Lab 1

Question1

```
S = [150 150 150 160]; % June 2004 salaries
S = S + 10; % June 2005 salaries
S = S + 0.1*S; % June 2006 salaries (10% increase)
S
```

Output

```
>> Q1

S =

   176   176   176   187
```

Question2

```
clear
clc
V = [2 8 7 3 1 0 8 9] % original vector
result =(-1).^([1 4 3 1 1 0 4 4]) % final vector with 1s and -1s
```

Output

```
V =

     2     8     7     3     1     0     8     9


result =

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

Quesion3

```
% number of elements in array must be >= 4
V = input('Enter the vector V: ')'
%add 2 to the last 3 elements
V(end-2:end) = V(end-2:end) + 2

% reverse the order of the last 4 elements
V(end-3:end) = flip(V(end-3:end))

%add the elements number 1, 3, 5 .. to the elements number 2, 4, 6 ..
V(2:2:end) = V(2:2:end) + V(1:2:end-1)
```

Output

```
Enter the vector V:
[1,2,3,4,5]

V =                     V =

     1                       1
     2                       7
     3                       6
     4                       5
     5                       2


V =                     V =

     1                       1
     2                       8
     5                       6
     6                      11
     7                       2
```

Quesion4

```
% Generate a sequence of squares of natural numbers from 1 to 10
squares = (1:10).^2

%rotate
sequence = fliplr((1:8).^2)
```

Output

```
squares =

     1     4     9    16    25    36    49    64    81   100


sequence =

    64    49    36    25    16     9     4     1
```

Quesion5

```
% Define the 4x4 array M
M = [1 2 3 4; -1 -2 -3 -4; 1 2 3 4; -1 -2 -3 -4];

% Reflect M left-side right
M_lr = M(:, end:-1:1);

% Reflect M upside down
M_ud = M(end:-1:1, :);

% Swap columns 2 and 3 of M
M_cswap = M(:, [1 3 2 4]);

% Swap rows 1 and 4 of M
M_rswap = M([4 2 3 1], :);

% Shuffle the rows and columns of M
M_shuffle = M([1 3 4 2], [3 2 4 1]);
```

Output

```
M =

     1     2     3     4
    -1    -2    -3    -4
     1     2     3     4
    -1    -2    -3    -4


M_lr =

     4     3     2     1
    -4    -3    -2    -1
     4     3     2     1
    -4    -3    -2    -1


M_ud =

    -1    -2    -3    -4
     1     2     3     4
    -1    -2    -3    -4
     1     2     3     4
```

```
M_cswap =

     1     3     2     4
    -1    -3    -2    -4
     1     3     2     4
    -1    -3    -2    -4


M_rswap =

    -1    -2    -3    -4
    -1    -2    -3    -4
     1     2     3     4
     1     2     3     4


M_shuffle =

     3     2     4     1
     3     2     4     1
    -3    -2    -4    -1
    -3    -2    -4    -1
```

Quesion6

```
% Generate Matrix X
X = zeros(5, 5);
r = (1:5)';
X(:,1) = r;
X(:,end) = -r;

%From X Generate Y
Y = X';

%From X Generate Z
Z = [X(1,:)', X(2,:)', X(3,:)' X(2,:)', X(1,:)'];

%From X Generate W
W = zeros(5, 5);
W(:, 1:4) = 100;
W(:, 5) = 0.1* abs(X(:,5));
W(:, 1) = 2 * X(:, 1);
```

Output

```
X =

     1     0     0     0    -1
     2     0     0     0    -2
     3     0     0     0    -3
     4     0     0     0    -4
     5     0     0     0    -5


Y =

     1     2     3     4     5
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
    -1    -2    -3    -4    -5
```

```
Z =

     1     2     3     2     1
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
    -1    -2    -3    -2    -1


W =

    2.0000  100.0000  100.0000  100.0000    0.1000
    4.0000  100.0000  100.0000  100.0000    0.2000
    6.0000  100.0000  100.0000  100.0000    0.3000
    8.0000  100.0000  100.0000  100.0000    0.4000
   10.0000  100.0000  100.0000  100.0000    0.5000
```

Quesion7

```
% Define the matrix A and the vector B
A = zeros(5) % creates a 5-by-5 matrix of zeros
B = zeros(5,1) % creates a 5-by-1 column vector of zeros

A = [2 3 5 6 21;  5 0 2 2 0;6 7 8 9 11;0 13 17 5 6 ; 1 4 0 3 9]

B = [66; 127; 135; 19; 152]

% the rank of matrix A
S = (rank(A) == 5)

% Solve the system of equations
X = A \ B
```

Output

```
A =

     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


B =

     0
     0
     0
     0
     0


A =

     2     3     5     6    21
     5     0     2     2     0
     6     7     8     9    11
     0    13    17     5     6
     1     4     0     3     9
```

```
B =

    66
   127
   135
    19
   152


S =

  logical

   1


X =

   48.9421
   35.8279
  -17.3830
  -41.4722
    9.3514
```

Second Part

a) exp: Computes the exponential function, which is e raised to the power of a given number. For example, exp(1) returns the value of e (approximately 2.718).

b) log: Computes the natural logarithm of a given number. The natural logarithm is the logarithm to the base e. For example, log(10) returns the value of the natural logarithm of 10 (approximately 2.3026).

c) log2 and log10: Compute the base-2 and base-10 logarithms of a given number, respectively. For example, log2(8) returns the value of the base-2 logarithm of 8 (which is 3), and log10(1000) returns the value of the base-10 logarithm of 1000 (which is 3).

d) sqrt: Computes the square root of a given number. For example, sqrt(9) returns the value of 3.

e) sound: Plays sound data stored in a vector or matrix. The sound data can be in various formats such as waveform, frequency modulation, or white noise. For example, sound(x, Fs) plays the sound data in vector x with a sampling frequency of Fs Hz.

image: Displays grayscale or indexed images. The input image can be in various formats such as 2-D matrix, 3-D matrix, or an indexed image. For example, image(I) displays the grayscale image I.