Cmpe 362 | Homework 1

Problem 1

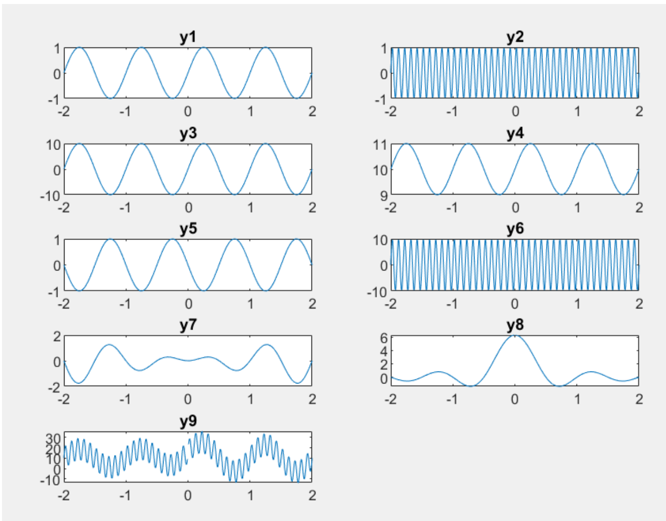
Code & Comments:

```
t = (-2:0.01:2);       % t and y1,y2..,y9 are given in the question

y1 = sin(2*pi.*t);

y2 = sin(2*pi*10.*t);

y3 = 10*sin(2*pi.*t);

y4 = sin(2*pi.*t)+10;

y5 = sin(2*pi.*(t-0.5));

y6 = 10*sin(2*pi*10.*t);

y7 = sin(2*pi.*t).*t;

y8 = sin(2*pi.*t)./t;

y9 = y1+y2+y3+y4+y5+y6+y7+y8;

%{

 For each sinusoid, there should be a subplot (5x2 subplots), and its names are written as their titles. Since the y values are dependent on t values, I've plotted as plot(t,y) , so x coordinate corresponds to t values, y coordinates corresponds to its yn value. I put x limits as [-2,2], since limit of t vector is also [-2,2].

}%

subplot(5,2,1), plot(t, y1), xlim([-2 2]), title('y1')

subplot(5,2,2), plot(t, y2), xlim([-2 2]), title('y2')

subplot(5,2,3), plot(t, y3), xlim([-2 2]), title('y3')

subplot(5,2,4), plot(t, y4), xlim([-2 2]), title('y4')

subplot(5,2,5), plot(t, y5), xlim([-2 2]), title('y5')

subplot(5,2,6), plot(t, y6), xlim([-2 2]), title('y6')

subplot(5,2,7), plot(t, y7), xlim([-2 2]), title('y7')

subplot(5,2,8), plot(t, y8), xlim([-2 2]), title('y8')

subplot(5,2,9), plot(t, y9), xlim([-2 2]), title('y9')
```

Explanation:

T vector and formulas of sinusoids were given in the question. I've learned how to use subplot, plot, xlim and title functions in matlab. I've used 5x2 subplots and plotted each of the sinusoid, given titles and decided x limits. I've noticed that frequencies of y1,y2,y3,y4,y5 and y6 are fixed(as can be seen from the figures, shapes of waves are periodic. However, in y7, the sinusoid formula is multiplied with t, so its shape changes as t values change, but it's symmetric with respect to x=0. The reason of this symmetry is that sin and t^1 are odd functions, odd x odd = even, so -t and t gives same result for y7. There is a space in shape of y8 at x=0; because sinusoid can't be divided by 0. This space can be also seen in y9, because it's an addition and one of the elements of it is y8.

Figure:



Problem 2

Code & Comments:

z = randn(1, 401);   % generates 401 random numbers, which are zero-mean, unit variance Gaussian distributed numbers.

z = 0.1.*z;   % numbers are multiplied by 0.1

t = (-2:0.01:2);   % t vector is given

y10 = z;   % y10,y,11….,y19 are given

y11 = z+t;

y12 = z+y1;

y13 = z.*y1;

y14 = sin(2*pi.*z).*t;

y15 = sin(2*pi.*(t+z));

y16 = z.*y2;

y17 = sin(2*pi.*(t+10.*z));

y18 = y1./z;

y19 = y11+y12+y12+y14+y15+y16+y17+y18;

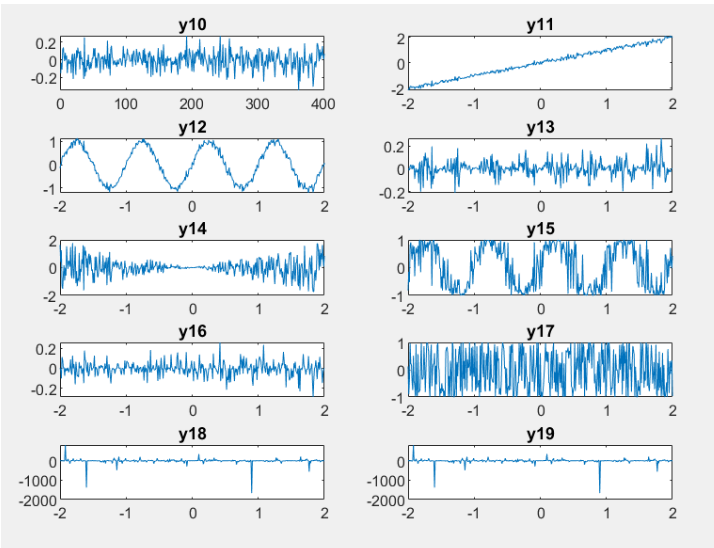
% For each sinusoid, there should be a subplot (5x2 subplots), and its names are written as their titles. I've plotted as plot(t,y) , because they are dependent on t or y values in previous question, which are also dependent on t. So x coordinate corresponds to t values, y coordinate corresponds to its yn value. I put x limits as [-2,2], since limits of t vector is also [-2,2]. (as in question 1) However, y10=z is independent of t, so I've plotted it as plot(y10), x coordinate corresponds to number of random values(401). So xlim is [0,401].

```
subplot(5,2,1), plot(y10), xlim([0 401]), title('y10');

subplot(5,2,2), plot(t, y11), xlim([-2 2]), title('y11');

subplot(5,2,3), plot(t, y12), xlim([-2 2]), title('y12');

subplot(5,2,4), plot(t, y13), xlim([-2 2]), title('y13');

subplot(5,2,5), plot(t, y14), xlim([-2 2]), title('y14');

subplot(5,2,6), plot(t, y15), xlim([-2 2]), title('y15');

subplot(5,2,7), plot(t, y16), xlim([-2 2]), title('y16');

subplot(5,2,8), plot(t, y17), xlim([-2 2]), title('y17');

subplot(5,2,9), plot(t, y18), xlim([-2 2]), title('y18');

subplot(5,2,10), plot(t, y19), xlim([-2 2]), title('y19');
```

Explanation:

I've used 5x2 subplots and plotted each of the sinusoid, given titles and decided x limits. I've again noticed that when a sinusoid is multiplied with t (formula of y14), it's approximately symmetrical(but not exactly) with respect to x=0 (as in question 1), because randn generates positive and negative values(odd x odd = even). Adding z value to the sinusoid or adding it inside of the sinusoid doesn't change the shape of the wave much(y12 and y15). Since random numbers change at every execution, the figure changes at every run, but these properties remain same.

Figure:



Problem 3

Code & Comments:

```
z = rand(1, 401);    % generates uniformly distributed number between 0 and 1

z = 0.1.*z;              % numbers are multiplied by 0.1

t = (-2:0.01:2);        % t vector is given

y20 = z;                    % y20, y21,….,y29 are given

y21 = z+t;

y22 = z+y1;

y23 = z.*y1;

y24 = sin(2*pi.*z).*t;

y25 = sin(2*pi.*(t+z));

y26 = z.*y2;

y27 = sin(2*pi.*(t+10.*z));

y28 = y1./z;

y29 = y21+y22+y23+y24+y25+y26+y27+y28;
```

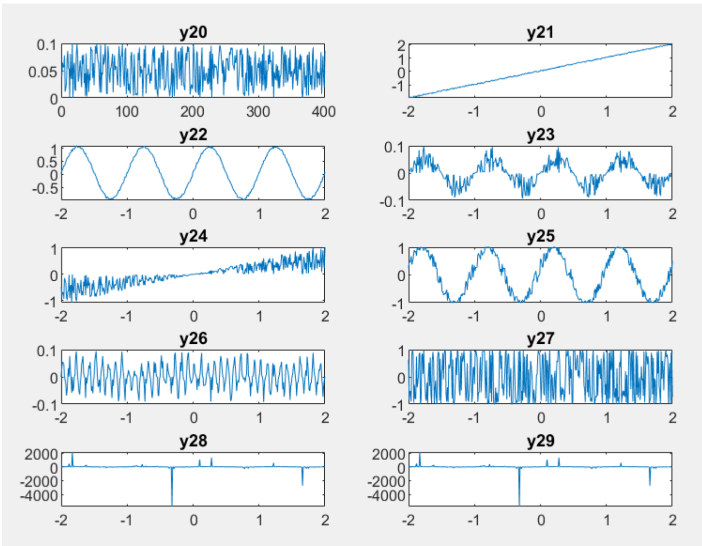
% For each sinusoid, there should be a subplot (5x2 subplots), and its names are written as their titles. I've plotted as plot(t,y) , because they are dependent on t or y values in previous questions, which are also dependent on t. So x coordinate corresponds to t values, y coordinate corresponds to its yn value. I put x limits as [-2,2], since limits of t vector is also [-2,2]. (as in question 1) However, y20=z is independent of t, so I've plotted it as plot(y10), x coordinate corresponds to number of random values(401). So xlim is [0,401]. (as in question 2)

```
subplot(5,2,1), plot(y20), xlim([0 401]), title('y20');

subplot(5,2,2), plot(t, y21), xlim([-2 2]), title('y21');

subplot(5,2,3), plot(t, y22), xlim([-2 2]), title('y22');

subplot(5,2,4), plot(t, y23), xlim([-2 2]), title('y23');

subplot(5,2,5), plot(t, y24), xlim([-2 2]), title('y24');

subplot(5,2,6), plot(t, y25), xlim([-2 2]), title('y25');

subplot(5,2,7), plot(t, y26), xlim([-2 2]), title('y26');

subplot(5,2,8), plot(t, y27), xlim([-2 2]), title('y27');

subplot(5,2,9), plot(t, y28), xlim([-2 2]), title('y28');

subplot(5,2,10), plot(t, y29), xlim([-2 2]), title('y29');
```

Explanation:

I've used 5x2 subplots and plotted each of the sinusoid, given titles and decided x limits. I've again noticed that adding z value to the sinusoid or adding it inside of the sinusoid doesn't change the shape of the wave much(y22 and y25). Multiplying the sinusoid values by the values between 0 and 1 also doesn't change the shape of wave much(y23). However, y24(sinusoid contains z multiplied with t) is not symmetrical with respect to x=0, because rand function generates only positive values. It becomes an approximately odd function(f(x)=a and f(-x)=-a). Since random numbers change at every execution, the figure changes at every run, but these properties remain same.

Figure:



Problem 4

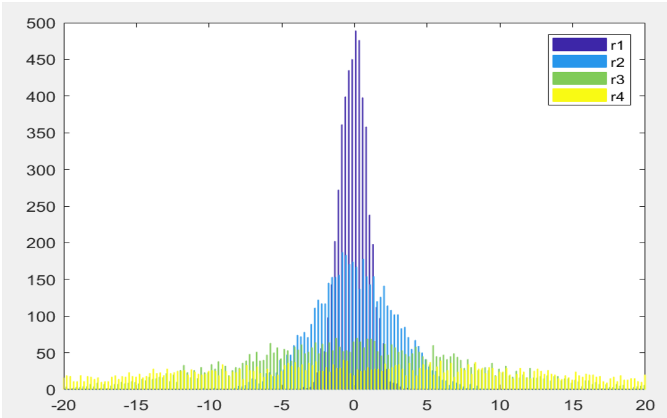
Code & Comments:

```
std = [1 sqrt(8) 8 16];        % standard deviation vector, standard deviation is the square root of the variance

H=normrnd(0,repmat(std,5000,1));   % sample size is 5000

hist(H,500), xlim([-20.0 20.0]), legend(["r1"; "r2"; "r3"; "r4"]);   % 500 is the number of bins, I put the x limit just because distribution looks better and more symmetrical
```

Explanation:

In order not to draw distributions of variables one by one, I've decided to put the values in matrix and plot them all with hist function. I've learned how to use normrnd, repmat and legend functions. Each column of H corresponds to a distribution with a spesific mean(all 0) and variance values that are given in the question. The size of H matrix is 5000x4. I've also noticed that as sample size grows, the closer the mean will get to 0. It's also very clear on the figure that as variance grows, the numbers are much more spread out from the mean.

Figure:



Problem 5

Code & Comments:

std = [1 2 1 2];   % standard deviation vector, standard deviation is the square root of the variance

me = [10 20 -10 -20];   % mean vector

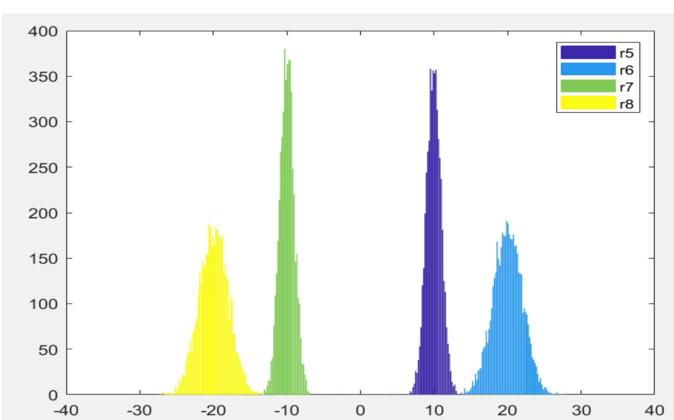
H = normrnd(repmat(me, 5000,1), repmat(std, 5000,1));   % sample size is 5000

hist(H,300), xlim([-40.0 40.0]), legend(["r5"; "r6"; "r7"; "r8"]);   % 300 is the number of bins, I put the x limit just because distribution looks better and more symmetrical

Explanation:

In order not to draw distributions of variables one by one, I've decided to put the values in matrix and plot them all with hist function. (as in question 4) Each column of H corresponds to a distribution with a spesific mean and variance values that are given in the question. The size of H matrix is 5000x4. Different mean values can be seen clearly on the figure. It's also very clear on the figure that as variance grows, the numbers are much more spread out from the mean.

Figure:



Problem 6

Code & Comments:

r11 = 8.*rand(5000,1)-4; % 5000 random numbers between -4 and 4

r21 = 40.*rand(5000,1)-20; % 5000 random numbers between -20 and 20

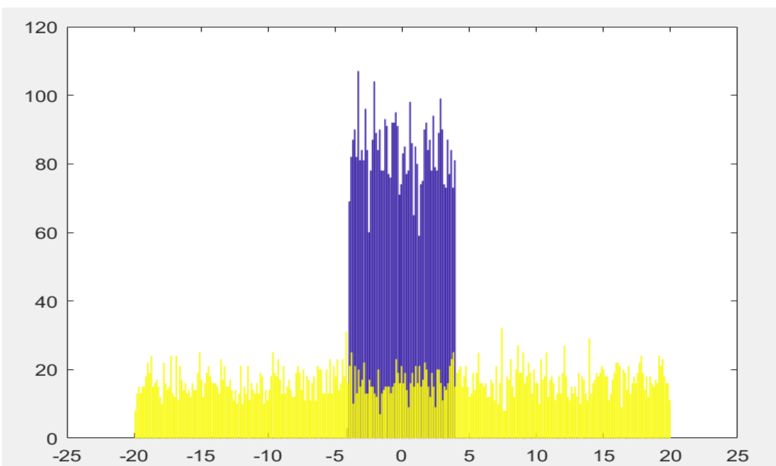
H = [r11 r21]; % putting two vectors in a matrix to show it on same figure

hist(H, 300), xlim([-25.0 25.0]); % 300 is the number of bins, I put the x limit just because distribution looks better and more symmetrical

Explanation:

I've learned the formula of creating random numbers in a given range in matlab. I've created the figure with the same way as in previous questions. I've put them in a matrix and used hist function.

Figure:



Problem 7

Code & Comments:

% After loading mysignal.mat; fs, t and x are loaded in workspace

l = length(x); % number of samples

ff = fft(x,l); % computes DFT of x using fast fourier transform

x0 = fftshift(abs(ff)); % shift values

f0 = (-l/2:l/2-1)*(fs/l); % 0-centered frequency range

plot(f0,x0); % frequency values (1), which are 25, 65 and 120

x_phase = angle(x0);

```
plot(f0, x_phase/pi);       % phase values (2), which are all 0
plot(abs(ff)/(l/2));        % amplitude values (3), which are all 1.7
```

Explanation:

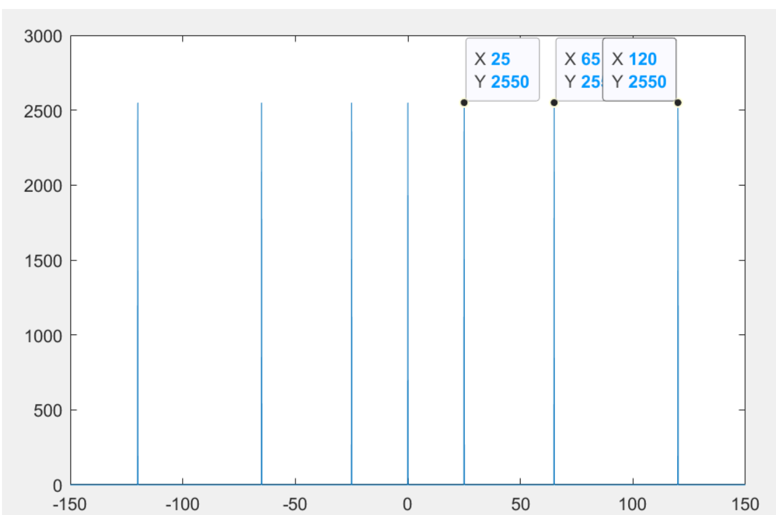
I've learned how to use fft function to extract different frequency values. At first, we need to do the transform with the help of this function. After that, to get rid of complex numbers, we need to get its absolute value form. To shift the values to a 0-centered range, we need to use fftshift function and create a frequency range for it. Finally we plot the frequency range and x values(figure 1), and we see different values of frequencies which are 25, 65 and 120. With the help of angle function, we get the phase of each frequency value(in figure 2), which are all 0. From first figure, we can see that all 3 amplitudes are equal to each other. However, to find its value, we need to plot absolute value of ff function divided by l/2. (If we divided to l, we would get the half of the amplitude. So, finally we can write the formula.

Formula:

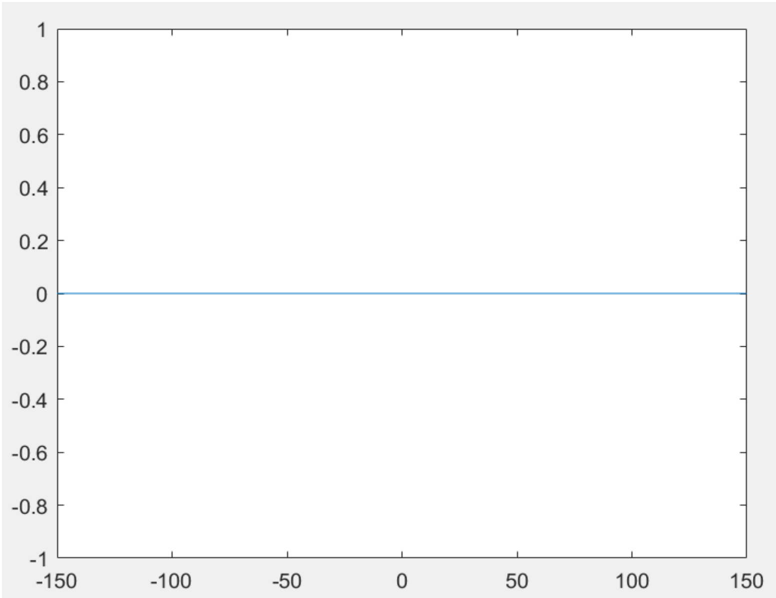
x = 1.7*cos(2*pi*25*t) + 1.7*cos(2*pi*65*t) + 1.7*cos(2*pi*120*t);

Figures:

(1)

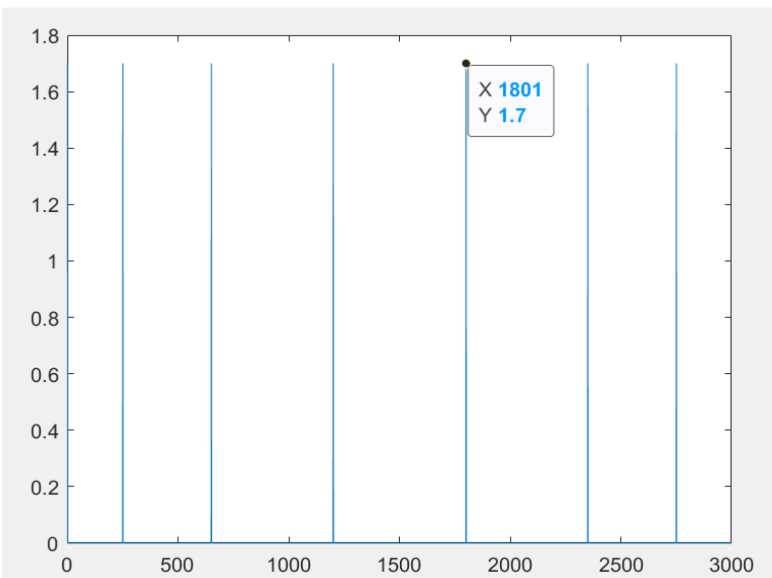


(2)



1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-150
-100
-50
0
50
100
150

(3)



X 1801
Y 1.7
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
500
1000
1500
2000
2500
3000

Problem 8

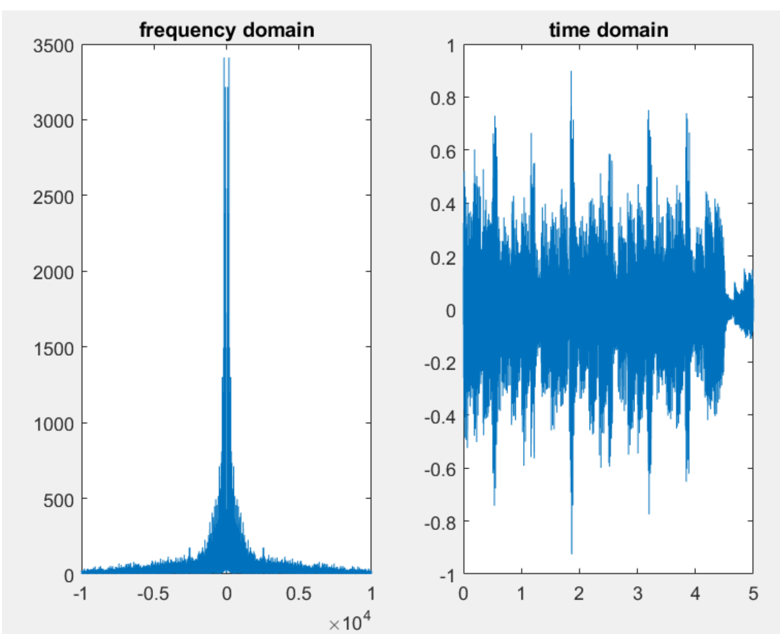
Code & Comments:

```
[y,fs] = audioread('C:\Users\Lenovo\Desktop\den.wav');            % returns sample data and sample rate for it

y1 = y(:,1);

y2 = y(:,2);

l=length(y1);                  % number of samples

Y = fft(y1, l);                %  computes DFT of x using fast fourier transform

Y0 = fftshift(abs(Y));         % shift y values

f0 = (-l/2:l/2-1)*(fs/l);      % 0-centered frequency range

size = l/fs;                   % time span signal(approximately 5 second)

t = 0:1/fs:size-1/fs;          % 5 second span time vector

subplot(1, 2, 1), plot(f0, Y0), xlim([-10000,10000]), title('frequency domain');            % frequency domain

subplot(1, 2, 2), plot(t, y1), xlim([0 5]),  title('time domain');                          % time domain
```

Explanation:

At first, we need to read the wav file with the help of audioread function. Then, we do the transform with the help of fft function. After that, to get rid of complex numbers, we need to get its absolute value form. To shift the values to a 0-centered range, we need to use fftshift function and create a frequency range for it. Finally we plot the frequency range and x values, and we see lots of different values of frequencies. To see time domain, we simply call plot(t, y1).

Figure:



Problem 9

Code & Comments:

```
RGB = imread('lena.png');                 % reading the image
I = rgb2gray(RGB);                        % converting the rgb image into a grayscale image
minVal = min(I(:));                       % minimum value of image matrix
[rowMin,colMin] = find(I==minVal);        % location of the minimum value of the matrix
maxVal = max(I(:));                       % maximum value of the matrix
[rowMax,colMax] = find(I==maxVal);        % location of the maximum value of the matrix
meanVal = mean(I(:));                     % mean value of the matrix
stdVal = std(double(I(:)));               % standard deviation value of the matrix
```

Explanation:

I've learned what a grayscale image matrix is and how to use imread and rgd2gray functions. I've learned how to use max, min, mean, find and std functions for a matrix. I've also noticed that std function gives error when it's called as *std(I(:)),* it needs to have a double conversion.

Values:

```
minVal = 25   maxVal = 245   meanVal = 124.0425
rowMin = 72   rowMax = 274   stdVal = 47.8557
colMin = 4    colMax = 396
```