Toronto Metropolitan University

Department of Electrical, Computer, & Biomedical Engineering
Faculty of Engineering & Architectural Science

Course Title:	Signals and Systems I
Course Number:	ELE-532
Semester/Year (e.g.F2016)	W2025

Instructor:	Dr.Beheshti

Assignment/Lab Number:	1
Assignment/Lab Title:	Working with Matlab, Visualization of Signals

Submission Date:	2025-09-28
Due Date:	2025-09-28

Student LAST Name	Student FIRST Name	Student Number	Section	Signature*
Malik	Hamza	501112545	12	H.M
Aneesh	Kattoji	501233584	12	A.K

Figures and Discussion

Part A: Anonymous functions and plotting continuous functions

Problem A.1:

Figure 1.46

Figure 1.46
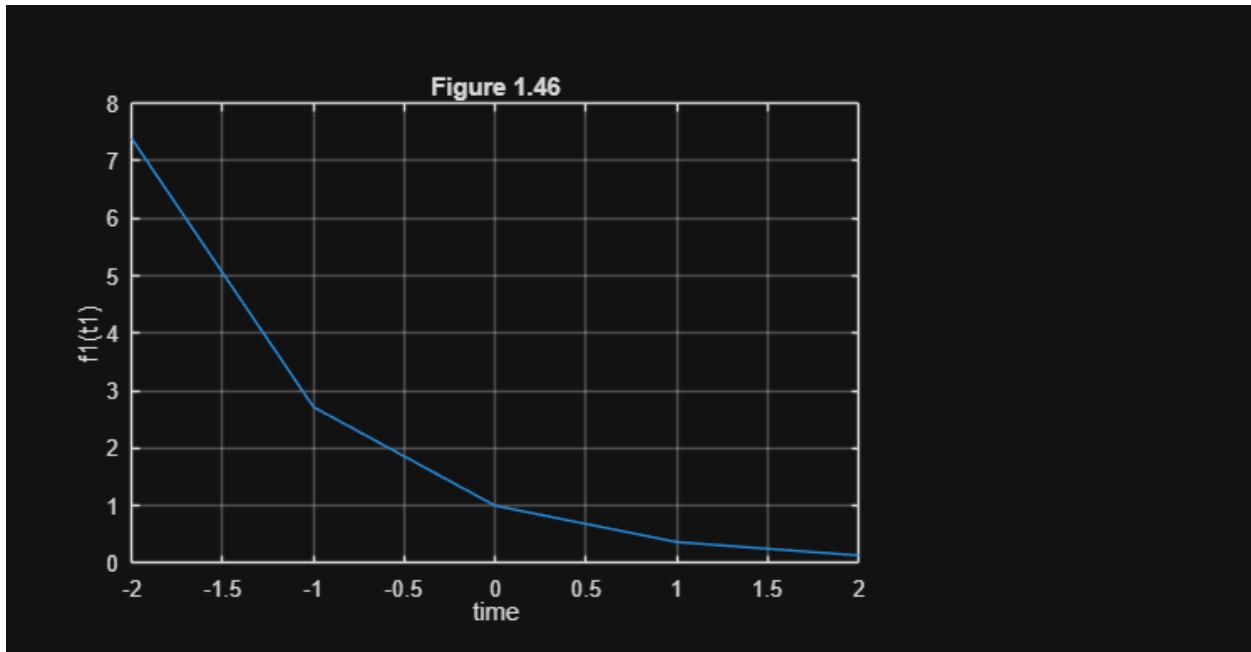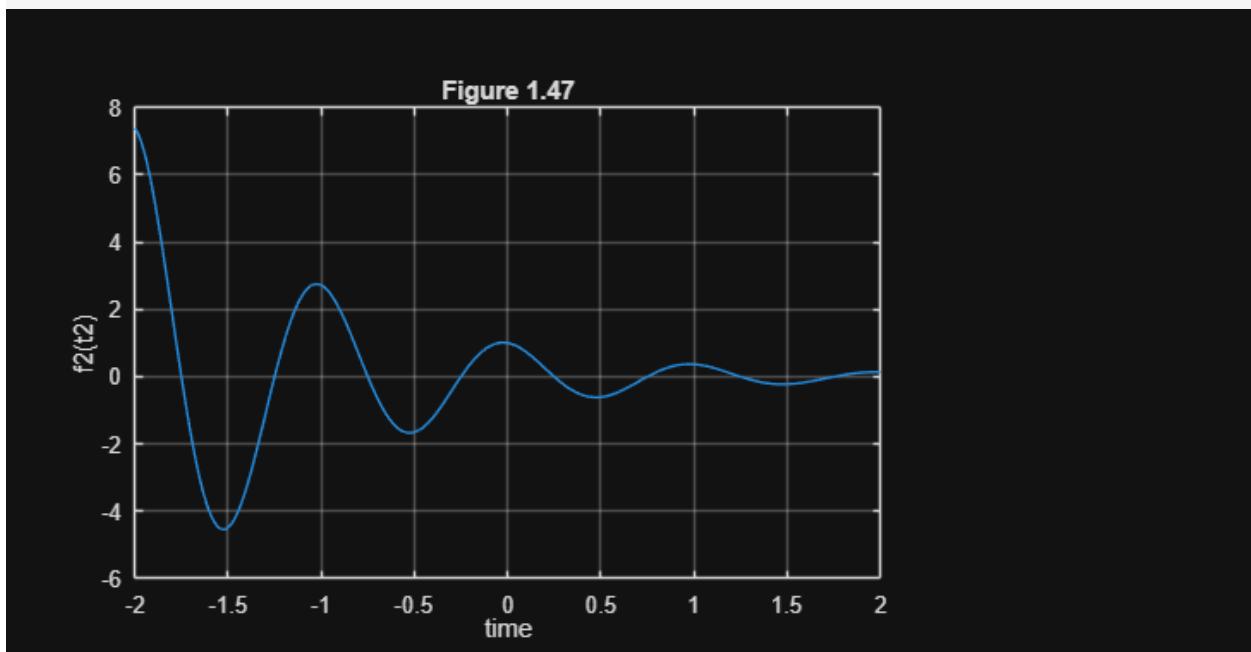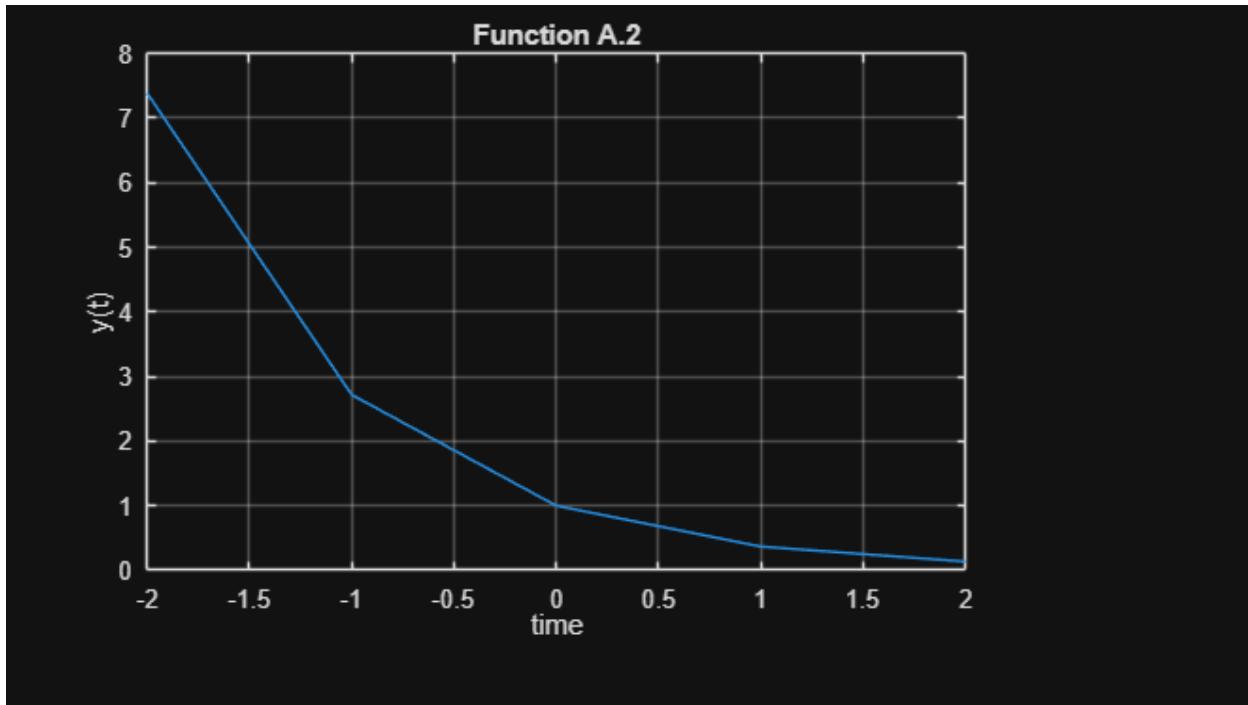
Figure 1.47

Figure 1.47

Problem A.2:
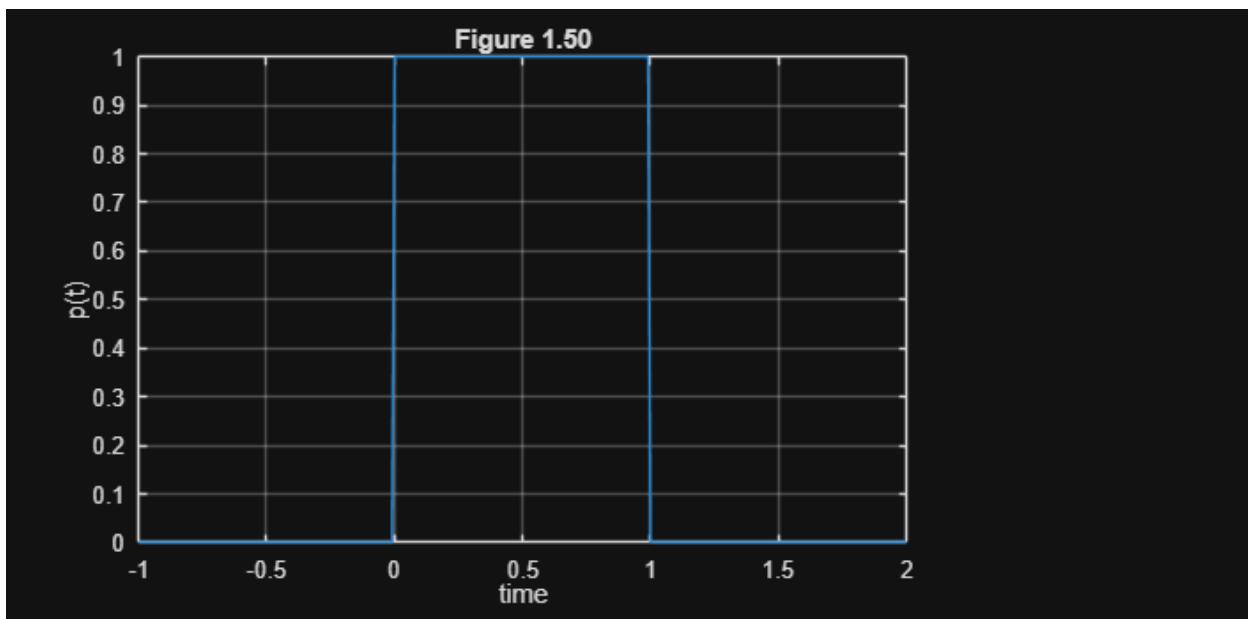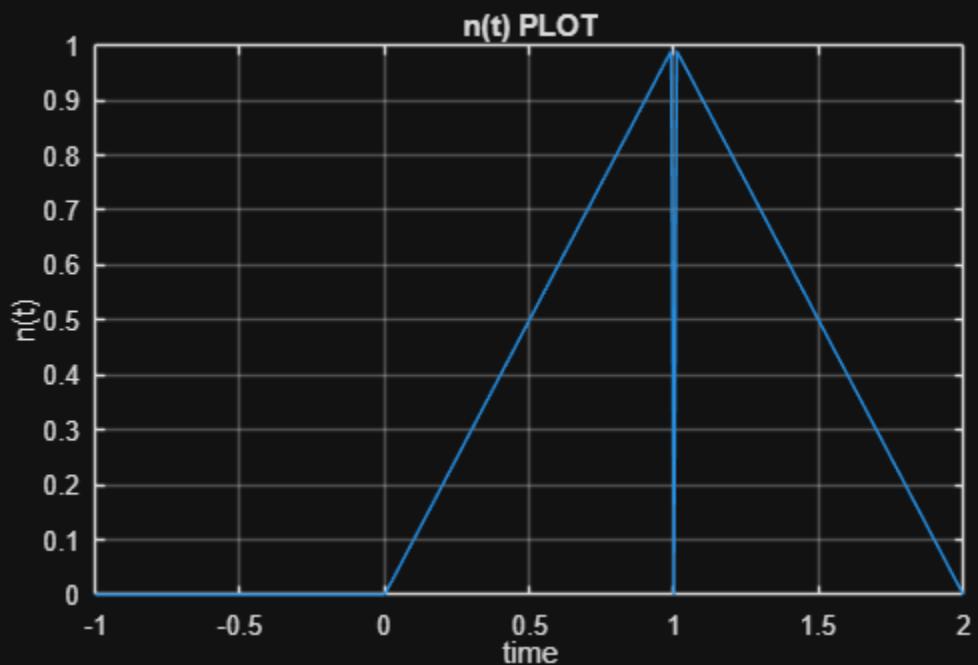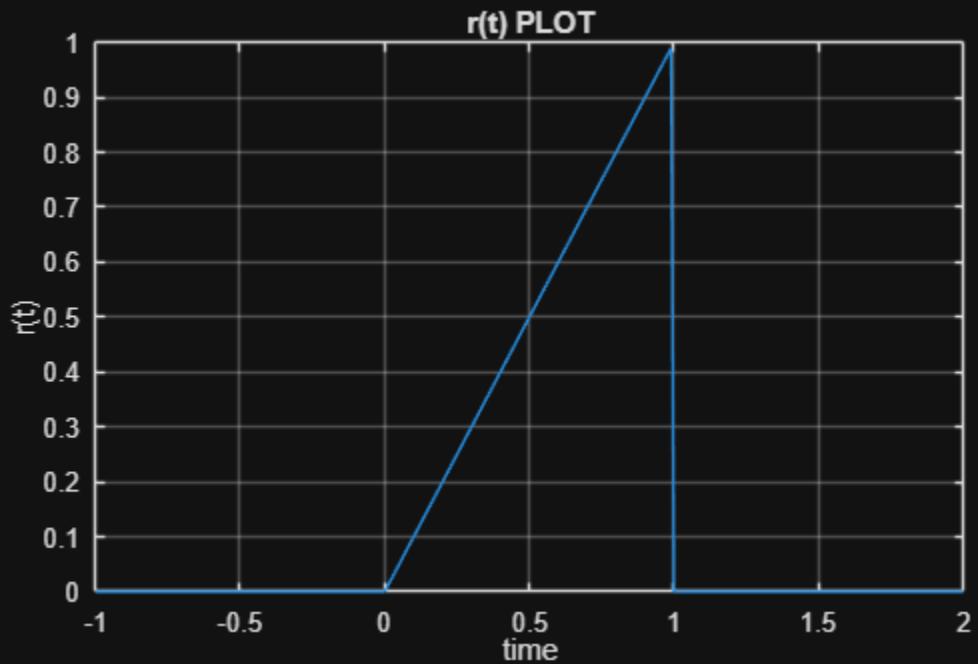

Function A.2

Problem A.3:

The function plotted in problem A.2 is the exact same as the visual

representation of the function of figure 1.46 in problem A.1

Part B: Time shifting and time scaling

Problem B.1

Figure 1.50
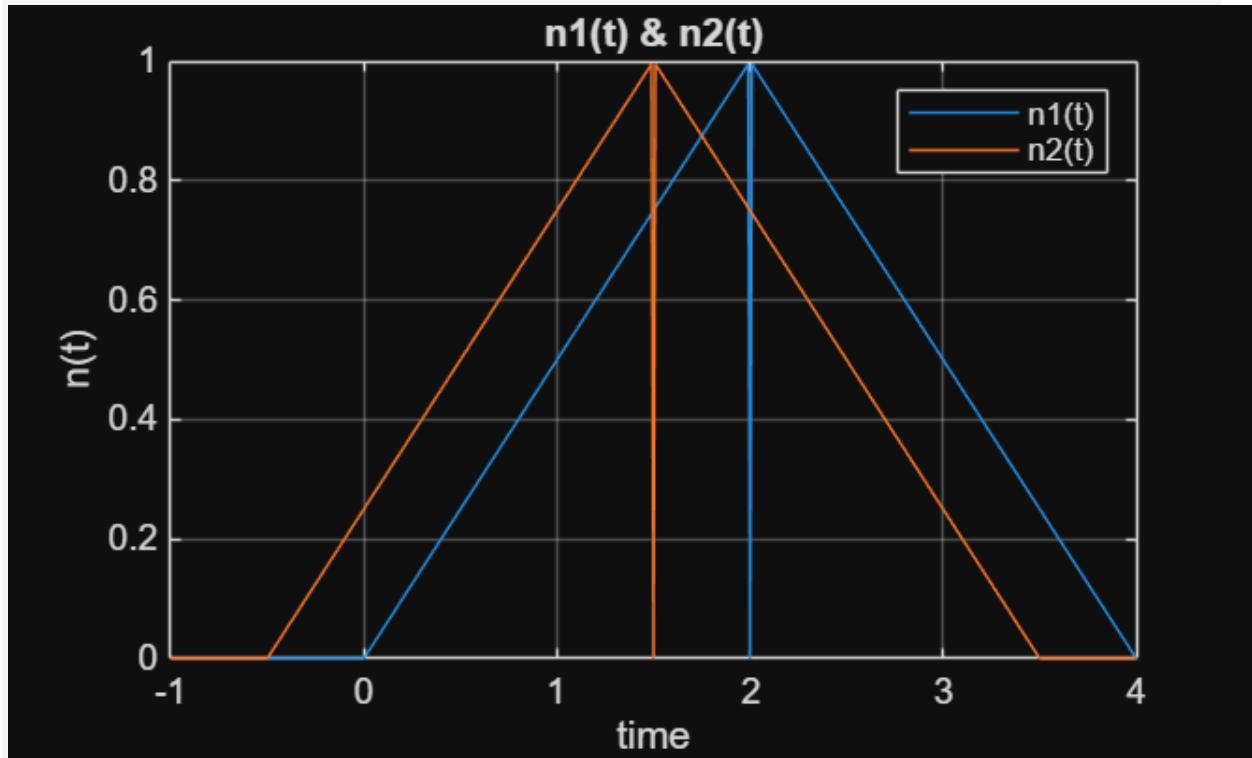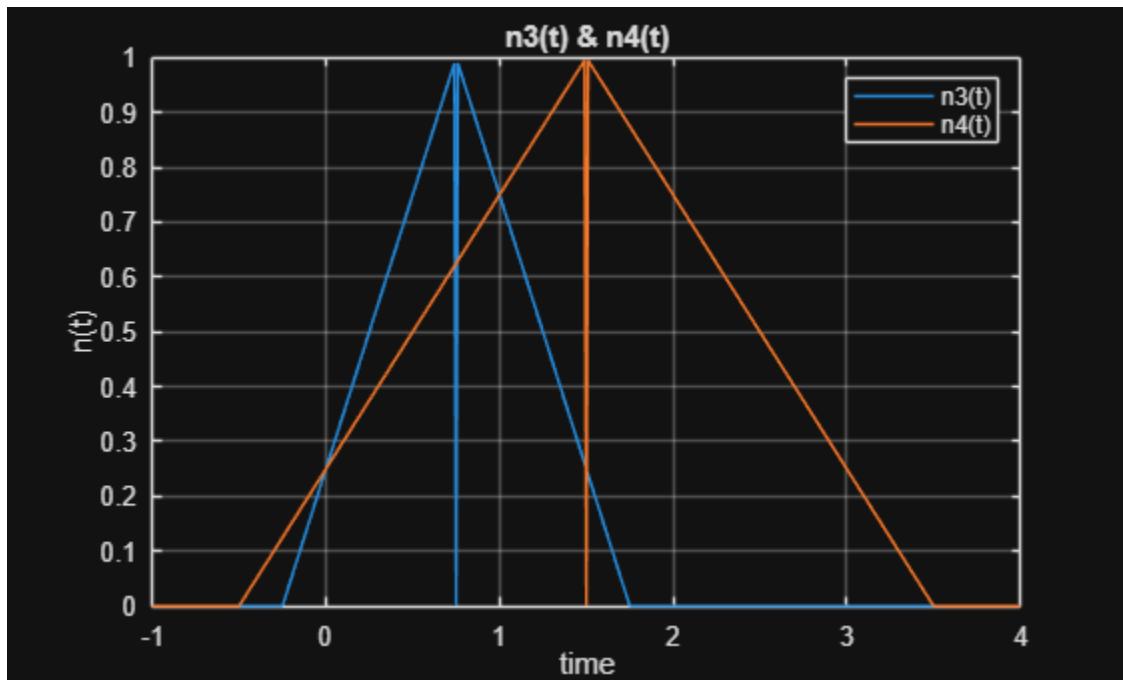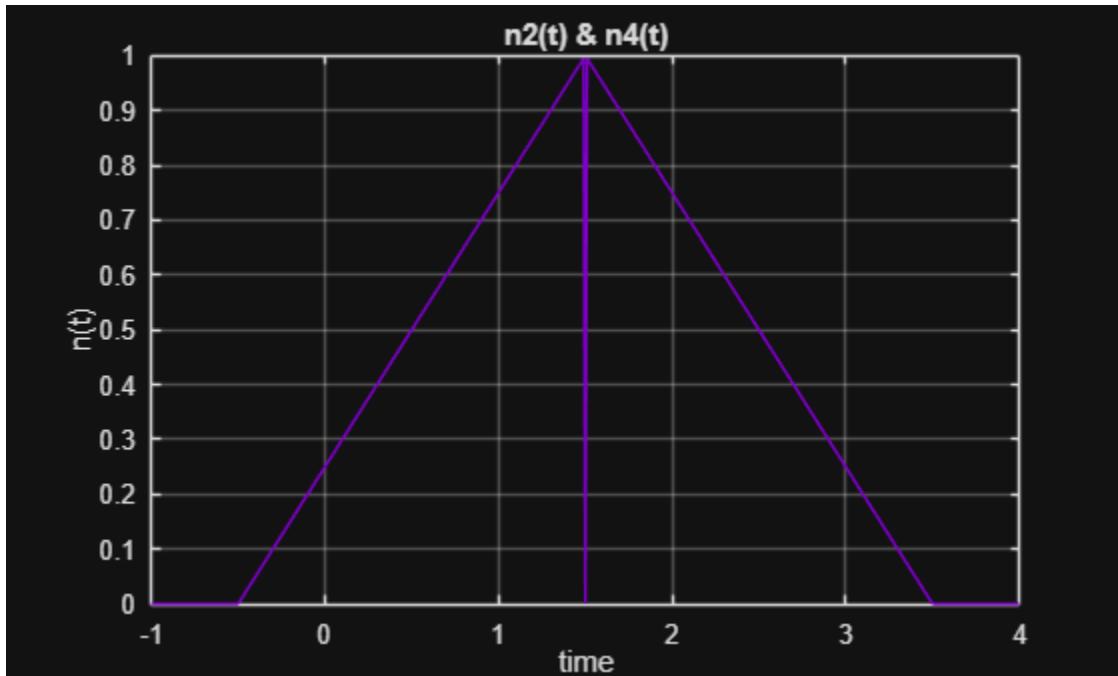
Problem B.2

%Problem B.3

Problem B.4
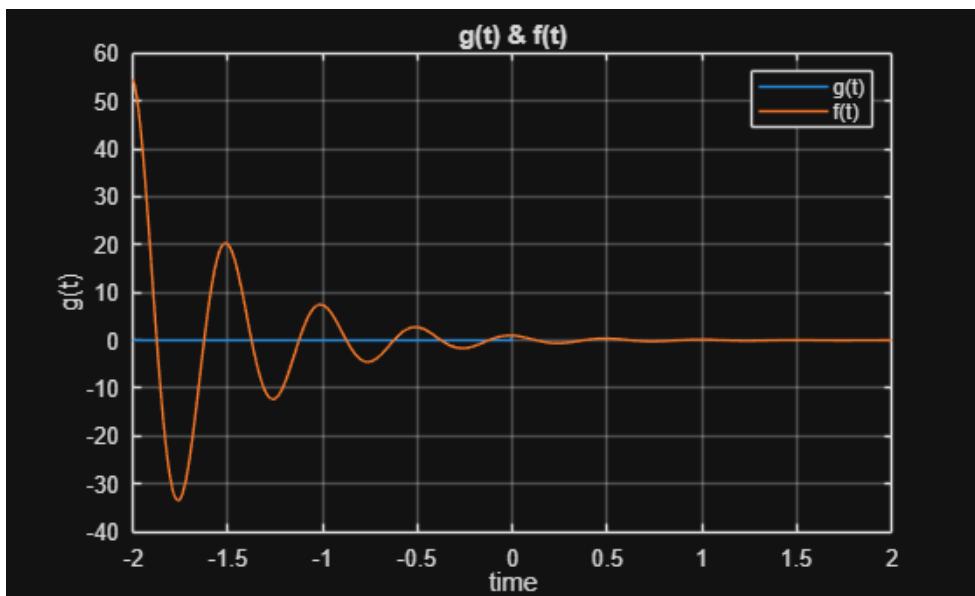
Problem B.5
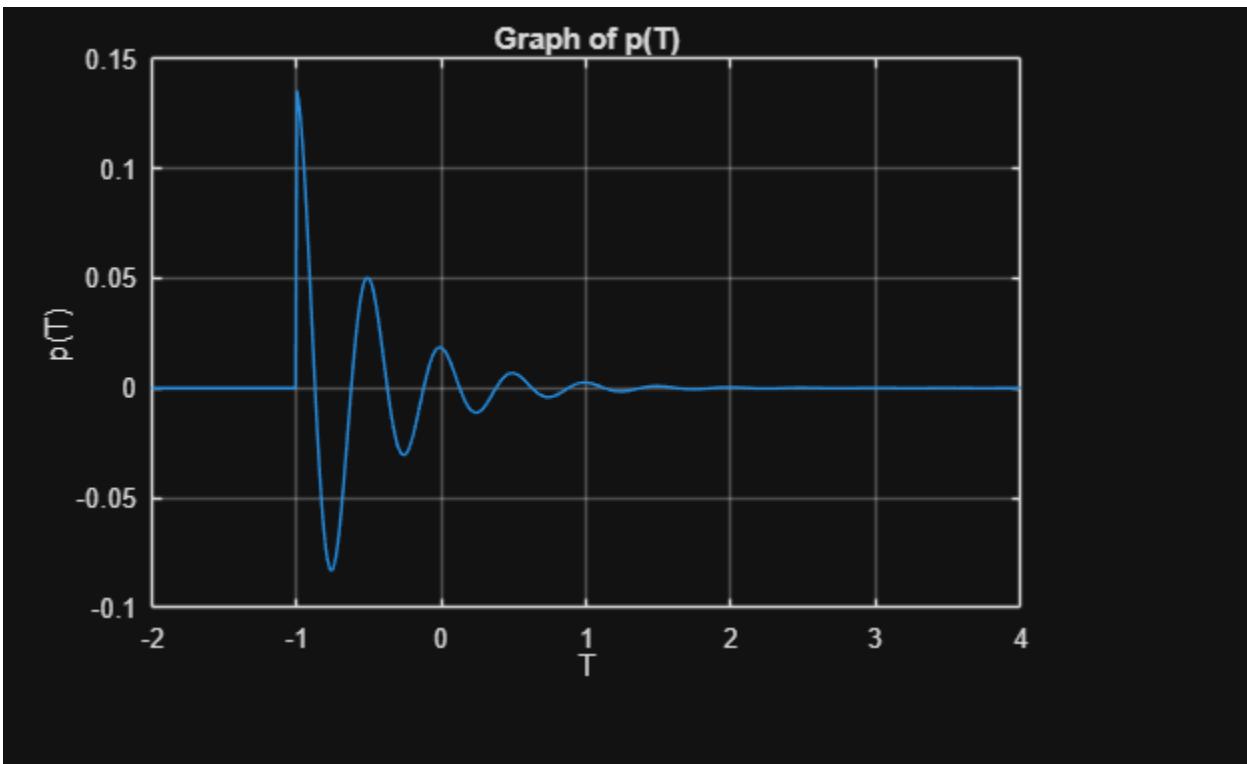
 it seems the n4 and n2 are the same graph as they overlap

this makes sense, because if you expand both equation until

they are in terms of t, they become the same equation
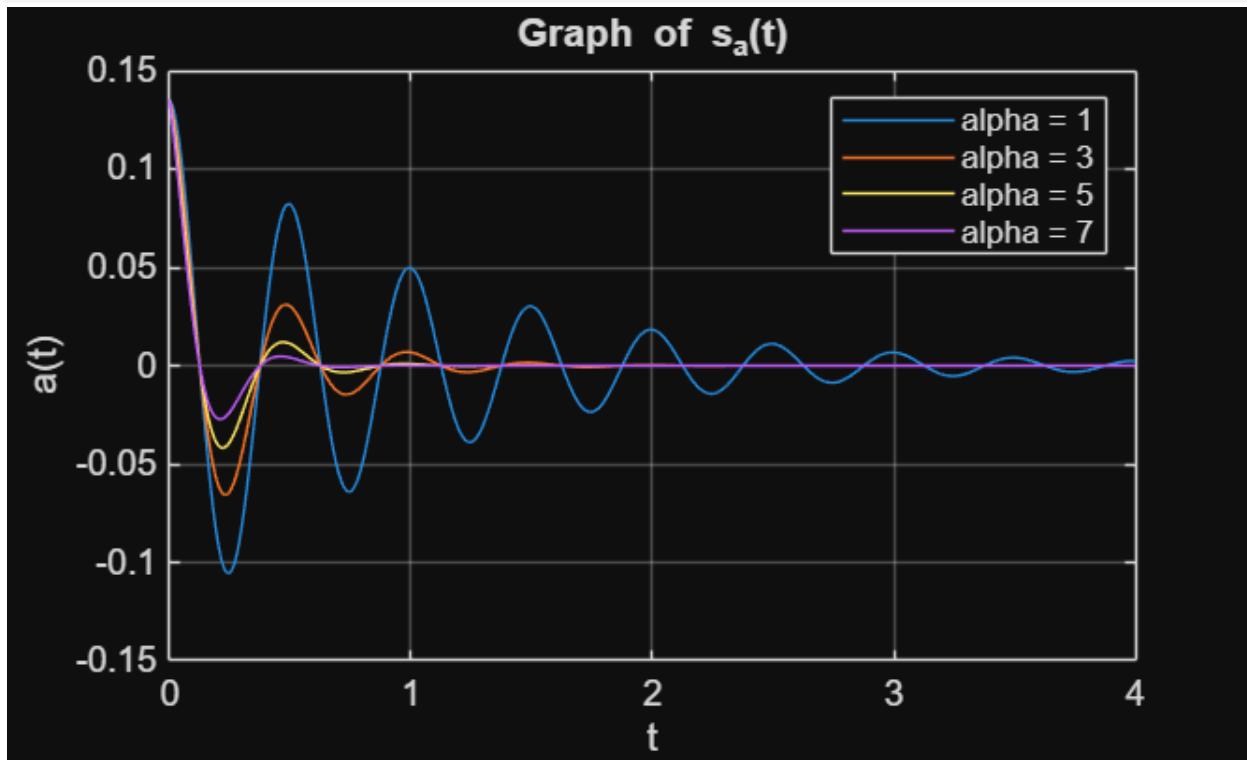
%Problem C.1

Problem C.2

Graph of p(T)

Problem C.3

Graph of $s_a(t)$

%Problem C.4

the size of the array

4x401x

%% Part D: Array Indexing

%Problem D.1

ans = *20×1*

  0.5377

  1.8339

-2.2588

  0.8622

  0.3188

-1.3077

-0.4336

  0.3426

  3.5784

  2.7694

-1.3499

  3.0349

  0.7254

-0.0631

  0.7147

ans = *20×1*

(*Rows 1:10*)

  0.5377

  1.8339

-2.2588

  0.8622

  0.3188

-1.3077

-0.4336

0.3426

3.5784

2.7694

-1.3499

3.0349

0.7254

-0.0631

0.7147

A([2 4 7])

ans = *1×3*

  1.8339   0.8622 -0.4336

ans = *1×3*

(*Columns 1:3*)

  1.8339   0.8622 -0.4336

[A>=0.2]

ans = *5×4 logical array*

 1 0 0 0

 1 0 1 0

 0 1 1 1

 1 1 0 1

 1 1 1 1

ans = *5×4 logical array*

(*Rows 1:5 | Columns 1:4*)

 1 0 0 0

 1 0 1 0

 0 1 1 1

 1 1 0 1

 1 1 1 1

A([A >= 0.2])

ans = *13×1*

0.5377

1.8339

0.8622

0.3188

0.3426

3.5784

2.7694

3.0349

0.7254

0.7147

1.4897

1.4090

1.4172

ans = *13×1*

(*Rows 1:10*)

0.5377

1.8339

0.8622

0.3188

0.3426

3.5784

2.7694

3.0349

0.7254

0.7147

1.4897

1.4090

1.4172

PART A execution time:

Elapsed time is 0.006380 seconds.

PART B execution time:

Elapsed time is 0.005046 seconds.

% Problem D.3

ans = 58

For the audio data set, the threshold is: 58

Full Code

```matlab
clear;

% Name: Hamza Malik, Student number: 501112545, Section: 12

% Aneesh Kattoji, Student number: 501233584, Section: 12

%%

% ELE532_LAB_1: Working with Matlab Functions, Visualization of
Signals, and Signals Properties

%

%% Part A: Anonymous functions and plotting continuous functions

%Problem A.1:

  %Figure 1.46

t1 = (-2:2);

f1= @(t1) exp(-t1).*cos(2*pi*t1);

figure;

plot(t1,f1(t1));

xlabel('time'); ylabel('f1(t1)'); title('Figure 1.46'); grid;

hold off;

  %Figure 1.47

t2 = (-2:0.01:2);

f2= @(t2) exp(-t2).*cos(2*pi*t2);

figure;

plot(t2,f2(t2));

xlabel('time'); ylabel('f2(t2)'); title('Figure 1.47'); grid;

hold off;

%Problem A.2:

t = (-2:1:2);

y = exp(-t);

figure;
```

```matlab
plot(t,y);

xlabel('time'); ylabel('y(t)'); title('Function A.2'); grid;

hold off;

%Problem A.3:

  %The function plotted in problem A.2 is the exact same as the visual

  %representation of the function of figure 1.46 in problem A.1

%% Part B: Time shifting and time scaling

%Problem B.1

t = (-1:0.01:2);

p = @(t) 1.0.*((t>=0)&(t<1));

figure;

plot(t,p(t));

xlabel('time'); ylabel('p(t)'); title('Figure 1.50'); grid;

hold off;

%Problem B.2

r = @(t) p(t) .* t;

figure;

plot(t,r(t));

xlabel('time'); ylabel('r(t)'); title('r(t) PLOT'); grid;

hold off;

n = @(t) r(t) + r(-t+2);

figure;

plot(t,n(t));

xlabel('time'); ylabel('n(t)'); title('n(t) PLOT'); grid;

hold off;

%Problem B.3

t = (-1:0.01:4);
```

```matlab
n1 = @(t) n(t*0.5);

n2 = @(t) n1(t + 0.5);

figure;

plot(t, n1(t), t, n2(t));

legend('n1(t)','n2(t)');

xlabel('time'); ylabel('n(t)'); title('n1(t) & n2(t)'); grid;

hold on;

%Problem B.4

n3 = @(t) n(t + 0.25);

n4 = @(t) n3(t .* 0.5);

figure;

plot(t, n3(t), t, n4(t));

legend('n3(t)','n4(t)');

xlabel('time'); ylabel('n(t)'); title('n3(t) & n4(t)'); grid;

hold on;

%Problem B.5

figure;

plot(t, n2(t),t, n4(t),'Color',[0.5 0 0.8]);

xlabel('time'); ylabel('n(t)'); title('n2(t) & n4(t)'); grid;

hold on;

% it seems the n4 and n2 are the same graph as they overlap

%this makes sense, because if you expand both equation until

%they are interms of t, they become the same equation

%Problem C.1

t = (-2:0.01:2);

f = @(t) exp(-2.*t).*cos(4.*pi.*t);

u = @(t) 1.0.*(t>=0);

g = @(t) f(t).*u(t);
```

```matlab
figure;

plot(t, g(t), t, f(t));

legend('g(t)','f(t)');

xlabel('time'); ylabel('g(t)'); title('g(t) & f(t)'); grid;

hold on;

%Problem C.2

T = (-2:0.01:4);

p = @(T) exp(-2).*g(T+1);

figure;

plot(T, p(T));

xlabel('T'); ylabel('p(T)'); title('Graph of p(T)'); grid;

hold off;

% Problem C.3

u = @(t) 1.0.*(t>=0);

t = (0:0.01:4);

a = [1, 3, 5, 7];

s_alpha = zeros(length(a), length(t));

for alpha = a

  s = @(t) exp(-2).*exp(-alpha.*t).*cos(4*pi*t).*u(t);

  plot(t,s(t));

  hold on;

end

xlabel("t"); ylabel("a(t)"); title("Graph of s_a(t)"); grid;

legend('alpha = 1', 'alpha = 3', 'alpha = 5', 'alpha = 7');

hold off;

%Problem C.4

fprintf("the size of the array")

fprintf('%dx',size(s_alpha));
```

```matlab
% Matrix would be 4 x 399 if transposed properly

%% Part D: Array Indexing

%Problem D.1

load("C:\Users\anees\Downloads\ELE532_Lab1_Data.mat")

A(:)

A([2 4 7])

[A>=0.2]

A([A >= 0.2])

A([A>=0.2]) = 0;

% ✅ Fix for Problem D.2: define B if it doesn't exist

Num_rows = size(B,1);

Num_cols = size(B,2);

for i = 1:Num_rows

 for j = 1:Num_cols

    if abs(B(i,j)) < 0.01

        B(i,j) = 0;

    end

 end

end

B([abs(B)>= 0.01]) = 0;

% part Ci

tic

for i = 1:Num_rows

 for j = 1:Num_cols

    if abs(B(i,j)) < 0.01

        B(i,j) = 0;

    end

 end
```

```matlab
end

fprintf('\nPART A execution time: ')

toc

% part Cii

tic

B([abs(B)>= 0.01]) = 0;

fprintf('PART B execution time: ')

toc

% Problem D.3

Num_rows = size(x_audio,1);

Num_cols = size(x_audio,2);

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

f = @(x_audio) sum(~x_audio(:));

f(x_audio)

fprintf("For the audio data set, the threshold is: " + threshold);

sound(x_audio,8000);
```