RYERSON UNIVERSITY

Faculty of Engineering, Architecture and Science

Department of Electrical and Computer Engineering

Course Number	ELE 532 - Section 12
Course Title	Signals and Systems 1
Semester / Year	F2023

Instructor	Luella Marcos

Assignment Number	1

Assignment Title	Working with Matlab Functions, Visualization of Signals, and Signals Properties

Submission Date	2023/09/30
Due Date	2023/10/01

Student Name	Sarim Shahwar
Student ID	501109286
Signature*	SS

*By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a “0” on the work, an “F” in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at: www.ryerson.ca/senate/current/pol60.pdf.

ELE532 - Lab 1

Problem A.1

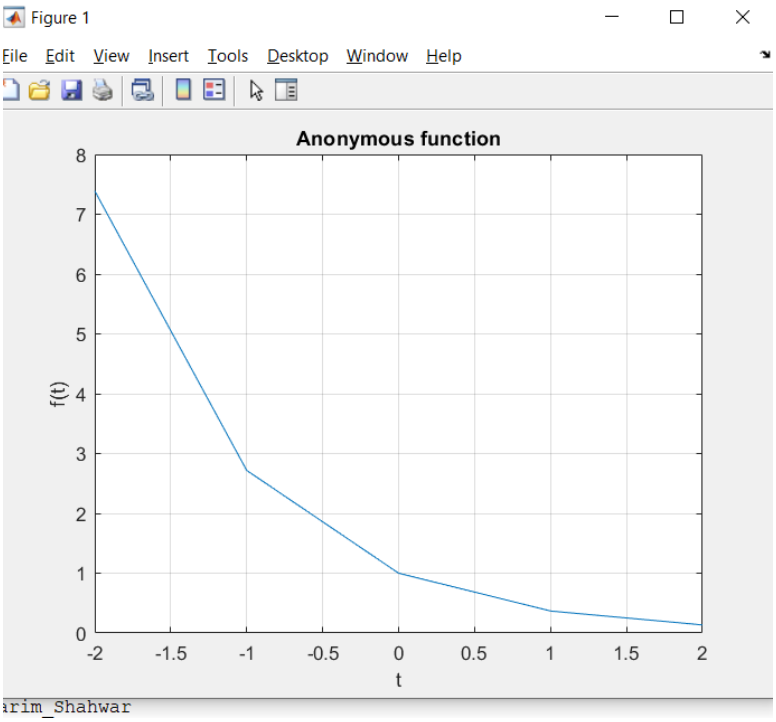


Figure 1.46: f(t) = e^−t cos(2πt) for t = (-2:2).

Code:

```
x= linspace(0,2*pi,100);
f = @(t)
exp(-t).*cos(2*pi*t);
t = (-2:2);
plot(t,f(t));
xlabel('t'); ylabel('f(t)');
grid; title("Anonymous
function");
```

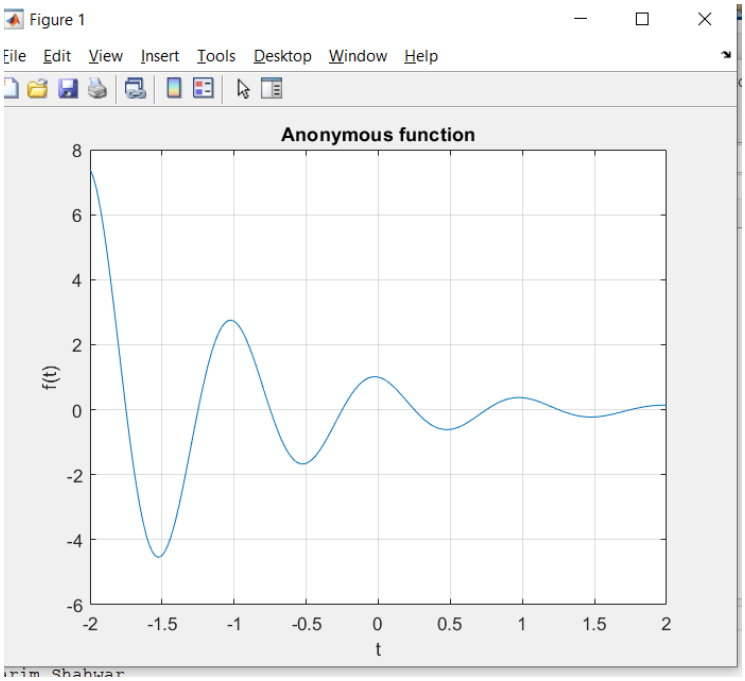
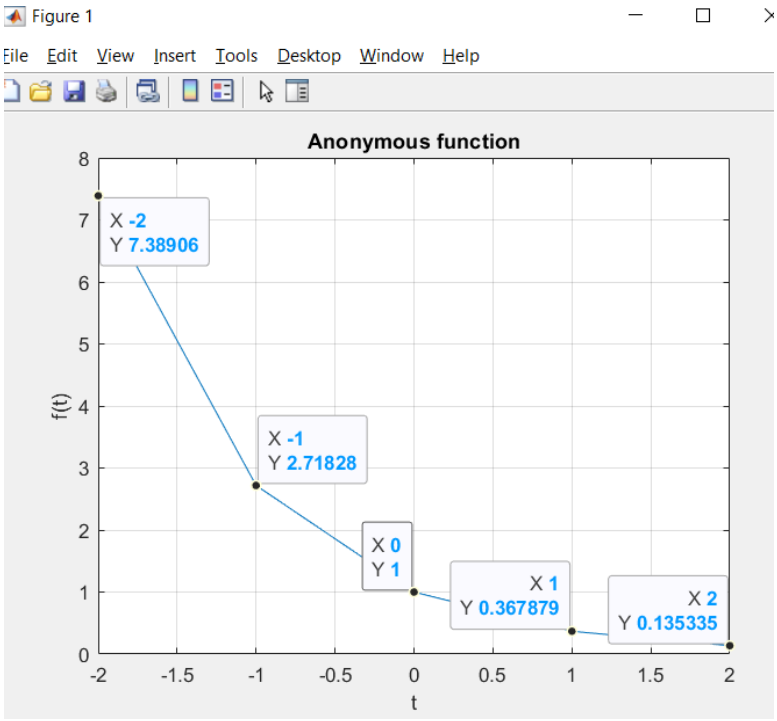


Figure 1.47: f(t) = e^−t cos(2πt) for t = (-2:0.01:2).

Code:

```
x= linspace(0,2*pi,100);
f = @(t)
exp(-t).*cos(2*pi*t);
t = (-2:0.01:2);
plot(t,f(t));
xlabel('t'); ylabel('f(t)');
grid; title("Anonymous
function");
```

Problem A.2



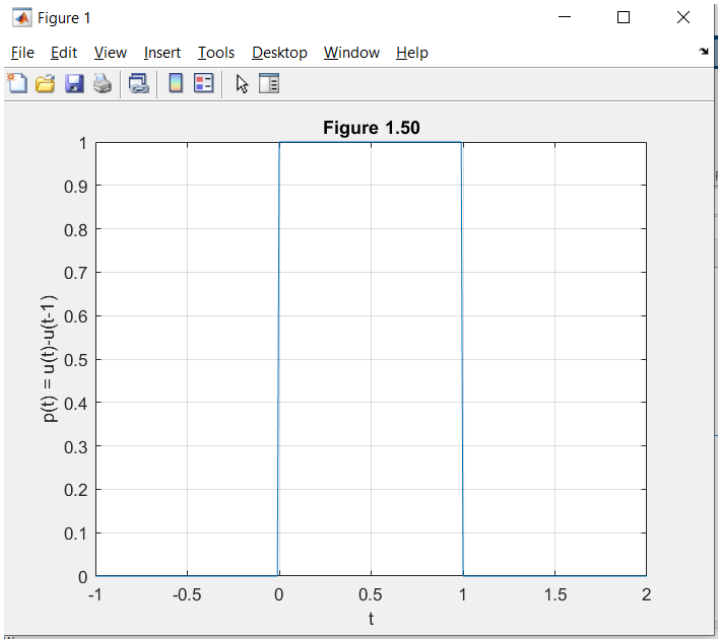
Code:

```
x= linspace(0,2*pi,100);
f = exp(-t);
t = (-2:1:2);
plot(t,f);
xlabel('t'); ylabel('f(t)');
grid; title("Anonymous
function");
```

Problem A.3

Looking at both of the graphs, the plots of the graphs are identical. Both of the functions produce the same graph and the same plots.

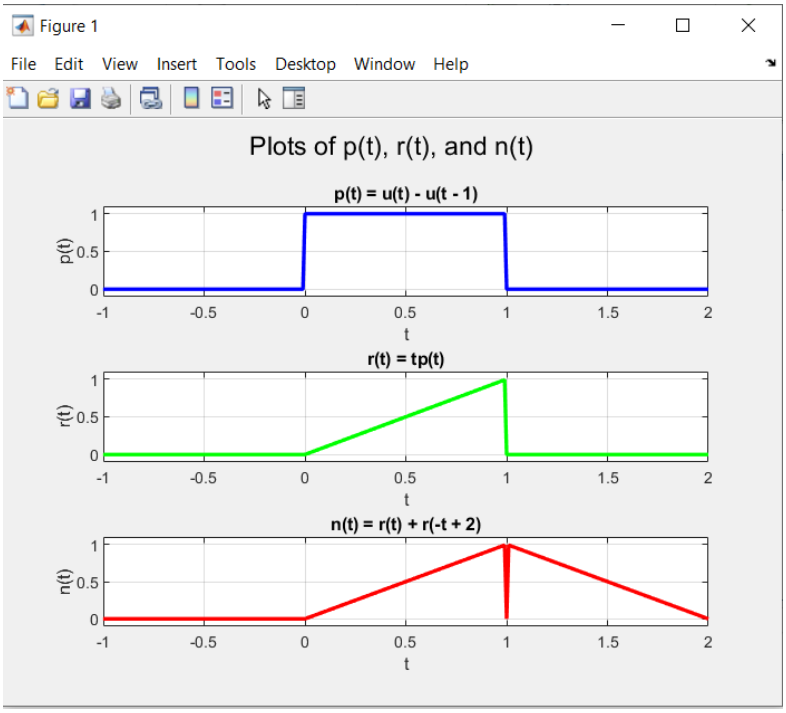
Problem B.1:



Code:

```
%% Part B
p = @(t)((t>=0)&(t<1));
t = (-1:0.01:2); plot(t,p(t));
xlabel('t'); ylabel('p(t) =
u(t)-u(t-1)'); title("Figure
1.50")
grid on;
```

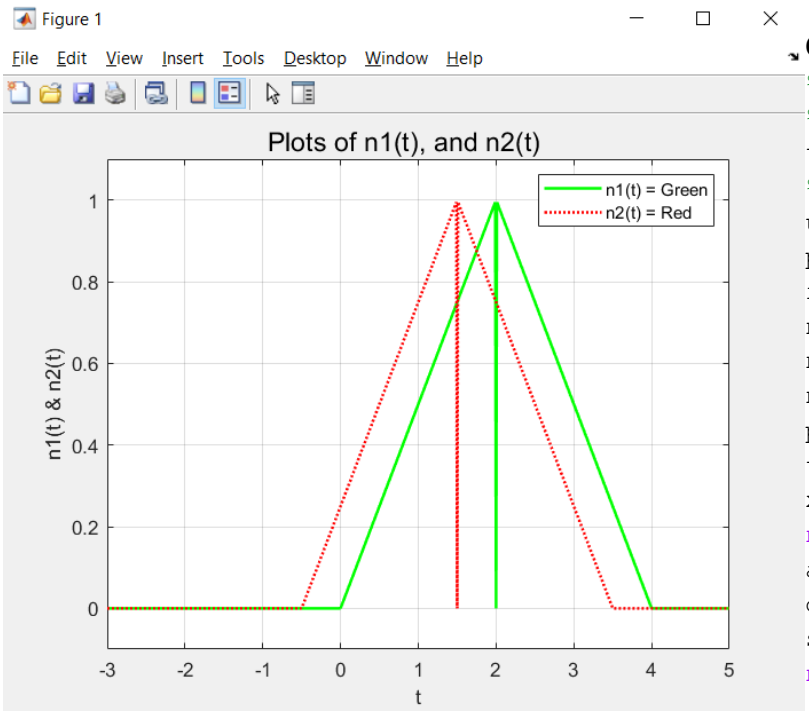
Problem B.2:



Code:

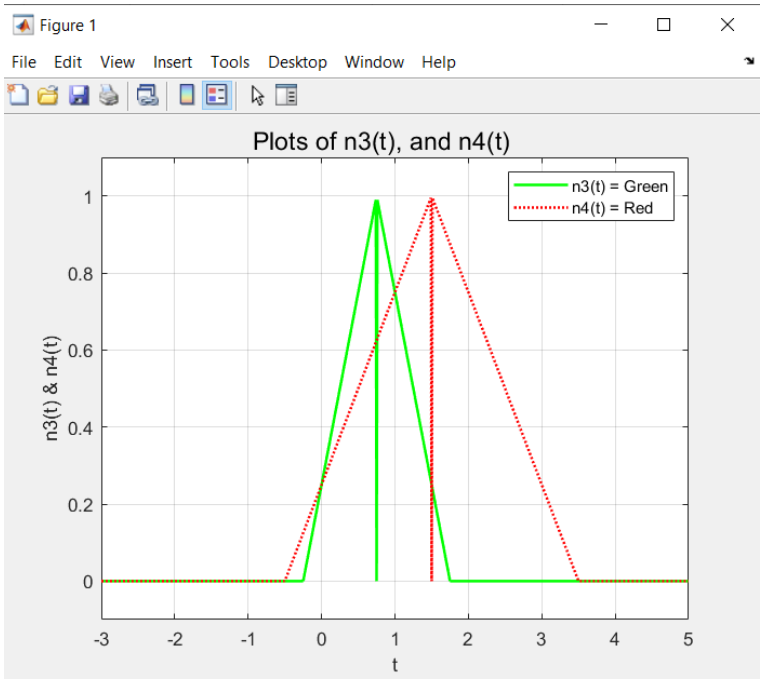
```
%% Part B2
%Value of t
t = (-2:0.01:2);
%Functions
u = @(t) 1.0.*(t>=0);
p = @(t) u(t)-u(t-1);
r = @(t) t.*p(t);
n = @(t) r(t) + r(-t+2);
subplot(3, 1, 1);
plot(t,p(t), 'b', 'LineWidth', 2);
xlabel('t'); ylabel('p(t)');
title("p(t) = u(t) - u(t - 1)")
axis([-1 2 -0.1 1.1]);
grid on;
subplot(3, 1, 2);
plot(t, r(t), 'g', 'LineWidth', 2);
xlabel('t'); ylabel('r(t)');
title("r(t) = tp(t)")
axis([-1 2 -0.1 1.1]);
grid on;
subplot(3, 1, 3);
plot(t, n(t), 'r', 'LineWidth', 2);
  xlabel('t'); ylabel('n(t)');
  title("n(t) = r(t) + r(-t + 2)")
axis([-1 2 -0.1 1.1]);
grid on;
sgtitle('Plots of p(t), r(t), and n(t)');
```

Problem B.3:



Code:

```
%% Part B3
%Value of t
t = (-3:0.01:5);
%Functions
u = @(t) 1.0.*(t>=0);
p = @(t) u(t)-u(t-1);
r = @(t) t.*p(t);
n = @(t) r(t) + r(-t+2);
n1 = @(t) n(0.5.*t);
n2 = @(t) n1(t+0.5);
plot(t, n1(t), 'g',
t,n2(t),':r', LineWidth=1.5 );
xlabel('t'); ylabel('n1(t) &
n2(t)');
axis([-3 5 -0.1 1.1]);
grid on;
sgtitle('Plots of n1(t), and
n2(t)');
```



Problem B.4:

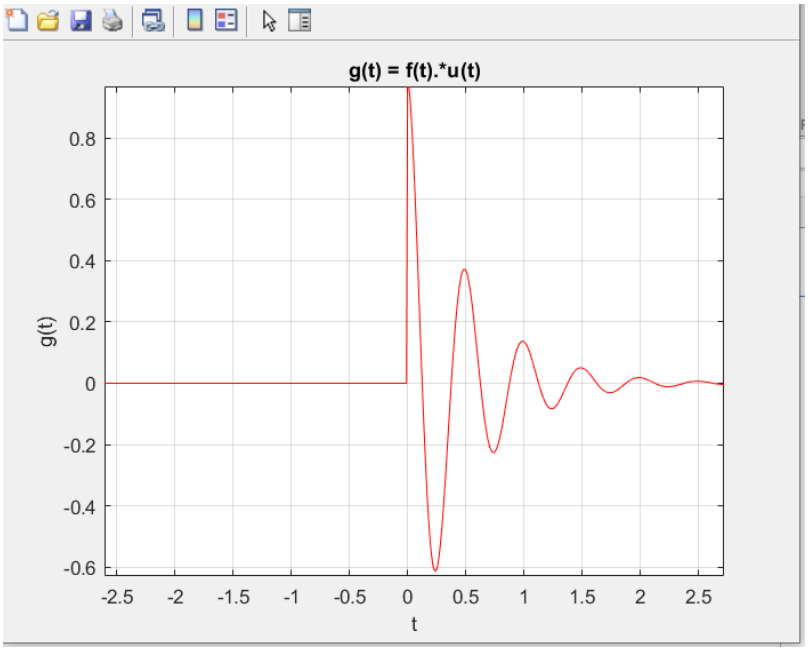
Code:

```
%% Part B4
%Value of t
t = (-3:0.01:5);
%Functions
u = @(t) 1.0.*(t>=0);
p = @(t) u(t)-u(t-1);
r = @(t) t.*p(t);
n = @(t) r(t) + r(-t+2);
n3 = @(t) n(t+0.25);
n4 = @(t) n3(0.5.*t);
plot(t, n3(t), 'g', t,n4(t),':r',
LineWidth=1.5 );
xlabel('t'); ylabel('n3(t) &
n4(t)');
axis([-3 5 -0.1 1.1]);
grid on;
sgtitle('Plots of n3(t), and
n4(t)');
```

Problem B.5:

Looking at the created graphs, both n2(t) and n4(t) produce the same graphs. Both of the plotted graphs have the same plotted points. Graphs n1 and n3 are not the same as one has a horizontal compression.

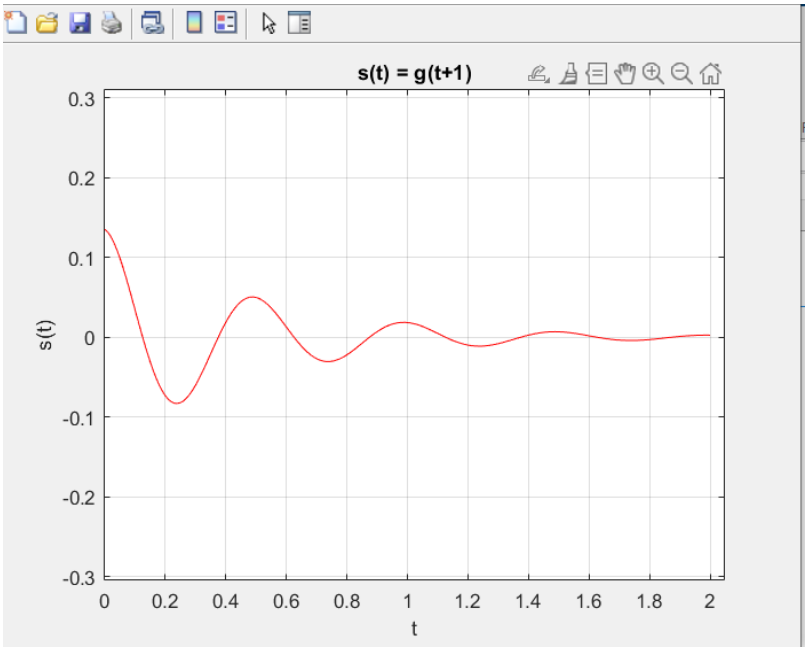
Problem C.1:



Code:

```
%% Part C1
%Value of t
t = (-4:0.01:4);
u = @(t) 1.0.*(t>=0);
f = @(t)
exp(-2.*t).*cos(4*pi*t);
g = @(t) f(t).*u(t);
plot(t, g(t), 'r');
xlabel('t');
ylabel('g(t)'); title("g(t)
= f(t).*u(t)");
axis([-2 2 -0.1 1.1]);
grid on;
```

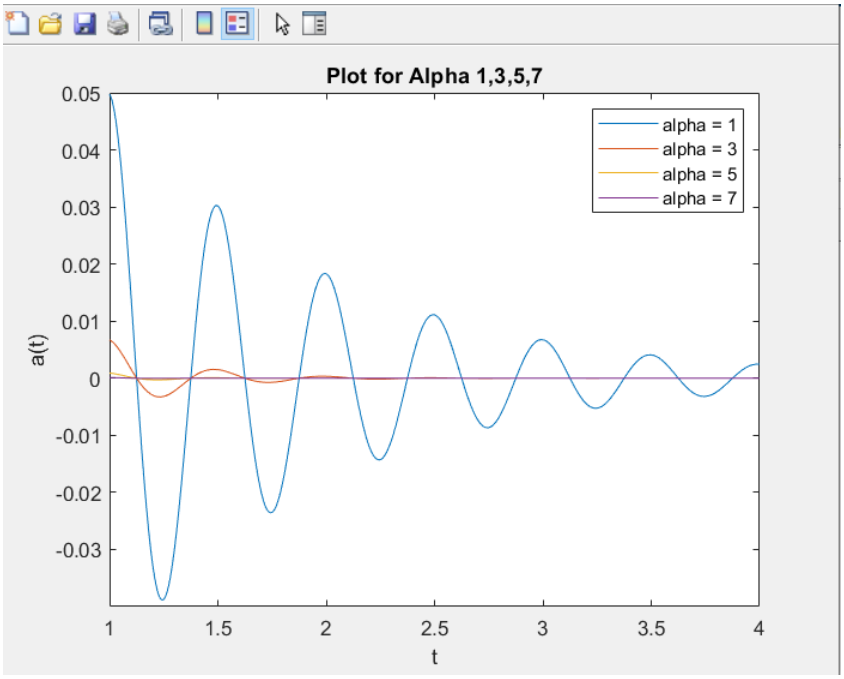
Problem C.2:



Code:

```
%% Part C2
%Value of t
t = (0:0.01:2);
u = @(t) 1.0.*(t>=0);
f = @(t)
exp(-2.*t).*cos(4*pi*t);
g = @(t) f(t).*u(t);
s = @(t) g(t+1);
plot(t, s(t), 'r');
xlabel('t');
ylabel('s(t)'); title("s(t)
= g(t+1)");
axis([-2 2 -0.1 1.1]);
grid on;
```

Problem C.3



Code:

```
%% Part C3
t = (1:0.01:4); %% (4/0.01)
+ 1 = 401
u = @(t) 1.0.*(t>=0);
% values of alpha
alpha_vals = [1, 3, 5, 7];
% Creating matrix to store
the signals for different
alpha values
s_alpha = zeros
(length(alpha_vals),
length(t));
% Calculate s_alpha(t) for
all alpha values using the
vector operations
% Converts column vector
alpha_matrix =
alpha_vals.';
t_matrix = t;
%created matrix is a 401 x 4
for alpha = 1:2:7
s = @(t) exp(-2).*exp(-alpha.*t).*cos(4*pi*t).*u(t);
plot(t,s(t));
xlabel("t"); ylabel("a(t)"); title("Plot for Alpha 1,3,5,7");
hold on
end
legend('alpha = 1', 'alpha = 3', 'alpha = 5', 'alpha = 7');
hold off;
```

Problem C.4:

The vector has 401 elements (from 0 to 4 with a step size of 0.01). So, the size of the matrix generated is 1604, which means it's a 4x401 matrix. Each section is labelled in the Matlab code above.

Problem D.1:

Function	Description
A(:)	The function displays the matrix components in a vertical format, starting from the left column to the right.
A([2 4 7])	When the operation is performed, the function outputs the data from Row 2 (Column 1), Row 4 (Column 1), and Row 2 (Column 2). These are the data points for 2,4 and 7.
[A >= 0.2]	Function outputs a 5 x 4 logical array for values that are 0 and 1. Such matrix displays a value of 0 for numbers that are not greater or equal to 0.2. For any value greater than or equal to 0.2, the matrix displays a value of 1.
A([A>=0.2])	This function displays all values greater than or equal to 0.2.
A([A>=0.2]) = 0	This function displays a “0” where any matrix value is greater or equal to 0.2. For indexes that are less than 0.2, the function displays their values.

Problem D.2:

```
For the code in part a, the execution time is the following:
Elapsed time is 0.014071 seconds.
```

Code:

```
%% Part D2 (c)
tic
load ('ELE532_Lab1_Data.mat')
Num_rows = size(B,1); %Allocaing Matrix Size
Num_cols = size(B,2); %Allocaing Matrix Size
for i = 1:1:Num_rows %First For loop for Rows
for j = 1:1:Num_cols %Second For loop for columns
if (abs(B(i,j)) < 0.01) % Absolute function of B(i,j) < 0.01
B(i,j) = 0; % Returning magnitude values below 0.01 to zero
end
end
end
fprintf('\nFor the code in part a, the execution time is the following:\n')
toc
```

```
For the code in part b, the execution time is the following:
Elapsed time is 0.239734 seconds.
```

Code:

```
%% Part D2 (c continued)
tic
load ('ELE532_Lab1_Data.mat')
%Creted the same function as posted in part D1 but with the criteria of
%part D2
B([abs(B)>= 0.01]) = 0
fprintf('\nFor the code in part b, the execution time is the following:\n')
toc
```

Problem D.3:

```
For the Audio Data Set there are 58>>
```

Code:

```
%% Part D3
load ('ELE532_Lab1_Data.mat')
Num_rows = size(x_audio,1); %Allocaing Matrix Size
Num_cols = size(x_audio,2); %Allocaing Matrix Size
threshold = 0;
for i = 1: Num_rows
for j = 1: Num_cols
if(abs(x_audio(i,j) == 0))
threshold = threshold + 1;
end
end
end
fprintf("\nFor the Audio Data Set there are " + threshold);
sound(x_audio,8000)
```