```
disp('AI In Speech Processing')
disp('Name: J Viswaksena')
```

```
Name: J Viswaksena
```

```
disp('RollNo: AM.EN.U4AIE21035')
```

```
RollNo: AM.EN.U4AIE21035
```

```
C =    [-1, 0, 0; 1,1,0; 1,-1,0; 0,0,2] ;
fprintf('Method-1')
C
```

```

```

```
C2 =    [-1, 0, 0
        1, 1, 0
        1, -1, 0
        0, 0,  2];
fprintf('Method-2')
C2
```

```
B = [3,5,6];
S = [7, 8, B];
F = [ 1, 2, 3, 4, 5,6, 7, 8, 9, 10];
F
```

```
F = 1×10
     1     2     3     4     5     6     7     8     9    10
```

```
F = [ 1, 2, 3, 4, 5,...
6, 7, 8, 9, 10];
F
```

```
F = 1×10
     1     2     3     4     5     6     7     8     9    10
```

```
fprintf('Method-3')
```

```
Method-3
```

```
S
```

```
S = 1×5
     7     8     3     5     6
```

```
fprintf('Matrix manipulations')
```

```
Matrix manipulations
```

```
S(2) = 2 %(change value of 2nd index)
```

```
S = 1×5
     7     2     3     5     6
```

```
S(3:4) = [6,3] %(change range of values)
```

```
S = 1×5
     7     2     6     3     6
```

```
S(6) = 9 %(add an element at end)
```

```
S = 1×6
     7     2     6     3     6     9
```

```
S(9) = 13 %(add a 9th element; Since index 7 and 8 don%t exist, they
automatically get initialized to 0).
```

```
S = 1×9
     7     2     6     3     6     9     0     0    13
```

```
I = [1,2,4] ;
S(I) = 42 %=> Change 1st, 2nd and 4th element to 42 ie: changing an
arbitrary subset of a matrix.
```

```
S = 1×9
    42    42     6    42     6     9     0     0    13
```

```
S(I+1) = 45 %=> change 2nd, 3rd and 5th elements to 45
```

```
S = 1×9
    42    45    45    42    45     9     0     0    13
```

```
% Creating empty matrices
A = [];
B = 4:-1:5;

% Displaying the empty matrices
disp('Empty Matrix A:');
```

```
Empty Matrix A:
```

```
A
```

```
A =

     []
```

```
disp('Empty Matrix B:');
```

```
Empty Matrix B:
```

```
B
```

```
B =

  1x0 empty double row vector
```

```
A = [1, 2, 3; 4, 5, 6; 7, 8, 9];

% Transpose of the matrix
A_transpose = A';

% Displaying the transpose
disp('Transpose of Matrix A:');
```

```
Transpose of Matrix A:
```

```
disp(A_transpose);
```

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

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

% Calculate the inverse
A_inv = inv(A);

% Display the inverse
disp('Inverse of Matrix A:');
```

```
Inverse of Matrix A:
```

```
disp(A_inv);
```

```
   -2.0000    1.0000
    1.5000   -0.5000
```

```
A = [1, 2; 3, 4];
B = [2, 0; 1, 3];

% Dot product
C = A .* B;

% Display the result
disp('Dot Product:');
```

```
Dot Product:
```

```
disp(C);
```

```
     2     0
     3    12
```

```
A = [1, 2; 3, 4];
B = [2, 3; 4, 5];

% Addition of matrices
C = A + B;

% Display the result
disp('Addition of Matrices A and B:');
```

```
Addition of Matrices A and B:
```

```
disp(C);
```

```
     3     5
     7     9
```

```
% Subtraction of matrices
C = A - B;

% Display the result
disp('Subtraction of Matrices A and B:');
```

```
Subtraction of Matrices A and B:
```

```
disp(C);
```

```
    -1    -1
    -1    -1
```

```
% Multiplication of matrices
C = A * B;

% Display the result
disp('Multiplication of Matrices A and B:');
```

```
Multiplication of Matrices A and B:
```

```
disp(C);
```

```
    10    13
    22    29
```

```
% Division of matrix by scalar
C = A / B;

% Display the result
disp('Division of Matrix:');
```

```
Division of Matrix:
```

```
disp(C);
```

```
    1.5000   -0.5000
    0.5000    0.5000
```

```
% Exponentiation of matrix
B = A ^ 2;

% Display the result
disp('Exponentiation of Matrix A to the Power of 2:');
```

```
Exponentiation of Matrix A to the Power of 2:
```

```
disp(B);
```

```
     7    10
    15    22
```

```
A = [1, 2; 3, 4];
B = [2, 0; 1, 3];

% Matrix multiplication
C = A * B;

% Display the result
disp('Matrix Multiplication:');
```

```
Matrix Multiplication:
```

```
disp(C);
```

```
     4     6
    10    12
```

```
scalar = 2;
matrix = [1, 2; 3, 4];

% Scalar multiplication using array operation
result_array = scalar * matrix;

% Scalar multiplication using matrix operation
result_matrix = scalar .* matrix;

% Display results
disp('Result using Array Operation:');
```

```
Result using Array Operation:
```

```
disp(result_array);
```

```
     2     4
     6     8
```

```
disp('Result using Matrix Operation:');
```

```
Result using Matrix Operation:
```

```
disp(result_matrix);
```

```
     2     4
     6     8
```

```
A = [2, 4; 6, 8];

% Element-wise division
B = A ./ 2;

% Element-wise exponentiation
C = A .^ 2;

% Display the results
disp('Element-wise Division (A ./ 2):');
```

```
Element-wise Division (A ./ 2):
```

```
disp(B);
```

```
     1     2
     3     4
```

```
disp('Element-wise Exponentiation (A .^ 2):');
```

```
Element-wise Exponentiation (A .^ 2):
```

```
disp(C);
```

```
     4    16
    36    64
```

```
disp('Matrix of Zeros')
zeros (6)    %: 6 by 6 matrix of zeros
```

```
ans = 6×6
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
```

```
zeros (3,2) %: 3 by 2 matrix of zeros
```

```
ans = 3×2
     0     0
     0     0
     0     0
```

```
disp('Matrix of Ones:')
```

```
Matrix of Ones:
```

```
ones (6)    %: 6 by 6 matrix of ones
```

```
ans = 6×6
     1     1     1     1     1     1
     1     1     1     1     1     1
     1     1     1     1     1     1
     1     1     1     1     1     1
     1     1     1     1     1     1
     1     1     1     1     1     1
```

```
ones (3,2)  %: 3 by 2 matrix of ones
```

```
ans = 3×2
     1     1
     1     1
     1     1
```

```
disp('Identity matrix')
```

```
Identity matrix
```

```
eye(6)   %produces a 6x6 identity matrix
```

```
ans = 6×6
     1     0     0     0     0     0
     0     1     0     0     0     0
     0     0     1     0     0     0
     0     0     0     1     0     0
     0     0     0     0     1     0
     0     0     0     0     0     1
```

```
eye(3,2) %produces a 3x2 identity matrix
```

```
ans = 3×2
     1     0
     0     1
     0     0
```

```
disp('Diagonal matrix:')
```

```
Diagonal matrix:
```

```
C = [1, 2, 3 ; 4,5,6; 7,8,9]
```

```
C = 3×3
     1     2     3
     4     5     6
     7     8     9
```

```
diag (C)    %gives ans= [1,5,9]
```

```
ans = 3×1
```

```
     1
     5
     9
```

```
disp('Other diagonal elements')
```

```
Other diagonal elements
```

```
diag (C,1) %gives ans=[2,6]
```

```
ans = 2×1
     2
     6
```

```
disp('Diagonal matrices can also be created if an input vector (one row
vector is provided)')
```

```
Diagonal matrices can also be created if an input vector (one row vector is provided)
```

```
V = [1, 3, 4, 5]
```

```
V = 1×4
     1     3     4     5
```

```
diag(V,0) %: Creates a 4 x 4 matrix with diagonal elements as 1, 3, 4, 5
```

```
ans = 4×4
     1     0     0     0
     0     3     0     0
     0     0     4     0
     0     0     0     5
```

```
disp('Compute the square root of a scalar and a vector')
```

```
Compute the square root of a scalar and a vector
```

```
x_scalar = 4;
x_vector = [9, 16, 25];
sqrt_scalar = sqrt(x_scalar);
sqrt_vector = sqrt(x_vector);
disp('Square root of scalar:');
```

```
Square root of scalar:
```

```
disp(sqrt_scalar);
```

```
     2
```

```
disp('Square root of vector:');
```

```
Square root of vector:
```

```
disp(sqrt_vector);
```

```
     3     4     5
```

```
x_matrix = [1, 2, 3; 4, 5, 6];
[s_rows, s_cols] = size(x_matrix);
size(x_matrix)
```

```
ans = 1×2
     2     3
```

```
disp('Number of rows:');
```

```
Number of rows:
```

```
disp(s_rows);
```

```
     2
```

```
disp('Number of columns:');
```

```
Number of columns:
```

```
disp(s_cols);
```

```
     3
```

```
disp('Compute the absolute value of a scalar and a vector')
```

```
Compute the absolute value of a scalar and a vector
```

```
x_scalar = -5;
x_vector = [-2, 0, 4];
abs_scalar = abs(x_scalar);
abs_vector = abs(x_vector);
disp('Absolute value of scalar:');
```

```
Absolute value of scalar:
```

```
disp(abs_scalar);
```

```
     5
```

```
disp('Absolute value of vector:');
```

```
Absolute value of vector:
```

```
disp(abs_vector);
```

```
     2     0     4
```

```
disp('Compute the exponential of a scalar and a vector')
```

```
Compute the exponential of a scalar and a vector
```

```
x_scalar = 2;
```

```
x_vector = [1, 2, 3];
exp_scalar = exp(x_scalar);
exp_vector = exp(x_vector);
disp('Exponential of scalar:');
```

```
Exponential of scalar:
```

```
disp(exp_scalar);
```

```
    7.3891
```

```
disp('Exponential of vector:');
```

```
Exponential of vector:
```

```
disp(exp_vector);
```

```
    2.7183    7.3891   20.0855
```

```
disp('Compute the natural logarithm of a scalar and a vector')
```

```
Compute the natural logarithm of a scalar and a vector
```

```
x_scalar = 10;
x_vector = [1, exp(1), exp(2)];
log_scalar = log(x_scalar);
log_vector = log(x_vector);
disp('Natural logarithm of scalar:');
```

```
Natural logarithm of scalar:
```

```
disp(log_scalar);
```

```
    2.3026
```

```
disp('Natural logarithm of vector:');
```

```
Natural logarithm of vector:
```

```
disp(log_vector);
```

```
     0     1     2
```

```
disp('Compute the base-10 logarithm of a scalar and a vector')
```

```
Compute the base-10 logarithm of a scalar and a vector
```

```
x_scalar = 100;
x_vector = [10, 100, 1000];
log10_scalar = log10(x_scalar);
log10_vector = log10(x_vector);
disp('Base-10 logarithm of scalar:');
```

```
Base-10 logarithm of scalar:
```

```
disp(log10_scalar);
```

```
     2
```

```
disp('Base-10 logarithm of vector:');
```

```
Base-10 logarithm of vector:
```

```
disp(log10_vector);
```

```
     1     2     3
```

```
disp('Compute sine, cosine, and tangent of an angle in radians')
```

```
Compute sine, cosine, and tangent of an angle in radians
```

```
angle_radians = pi/6; % 30 degrees in radians
sin_value = sin(angle_radians);
cos_value = cos(angle_radians);
tan_value = tan(angle_radians);
disp('Sine of the angle:');
```

```
Sine of the angle:
```

```
disp(sin_value);
```

```
    0.5000
```

```
disp('Cosine of the angle:');
```

```
Cosine of the angle:
```

```
disp(cos_value);
```

```
    0.8660
```

```
disp('Tangent of the angle:');
```

```
Tangent of the angle:
```

```
disp(tan_value);
```

```
    0.5774
```

```
disp('Compute the maximum and minimum values of a vector')
```

```
Compute the maximum and minimum values of a vector
```

```
x_vector = [5, -3, 10, 2];
max_value = max(x_vector);
min_value = min(x_vector);
disp('Maximum value:');
```

```
Maximum value:
```

```
disp(max_value);
```

```
    10
```

```
disp('Minimum value:');
```

```
Minimum value:
```

```
disp(min_value);
```

```
    -3
```

```
disp('Compute element-wise maximum and minimum between two vectors')
```

```
Compute element-wise maximum and minimum between two vectors
```

```
x_vector = [1, 5, 3];
y_vector = [3, 2, 4];
max_result = max(x_vector, y_vector);
min_result = min(x_vector, y_vector);
disp('Element-wise maximum:');
```

```
Element-wise maximum:
```

```
disp(max_result);
```

```
     3     5     4
```

```
disp('Element-wise minimum:');
```

```
Element-wise minimum:
```

```
disp(min_result);
```

```
     1     2     3
```

```
disp('Compute the sum of elements in a vector')
```

```
Compute the sum of elements in a vector
```

```
x_vector = [1, 2, 3, 4, 5];
sum_result = sum(x_vector);
disp('Sum of elements:');
```

```
Sum of elements:
```

```
disp(sum_result);
```

```
    15
```

```
disp('Compute the complex conjugate of a complex vector')
```

```
Compute the complex conjugate of a complex vector
```

```
x_complex_vector = [2+3i, 4+5i, 7+6i];
conj_result = conj(x_complex_vector);
disp('Complex conjugate:');
```

```
Complex conjugate:
```

```
disp(conj_result);
```

```
   2.0000 - 3.0000i   4.0000 - 5.0000i   7.0000 - 6.0000i
```

```
disp('Generate random matrices')
```

```
Generate random matrices
```

```
rand_matrix_1 = rand(3); % 3x3 matrix
rand_matrix_2 = rand(2, 4); % 2x4 matrix
disp('Random matrix 1:');
```

```
Random matrix 1:
```

```
disp(rand_matrix_1);
```

```
    0.7094    0.6797    0.1190
    0.7547    0.6551    0.4984
    0.2760    0.1626    0.9597
```

```
disp('Random matrix 2:');
```

```
Random matrix 2:
```

```
disp(rand_matrix_2);
```

```
    0.3404    0.2238    0.2551    0.6991
    0.5853    0.7513    0.5060    0.8909
```