**ITW Experiment 5**

**Anshal Kapil, 7**

**Aim:** To Implement basic operations in MATLAB along with matrix operations.

**Theory:**

Matrices are foundational mathematical constructs with wide-ranging applications in various disciplines, notably in linear algebra, physics, computer science, and engineering. A matrix is a two-dimensional array organized in rows and columns, usually denoted with a capital letter like A, with elements represented as a\_ij, where "i" denotes the row and "j" the column index. The dimensions of a matrix are expressed as "m x n," indicating the number of rows (m) and columns (n), such as a 3x2 matrix with 3 rows and 2 columns.

Matrix operations include addition (for matrices of the same dimensions), scalar multiplication (scaling all elements by a single number), and matrix multiplication (where the number of columns in the first matrix must match the number of rows in the second). Matrices can also be transposed by switching rows and columns, while determinants provide insight into invertibility and linear independence. Eigenvalues and eigenvectors help analyze matrix transformations, and matrix decompositions simplify complex matrices into more manageable forms, like LU decomposition or singular value decomposition. Ultimately, matrices serve as a fundamental tool for solving equations, representing data, and solving a wide array of mathematical and real-world problems.

**Code:**

disp('Simple Calculator');

disp('1. Addition');

disp('2. Subtraction');

disp('3. Multiplication');

disp('4. Division');

choice = input('Enter your choice (1/2/3/4): ');

num1 = input('Enter the first number: ');

num2 = input('Enter the second number: ');

switch choice

case 1

result = num1 + num2;

fprintf('Result: %f\n', result);

case 2

result = num1 - num2;

fprintf('Result: %f\n', result);

case 3

result = num1 \* num2;

fprintf('Result: %f\n', result);

case 4

if num2 == 0

fprintf('Error: Division by zero is not allowed.\n');

else

result = num1 / num2;

fprintf('Result: %f\n', result);

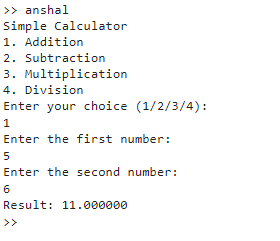
end

otherwise

fprintf('Invalid choice. Please enter a valid choice (1/2/3/4).\n');

end

**Output:**



**Code:**

input\_matrix = [1, 2, 3; 4, 5, 6; 7, 8, 9];

transpose\_matrix = transpose(input\_matrix);

if size(input\_matrix, 1) == size(input\_matrix, 2)

inverse\_matrix = inv(input\_matrix);

disp('Inverse Matrix:');

disp(inverse\_matrix);

else

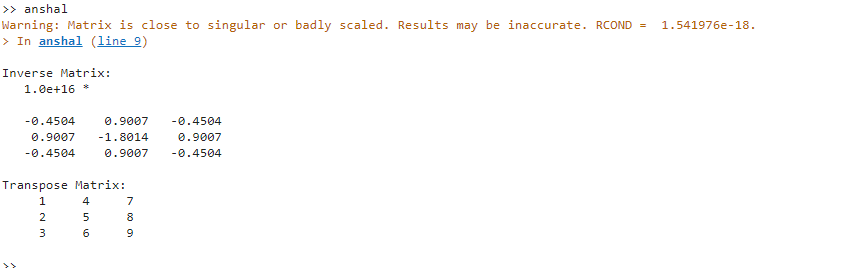
disp('Input matrix is not square, so the inverse cannot be calculated.');

end

disp('Transpose Matrix:');

disp(transpose\_matrix);

**Output:**



**Conclusion:** Thus we have implemented basic operations in MATLAB and some basic Matrix operations.