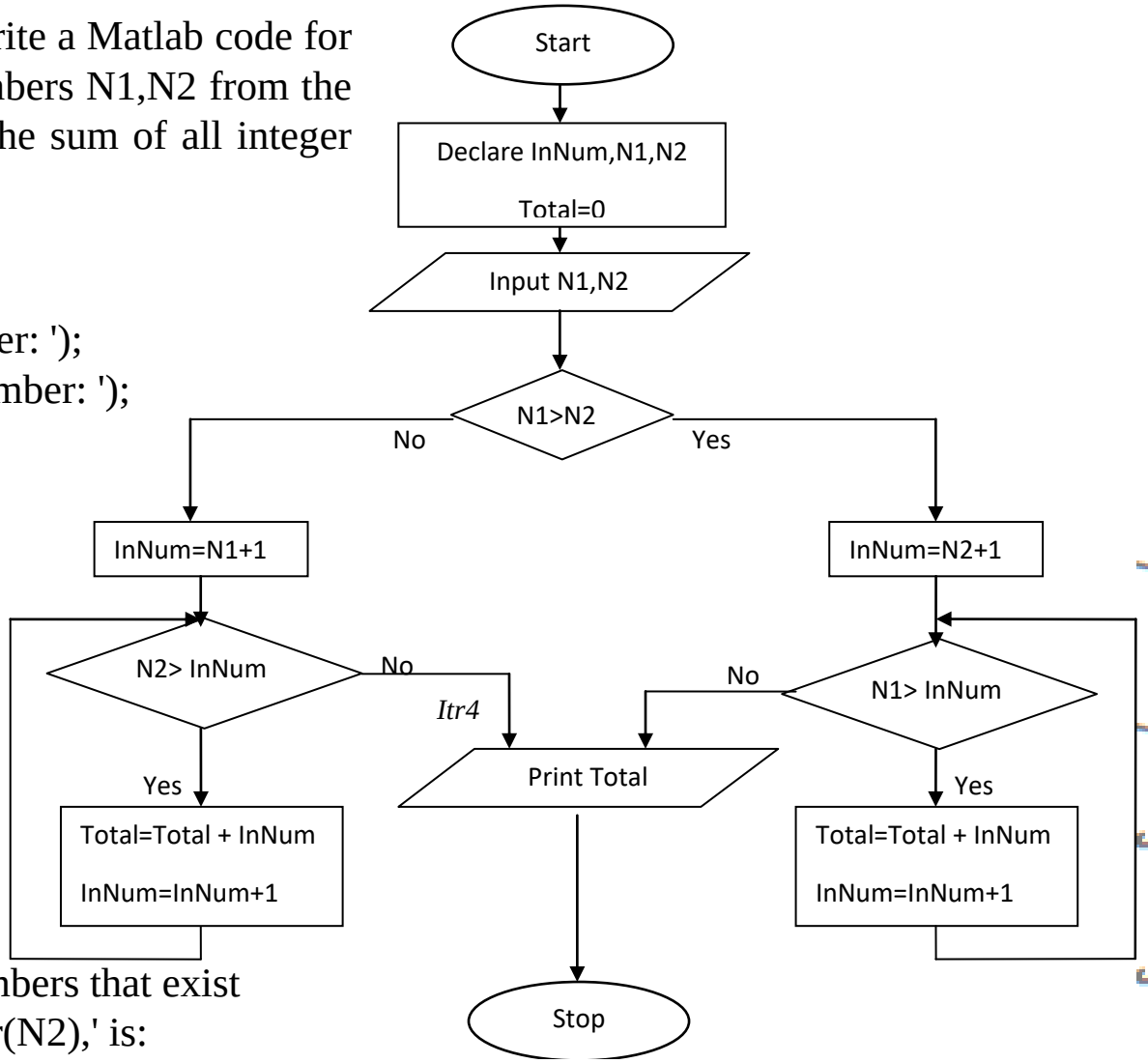
Language Fundamentals

The Repetition Structure

Problem 12: Draw a flowchart and write a Matlab code for a program that reads two integer numbers N1,N2 from the user, and then calculates and prints the sum of all integer numbers that exist between them.

Matlab code:

```
Total=0;
N1=input('Please input the first number: ');
N2=input('Please input the second number: ');
if N1>N2
  InNum=N2+1;
  for counter=InNum:N1-1;
    Total=Total+counter;
  end
else
  InNum=N1+1;
  for counter=InNum:N2-1;
   Total=Total+counter;
  end
end
Message=['The sum of all integer numbers that exist
between ',num2str(N1),' and ',num2str(N2),' is:
',num2str(Total)];
disp(Message)
```



T.C. Adnan Menderes University Faculty of Engineering

Scripts vs. User-Defined Functions

- User-defined functions are like scripts, **except** that they have the advantage of **accepting input arguments and producing output arguments.**
- **Syntax**:

  function **[y1,...,yN]** = myfun **(x1,...,xM)**
- **Description:**
  - Declares a function named myfun that **accepts inputs x1,...,xM** and **returns output y1,...,yN**. **This declaration statement must be the first executable line of the function.**
  - **The name of the file should match the name of the first function in the file**. Valid function names begin with an alphabetic character, and can contain letters, numbers, or underscores.
  - Files can include multiple local functions or nested functions. Use the end keyword to indicate the end of each function in a file if:
    - Any function in the file contains a nested function.
    - Any local function in the file uses the end keyword.

  Otherwise, the end keyword is optional.

http://www.mathworks.com/help/matlab/ref/function.html

T.C. Adnan Menderes University Faculty of Engineering

User-Defined Functions Examples
Function with One Output

- Define a function in a file named average.m that accepts an input vector, calculates the average of the values, and returns a single result.

Keyword: function

Output Argument(s)

Function Name (same as file name .m)

```
function y = average(x)
% input A vector of numbers
% output the average value of those numbers
 if ~isvector(x)
error('Input must be a vector')
end
y = sum(x)/length(x);
end
```

Input Argument(s)

Comments

MATLAB Code

- Call the function from the command line.

```
z = 1:99; average(z)
ans = 50
```


T.C. Adnan Menderes University Faculty of Engineering

User-Defined Functions Examples
Function with Multiple Outputs

- Define a function in a file named stat.m that returns the mean and standard deviation of an input vector.

```
function [m,s] = stat(x)
 n = length(x);
 m = sum(x)/n;
 s = sqrt(sum((x-m).^2/n));
 end
```

- Call the function from the command line.

```
values = [12.7, 45.4, 98.9, 26.6, 53.1];

[ave,stdev] = stat(values)

ave = 47.3400

stdev = 29.4124
```

http://www.mathworks.com/help/matlab/ref/function.html

T.C. Adnan Menderes University Faculty of Engineering

Functions Examples
Multiple Functions in a File

- Define two functions in a file named stat2.m, where the first function calls the second.

```
function [m,s] = stat2(x)
n = length(x);
m = avg(x,n);
s = sqrt(sum((x-m).^2/n));
end
function m = avg(x,n)
m = sum(x)/n;
end
```

- Call the function stat2.m, from the command line.

```
values = [12.7, 45.4, 98.9, 26.6, 53.1];
[ave,stdev] = stat2(values)
ave = 47.3400
stdev = 29.4124
```

http://www.mathworks.com/help/matlab/ref/function.html

T.C. Adnan Menderes University Faculty of Engineering

User-Defined Functions Examples
Multiple Functions in Multible Files

- Define two functions in 2 separated files named stat.m and avg.m, where the first function calls the second.

```
function [m,s] = stat(x)
n = length(x);
m = avg(x,n);
s = sqrt(sum((x-m).^2/n));
end
```

```
function m = avg(x,n)
m = sum(x)/n;
end
```

- Call the function stat2.m, from the command line.

```
values = [12.7, 45.4, 98.9, 26.6, 53.1];
[ave,stdev] = stat(values)
ave = 47.3400
stdev = 29.4124
```

http://www.mathworks.com/help/matlab/ref/function.html

T.C. Adnan Menderes University Faculty of Engineering

Exercises

1. Manning's equation can be used to compute the velocity of water in a rectangular open channel:

$$U = \frac{\sqrt{S}}{n}\left(\frac{BH}{B+2H}\right)^{2/3}$$

Where U = velocity (m/s), S = channel slope, n = roughness coefficient, B = width (m), and H = depth (m).
Write a function that computes and outputs the velocity U, for any input set of S,n,B,H values.

2. Write an M-file for a program that calculates the standard deviation of any set of *N* numbers. Note: you should not use the built-in function std.

$$S = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}|A_i - \mu|^2} \quad \text{,where } \mu = \frac{1}{N}\sum_{i=1}^{N}A_i$$

T.C. Adnan Menderes University Faculty of Engineering

Copyright © 2011. The McGraw-Hill Companies, Inc.

References

- http://www.mathworks.com/help/matlab/getting-started-with-matlab.html

T.C. Adnan Menderes University Faculty of Engineering