# ***Task\_01***

% Create matrix a

a = [1 4 3; 4 2 6; 7 8 9];

% Determinant of a (singular matrix)

det\_a = det(a)

% Inverse of a (not possible for singular matrix)

inv\_a = inv(a) % This will result in an error message

% Transpose of a

a\_transpose = a'

% Minimum element from each column

min\_col = min(a)

% Smallest element in a

smallest = min(min(a))

% Maximum element from each column

max\_col = max(a)

% Largest element in a

largest = max(max(a))

% Square of each element

a\_squared = a.^2

% Sum of each column

sum\_col = sum(a)

% Sum of all elements

total\_sum = sum(sum(a))

% Number of rows and columns

[rows, cols] = size(a) % Or use size(a, 1) for rows, size(a, 2) for columns

% New matrix a for length example

b = [4 5 6];

% Number of elements in row vector

num\_elements = length(b)

% Sub-matrix extraction

sub\_matrix = a(1:2, 1:2) % Access rows 1 and 2, columns 1 and 2

%Range of the element selection

range\_selection = a(1:2)

% Row extraction

row\_vector = a(3) % Access elements from row 3

# ***Task\_02***

% 1. Generate a 6x6 matrix A with random values

A = rand(6, 6);

% 2. Generate a 6x1 matrix z with random values

z = rand(6, 1);

% 3. Solve the linear system of equations Ax = z

try

x = A \ z; % Use backslash for solving linear systems

fprintf("Solution to Ax = z:\n");

disp(x);

catch ME

if strcmp(ME.identifier, 'MATLAB:linsolve:NoSolution')

fprintf("The system may not have a unique solution due to a singular or ill-conditioned matrix A.\n");

else

rethrow(ME); % Rethrow other errors

end

end

% 4. Compute the determinant of matrix A

det\_A = det(A);

fprintf("\nDeterminant of A: %f\n", det\_A);

% 5. Extract a 4x4 sub-matrix from A

A\_sub = A(1:4, 1:4);

fprintf("\nExtracted 4x4 sub-matrix:\n");

disp(A\_sub);

% Extract 4x1 sub-matrices from z

z\_sub1 = z(1:4);

z\_sub2 = z(2:5); % Corrected: 2nd sub-matrix starts at row 2 (inclusive)

z\_sub3 = z(4:end);

fprintf("\nExtracted 4x1 sub-matrices from z:\n");

disp(z\_sub1);

disp(z\_sub2);

disp(z\_sub3);

# ***Task\_03***

% Question 1.3

% 1) Calculating results

numerator1 = 35.7 \* (64 - 7^4);

denominator1 = 45 + 5^3;

result1 = numerator1 / denominator1;

% 2) Calculating results

numerator2 = 3^7 \* log(76);

denominator2 = 7^3 + 564;

result2 = numerator2 / denominator2;

% 3) Calculating results

angle = 5\*pi/6;

result3 = cos(angle)^2 \* sin(7\*pi/8)^2 + tan(pi/6 \* log(8)) / sqrt(7);

% 4) Creating matrix B with equal spacing in all rows

B = [linspace(1, 25, 9);

linspace(72, 24, 9);

linspace(0, 1, 9)]; % Using linspace for constant spacing

% Display results

disp('Question 1.3:')

fprintf('1) %.4f\n', result1);

fprintf('2) %.4f\n', result2);

fprintf('3) %.4f\n', result3);

disp('4) ');

disp(B);

# Question 1.4

## **1) Write a MATLAB code that generates Delta (Impulse) Function.**

* **Use the plotting function stem to make the graphs**
* **Replace the stem command in the above code with the plot command and run the code again. How does this change the plot? And why?**

t = -5:0.01:5; % Time vector

impulse = t == 0; % Impulse function (non-zero only at t=0)

% Plot with stem

figure;

stem(t, impulse);

title('Delta (Impulse) Function - stem');

% Plot with plot

figure;

plot(t, impulse);

title('Delta (Impulse) Function - plot');

## **2) Write a MATLAB code that generate Unit Step function:**

* **Use the plotting function stem to make the graphs**
* **Replace the stem command in the above code with the plot command and run the code again. How does this change the plot?**

t = -5:0.01:5;

step = t >= 0; % Unit step function

% Plot with stem

figure;

stem(t, step);

title('Unit Step Function - stem');

% Plot with plot

figure;

plot(t, step);

title('Unit Step Function - plot');

## **3) Write a MATLAB code that generate Unit Ramp function**

* **Use the plotting function plot to make the graphs**
* **Replace the plot command in the above code with the stem command and run the code again. How does this change the plot?**

t = -5:0.01:5;

ramp = t .\* (t >= 0); % Unit ramp function

% Plot with plot

figure;

plot(t, ramp);

title('Unit Ramp Function - plot');

% Plot with stem

figure;

stem(t, ramp);

title('Unit Ramp Function - stem');