

Assignment -1
***Executive Post Graduate Certification in Electric
Vehicle Design***

Submitted by:

Bolli Sahithya

Problem Statement:

Understanding basic MATLAB operations and matrix manipulations is crucial for performing complex computations in engineering and scientific applications. This assignment will help you become familiar with MATLAB's interface and basic functionalities.

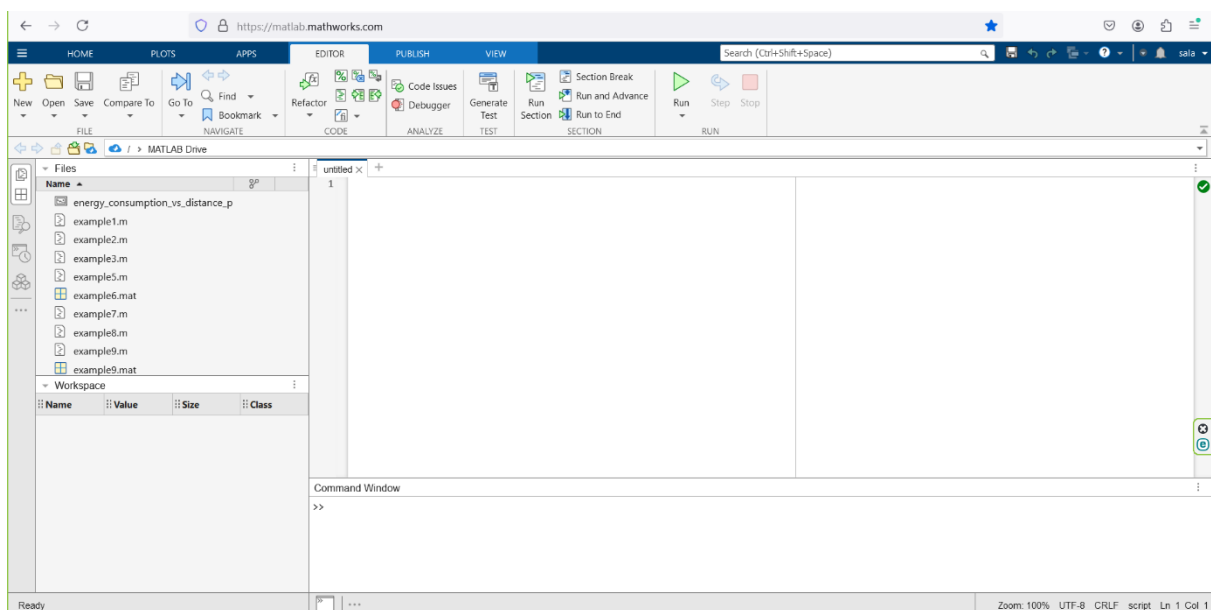
Objective:

- To familiarize students with MATLAB's environment and basic commands.
- To perform basic arithmetic operations and understand matrix manipulations.

Task to be performed:

1. Getting Started with MATLAB:

- Open MATLAB and explore the interface.



Command Window: We can enter commands and run them.

Workspace: Displays variables that are currently in the workspace.

Current Folder: Shows the files and folders in current directory.

Editor: Where you can write and run scripts.

- Use the help command to learn about basic functions

To learn about a basic MATLAB function, we can use the help command in the Command Window.

For example: `help plot`

This will show you how to use the plot function.

2. Basic Arithmetic Operations:

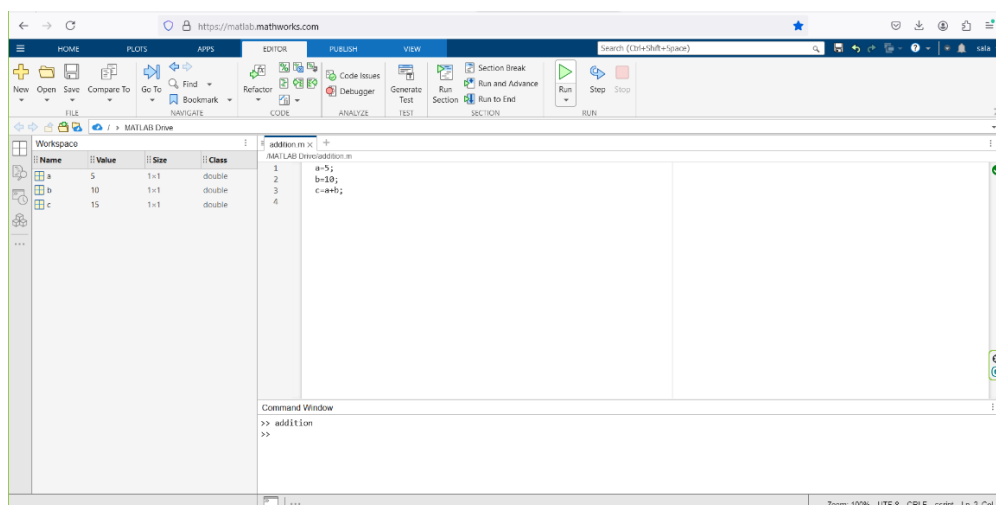
- Perform addition, subtraction, multiplication, and division using MATLAB commands.

Matlab program for addition:

```
a=5;  
b=10;  
c=a+b;
```

output:

```
c=15
```

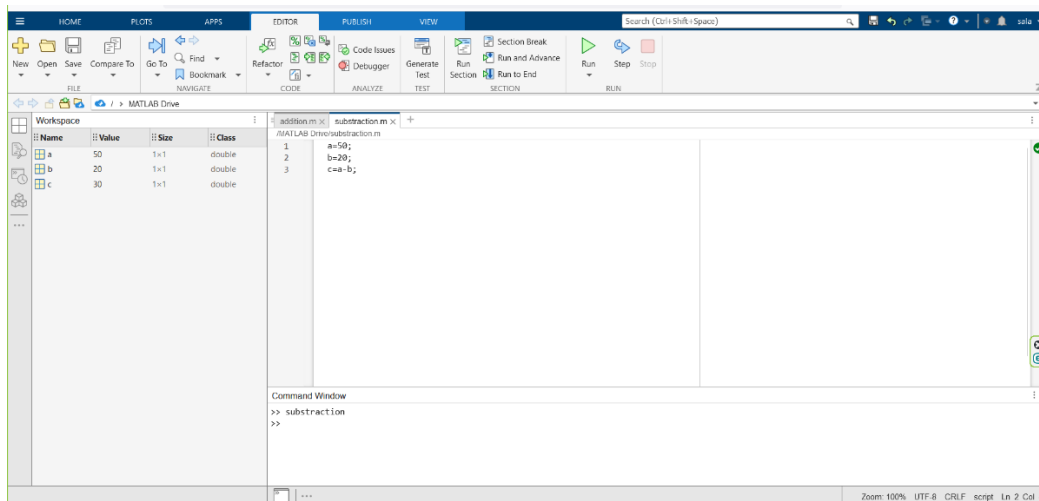


Matlab program for subtraction:

```
a=50;  
b=20;  
c=a-b;
```

output:

```
c=30
```

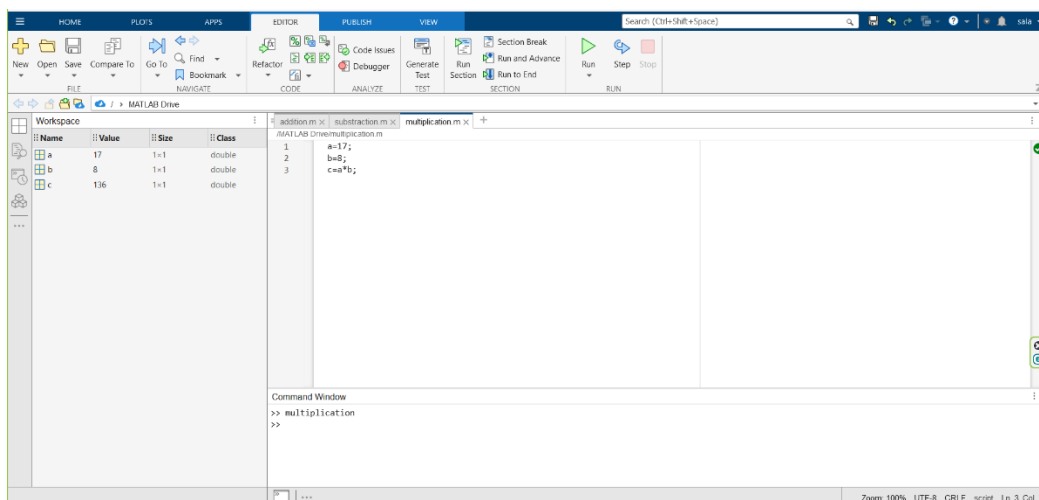


Matlab program for Multiplication:

```
a=17;  
b=8;  
c=a*b;
```

output:

```
c=136
```



Matlab program for Division:

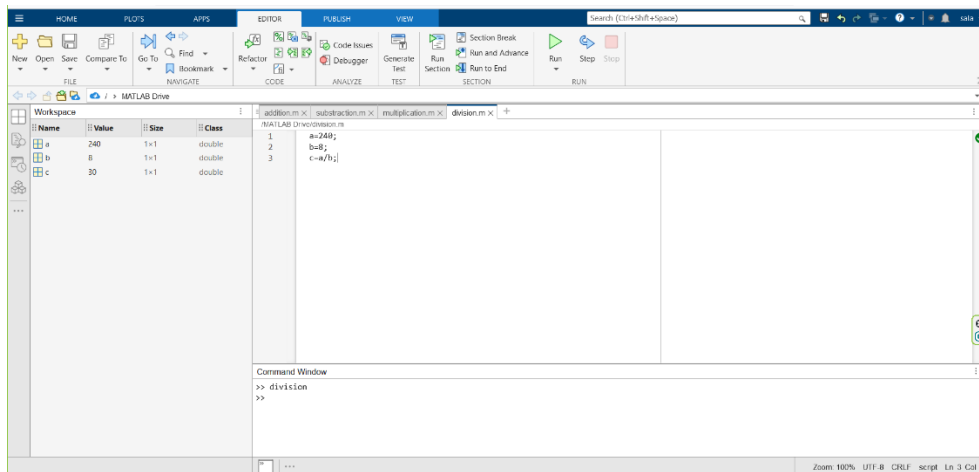
a=240;

b=8;

c=a/b;

output:

c=30



● Calculate powers and roots of numbers

Matlab program for powers:

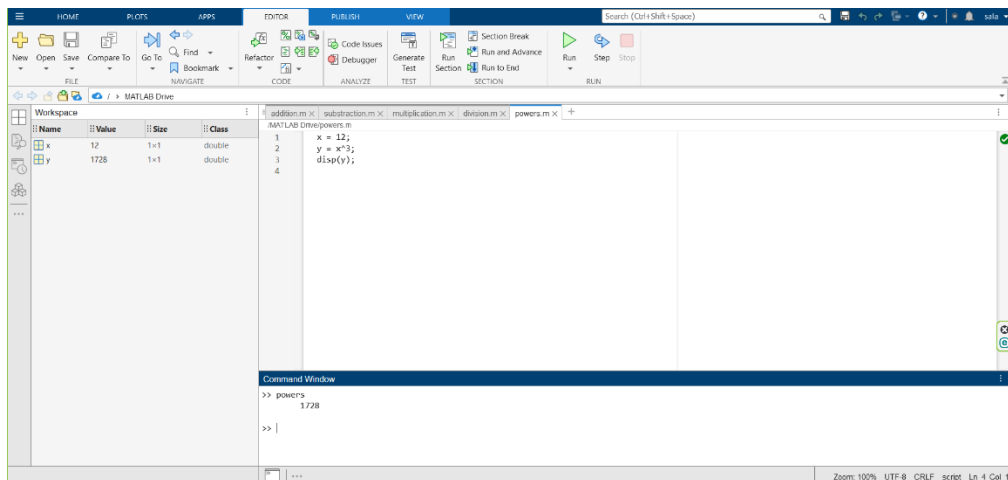
x = 12;

y = x^3;

disp(y);

output:

y=1728

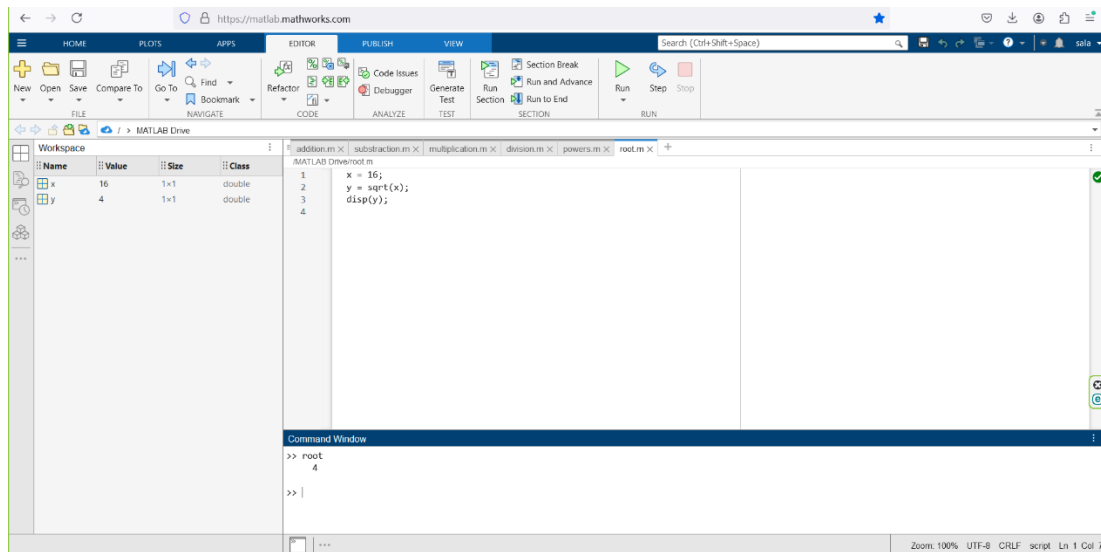


Matlab program for Roots:

```
x = 16;  
y = sqrt(x);  
disp(y);
```

output:

y=4



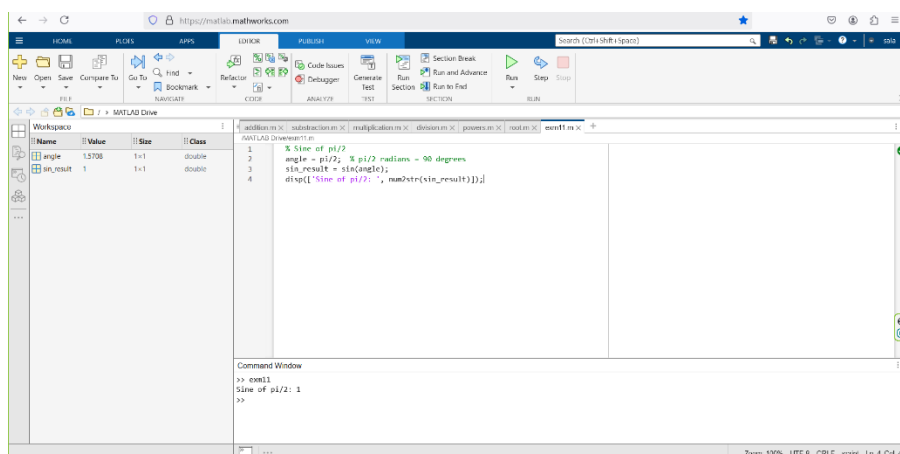
- Use built-in mathematical functions such as sin, cos, exp, and log.

Matlab program for Sin:

```
% Sine of pi/2  
angle = pi/2; % pi/2 radians = 90 degrees  
sin_result = sin(angle);  
disp(['Sine of pi/2: ', num2str(sin_result)]);
```

output:

sin of pi/2: 1



Matlab program for Cosine:

% Cosine of pi

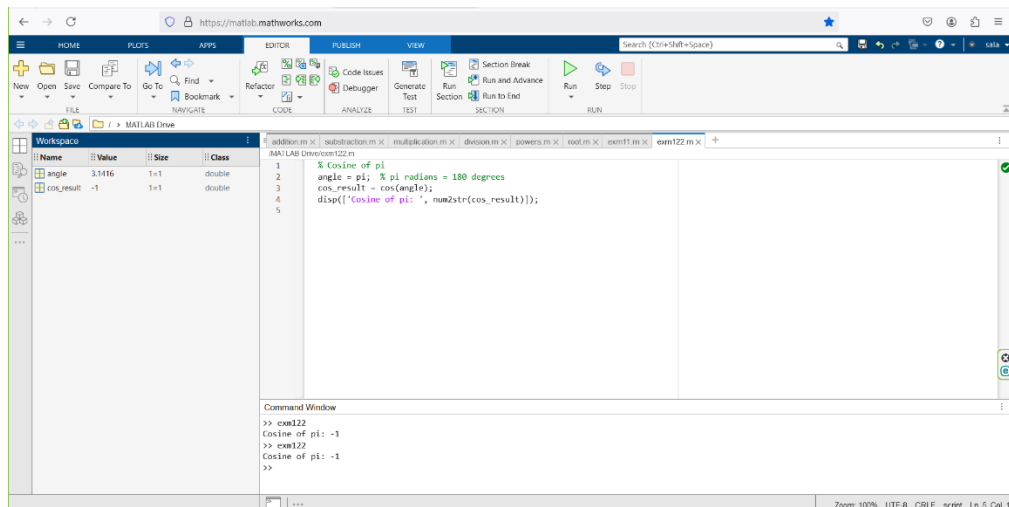
angle = pi; % pi radians = 180 degrees

cos_result = cos(angle);

disp(['Cosine of pi: ', num2str(cos_result)]);

output:

cosine of pi/2: -1



Matlab program for Exp:

% Exponential of 1

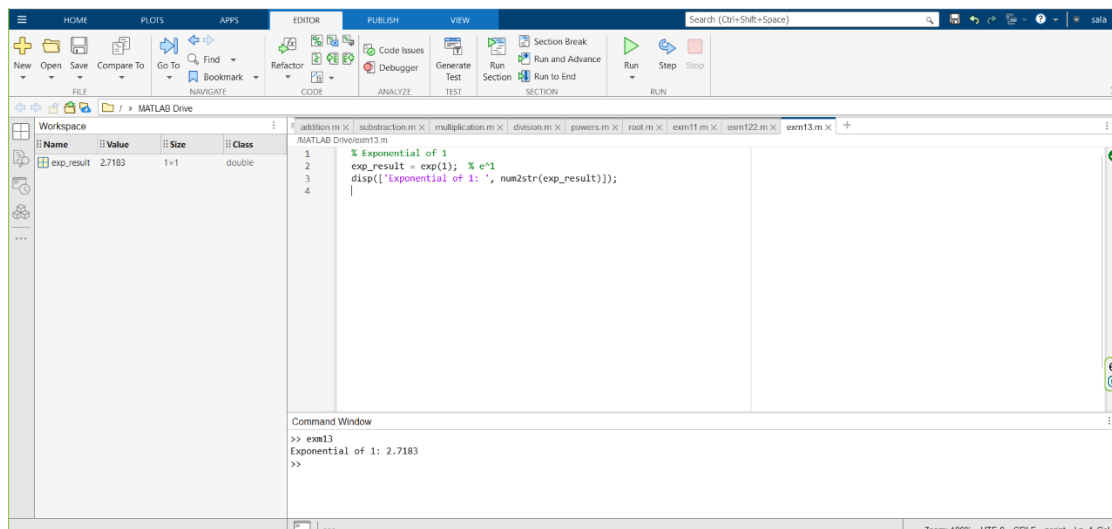
Exp_result = exp(1); % e^1

cos_result = cos(angle);

disp(['Exponential of 1: ', num2str(exp_result)]);

output:

Exponential of 1: 2.7183



3. Matrix Creation and Manipulation:

- Create different types of matrices (row, column, square, zero, identity).

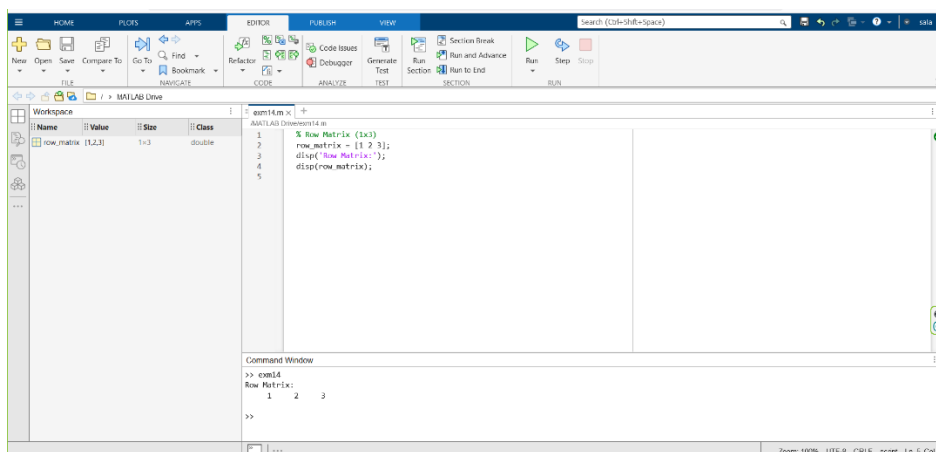
Matlab program for Row Matrix:

```
% Row Matrix (1x3)
row_matrix = [1 2 3];
disp('Row Matrix:');
disp(row_matrix);
```

output:

Row matrix:

1 2 3



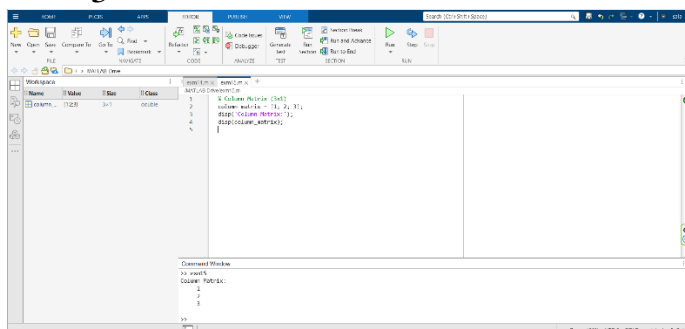
Matlab program for Column Matrix:

```
% Column Matrix (3x1)
column_matrix = [1; 2; 3];
disp('Column Matrix:');
disp(column_matrix);
```

output:

Column matrix:

1
2
3



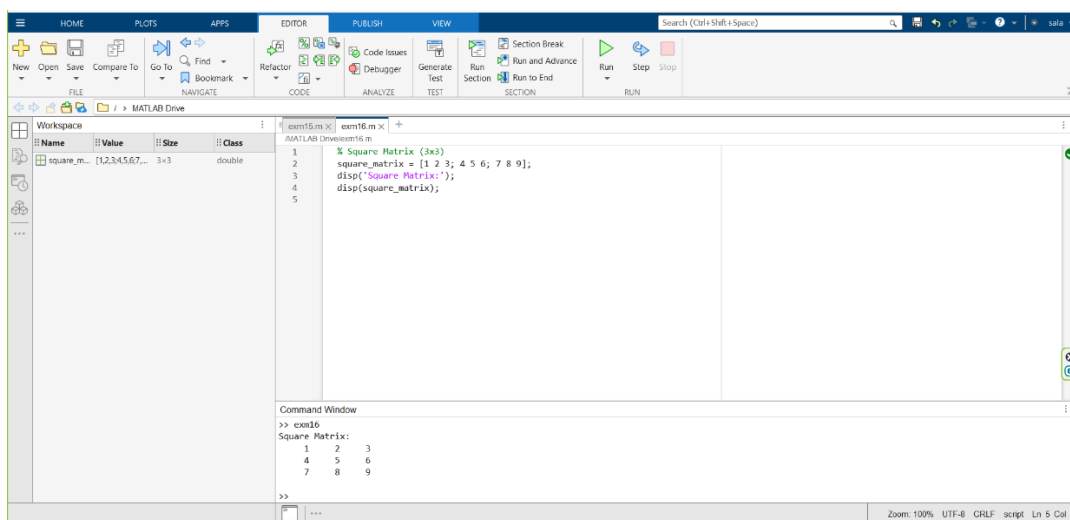
Matlab program for Square Matrix:

```
% Square Matrix (3x3)
square_matrix = [1 2 3; 4 5 6; 7 8 9];
disp('Square Matrix:');
disp(square_matrix);
```

output:

Square matrix:

```
1 2 3
4 5 6
7 8 9
```



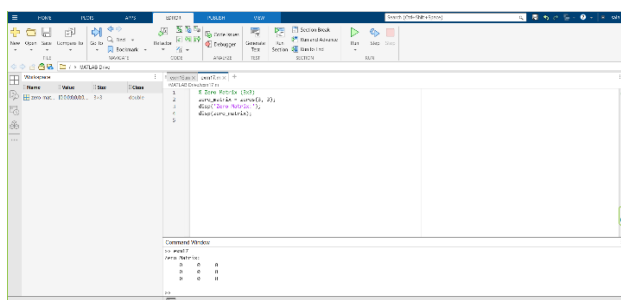
Matlab program for Zero Matrix:

```
% Zero Matrix (3x3)
zero_matrix = zeros(3, 3);
disp('Zero Matrix:');
disp(zero_matrix);
```

output:

Zero matrix:

```
0 0 0
0 0 0
0 0 0
```



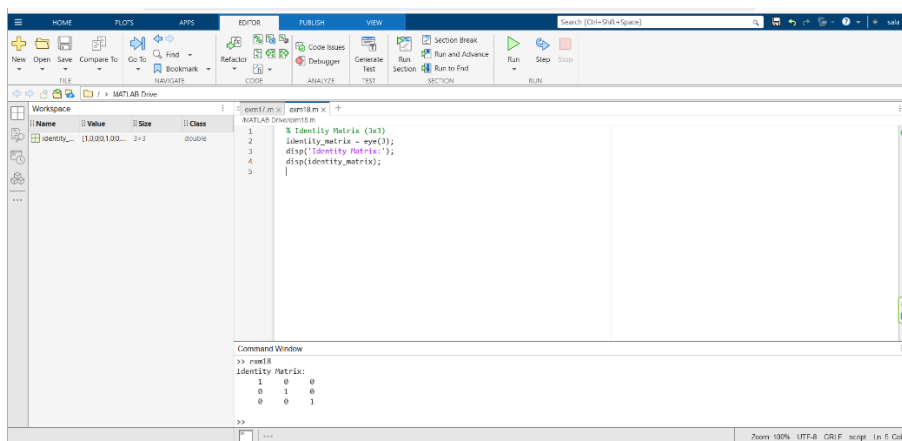
Matlab program for Identity Matrix:

```
% Identity Matrix (3x3)
identity_matrix = eye(3);
disp('Identity Matrix:');
disp(identity_matrix);
```

output:

Identity matrix:

```
1  0  0
0  1  0
0  0  1
```



- Perform matrix addition, subtraction, and multiplication.

Matlab program for Matrix Addition:

% Two matrices for addition

```
A = [1 2; 3 4];
```

```
B = [5 6; 7 8];
```

% Adding matrices

```
sum_matrix = A + B;
```

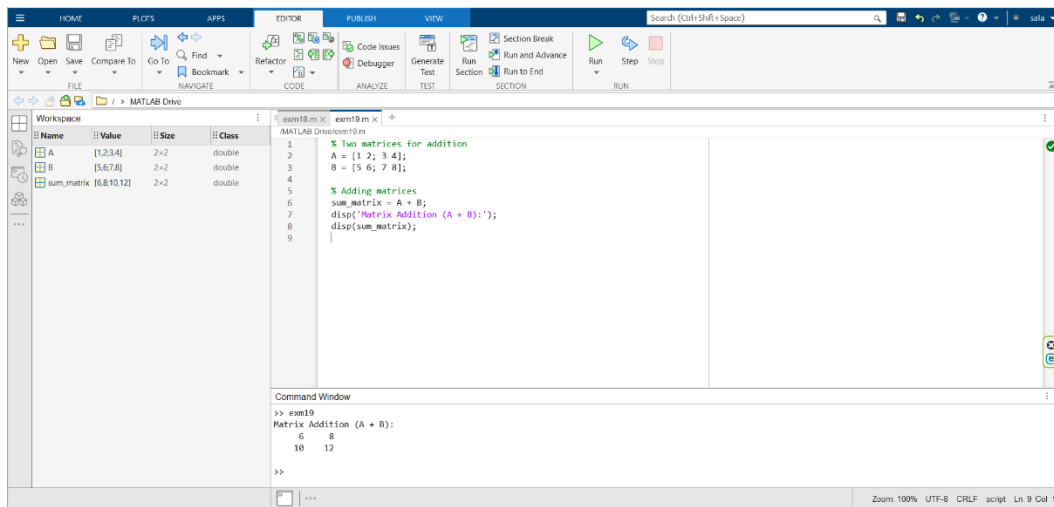
```
disp('Matrix Addition (A + B):');
```

```
disp(sum_matrix);
```

output:

Matrix Addition (A + B):

```
6  8
10 12
```



Matlab program for Matrix Subtraction:

% Two matrices for subtracting

A = [1 2; 3 4];

B = [5 6; 7 8];

% Subtracting matrices

difference_matrix = A-B;

disp ('Matrix Subtraction (A-B):');

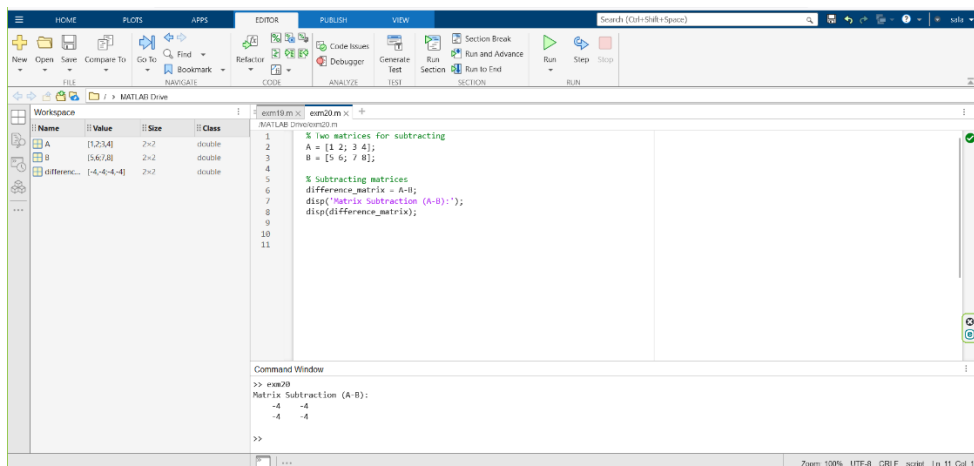
disp(difference_matrix);

output:

Matrix Subtraction (A-B):

-4 -4

-4 -4



Matlab program for Matrix Multiplication:

% Two matrices for multiplication

```
A = [1 2; 3 4];
```

```
B = [5 6; 7 8];
```

% Matrix multiplication

```
product_matrix = A * B;
```

```
disp('Matrix Multiplication (A * B):');
```

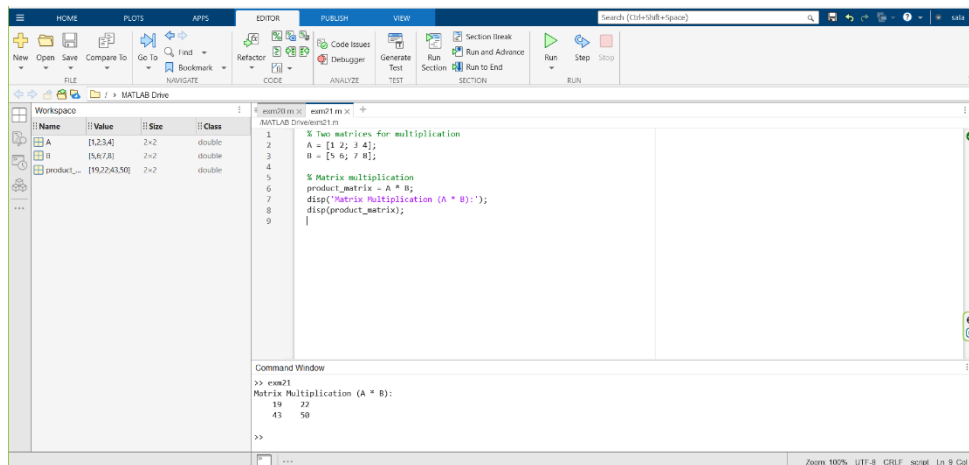
```
disp(product_matrix);
```

output:

Matrix Multiplication (A * B):

```
19    22
```

```
43    50
```



- Transpose a matrix.

Matlab program for Transpose a Matrix:

% matrices for Transpose

```
A = [1 2; 3 4];
```

% Transpose of matrix A

```
A_transpose = A';
```

```
Disp('Transpose of A:');
```

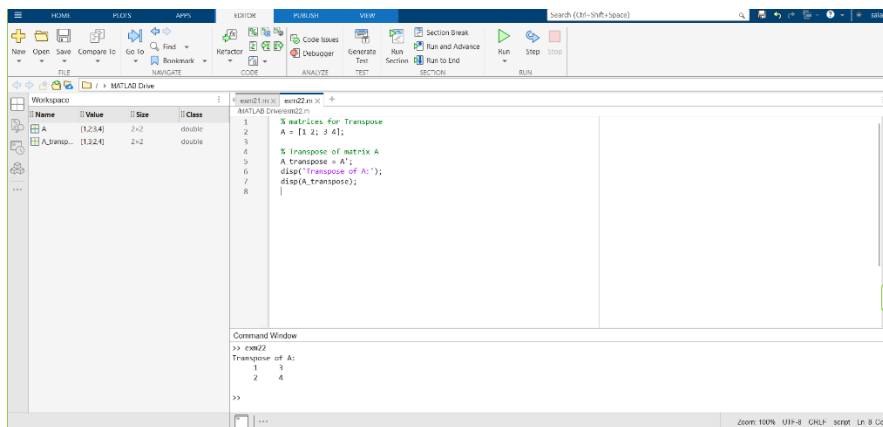
```
disp(A_transpose);
```

output:

Transpose of A:

```
1    3
```

```
2    4
```



● Extract rows, columns, and submatrices

Matlab program for Extract Rows:

% Matrix

A = [1 2; 3 4];

B = [5 6; 7 8];

% Extracting the first row from matrix A

row1 = A (1, :);

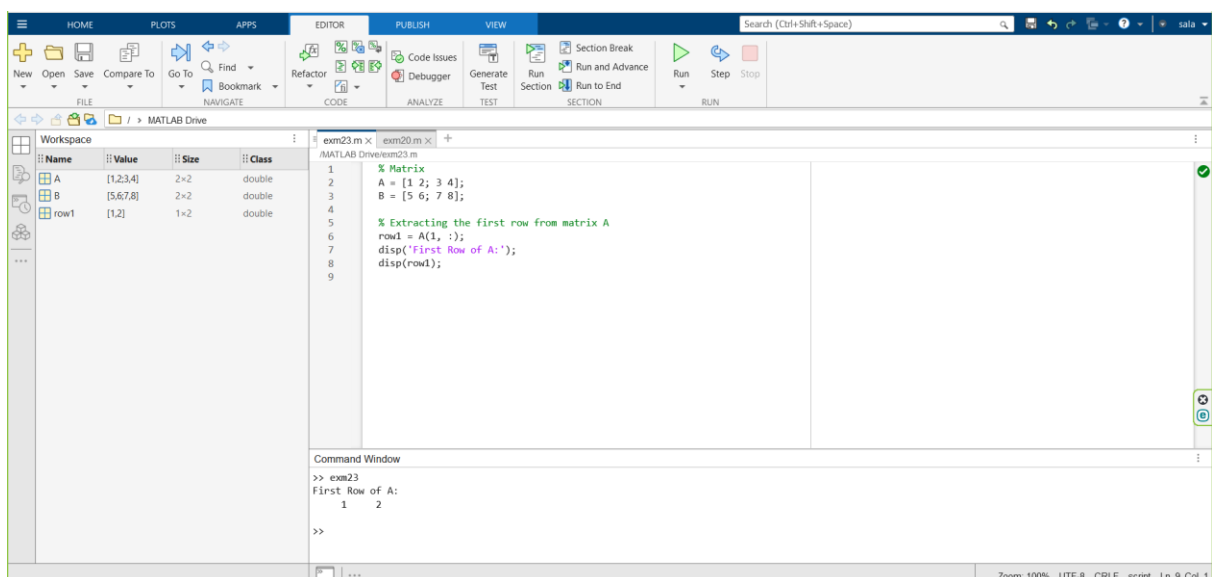
disp ('First Row of A:');

disp(row1);

output:

First Row of A:

1 2



Matlab program for Extract Columns:

% Matrix

```
A = [1 2; 3 4];
```

```
B = [5 6; 7 8];
```

% Extracting the first column from matrix A

```
col1 = A(:, 1);
```

```
disp('First Column of A:');
```

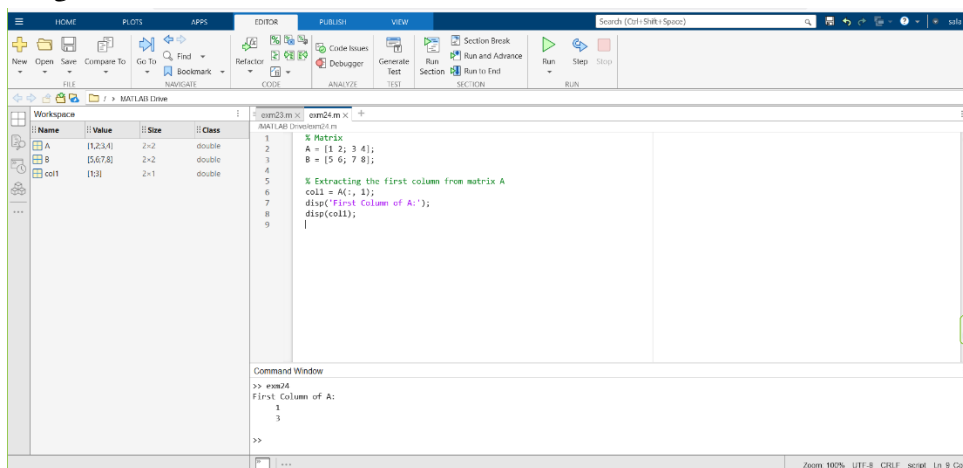
```
disp(col1);
```

output:

First Column of A:

1

3



Matlab program for Extract Submatrices:

% Matrix

```
A = [1 2; 3 4];
```

```
B = [5 6; 7 8];
```

% Extracting a submatrix from A (rows 1 to 2 and columns 1 to 2)

```
submatrix = A (1:2, 1:2);
```

```
disp('Submatrix (1:2, 1:2) of A:');
```

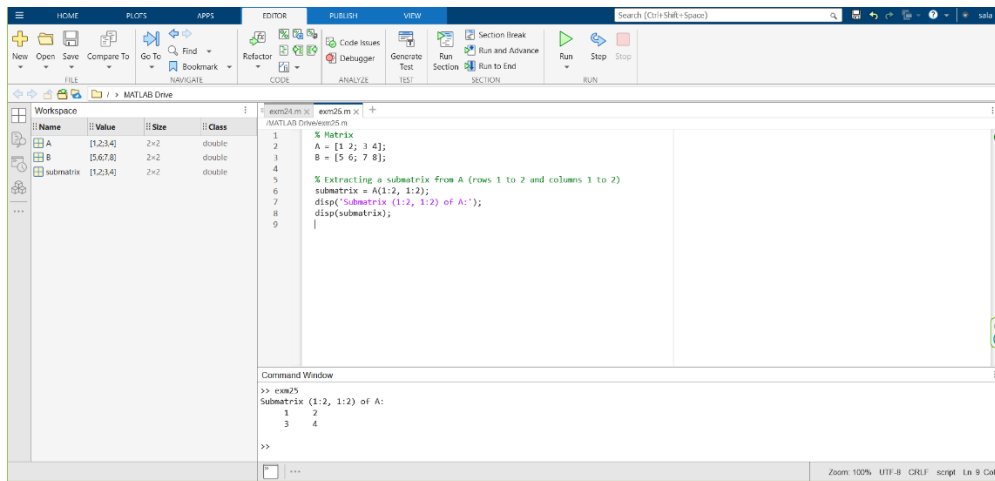
```
disp(submatrix);
```

output:

Submatrix (1:2, 1:2) of A:

1 2

3 4



4. Element-wise Operations:

- Perform element-wise multiplication and division.
- Use the `.*`, `./`, and `.^` operators.

Matlab program for Element-wise Multiplication (`.*`):

```

A = [1 2 3; 4 5 6];
B = [7 8 9; 10 11 12];
C = A.*B; % Element-wise multiplication
disp(C);

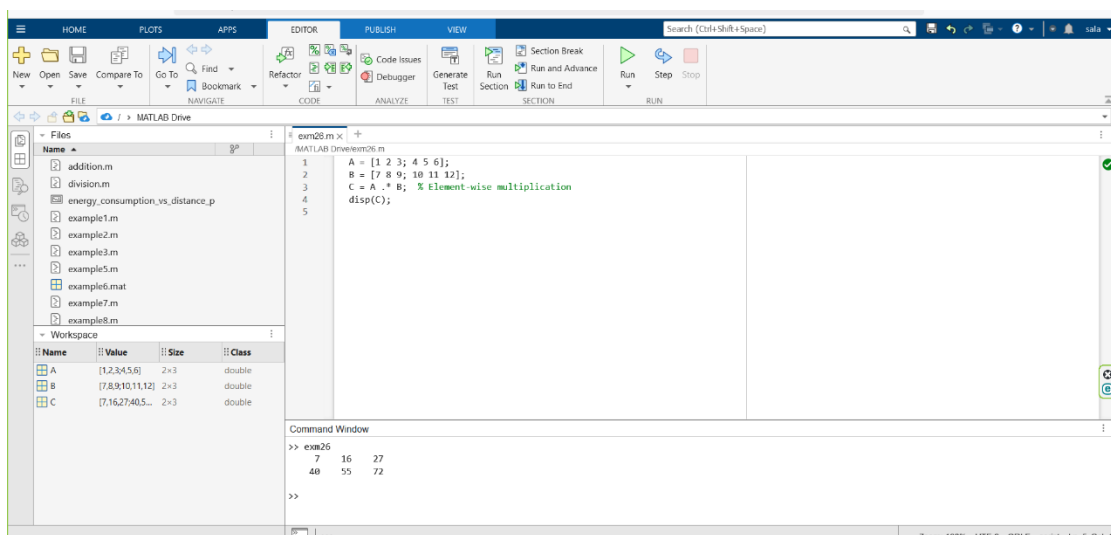
```

Output:

```

7    16    27
40    55    72

```

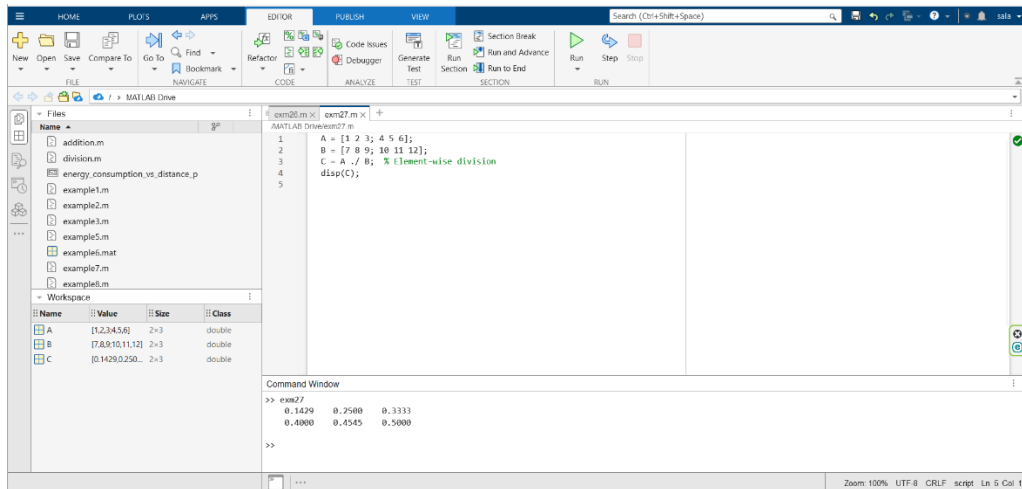


Matlab program for Element-wise Division (./):

```
A = [1 2 3; 4 5 6];  
B = [7 8 9; 10 11 12];  
C = A ./ B; % Element-wise division  
disp(C);
```

Output:

```
0.1429    0.2500    0.3333  
0.4000    0.4545    0.5000
```

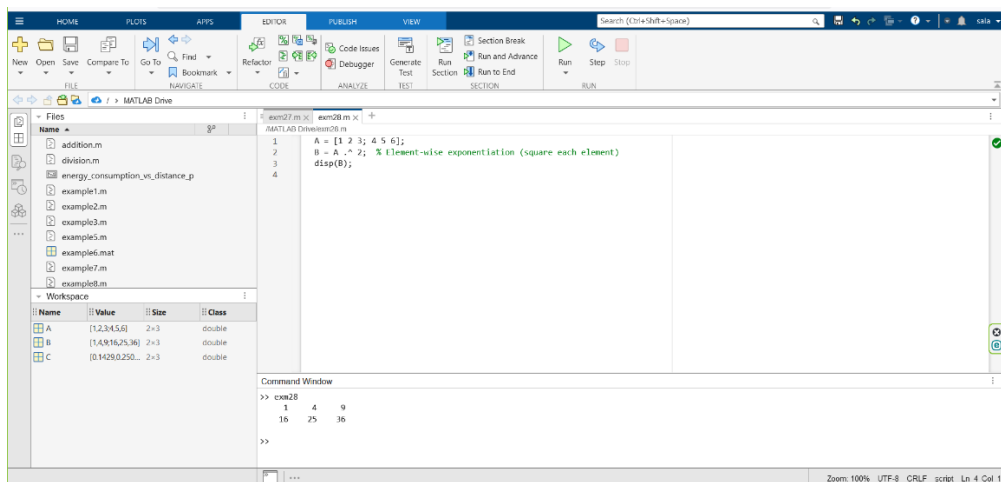


Matlab program for Element-wise Exponentiation (.^):

```
A = [1 2 3; 4 5 6];  
B = A .^ 2; % Element-wise exponentiation (square each element)  
disp(B);
```

Output:

```
1     4     9  
16    25    36
```



5.Solving Linear Equations:

- Solve a system of linear equations using matrix operations.
- Use the inv function to find the inverse of a matrix.
- Verify the solution using the linsolve function

Matlab program for Matrix Inversion (inv):

```
% Coefficient matrix A and right-hand side vector B
```

```
A = [3 2; 4 1];
```

```
B = [5; 6];
```

```
% Find the inverse of A
```

```
A_inv = inv(A);
```

```
% Solve for X using  $X = A\_inv * B$ 
```

```
X = A_inv * B;
```

```
% Display the solution
```

```
Disp ('Solution using inverse:');
```

```
disp(X);
```

Matlab program for linsolve function:

```
% Solve the system using linsolve function
```

```
X_linsolve = linsolve (A, B);
```

```
% Display the solution
```

```
Disp ('Solution using linsolve:');
```

```
disp(X_linsolve);
```

Output:

Solution using inverse:

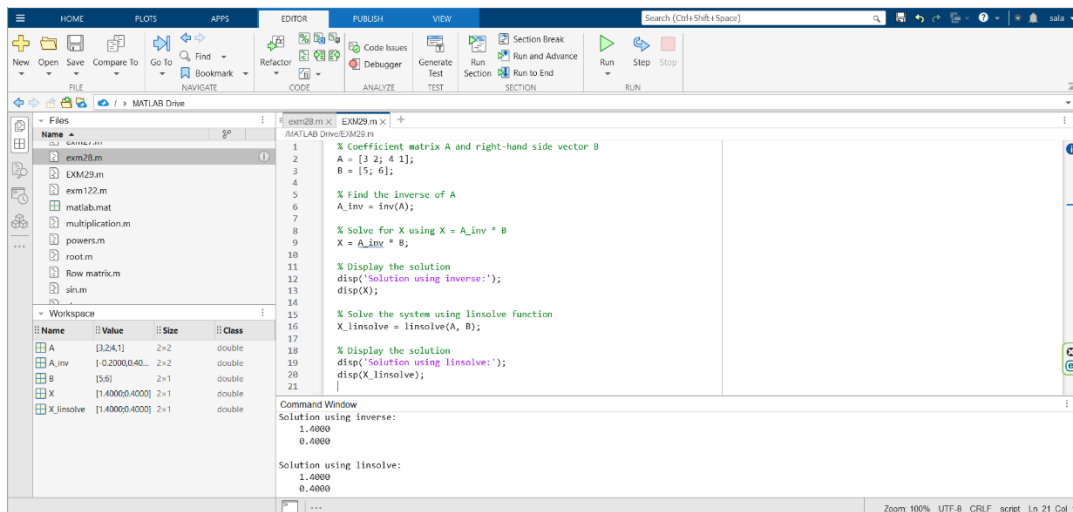
1.4000

0.4000

Solution using linsolve:

1.4000

0.4000



6. Plotting Basic Graphs:

1. Create simple plots using the plot function.
2. Label the axes and title the graph.
3. Customize the appearance of the plot (line style, color, markers).

Matlab program for Plotting Basic Graphs:

% Define x values from 0 to 2π

`x = linspace(0, 2*pi, 100);`

% Define y values as sine of x

`y = sin(x);`

% Create the plot

`figure; % Open a new figure window`

`plot(x, y, '--ro', 'MarkerFaceColor', 'r'); % Dashed red line with red circle markers`

% Label the axes

`Xlabel('x values'); % Label for the x-axis`

`ylabel('sin(x)'); % Label for the y-axis`

% Add a title

`Title('Plot of y = sin(x)');`

