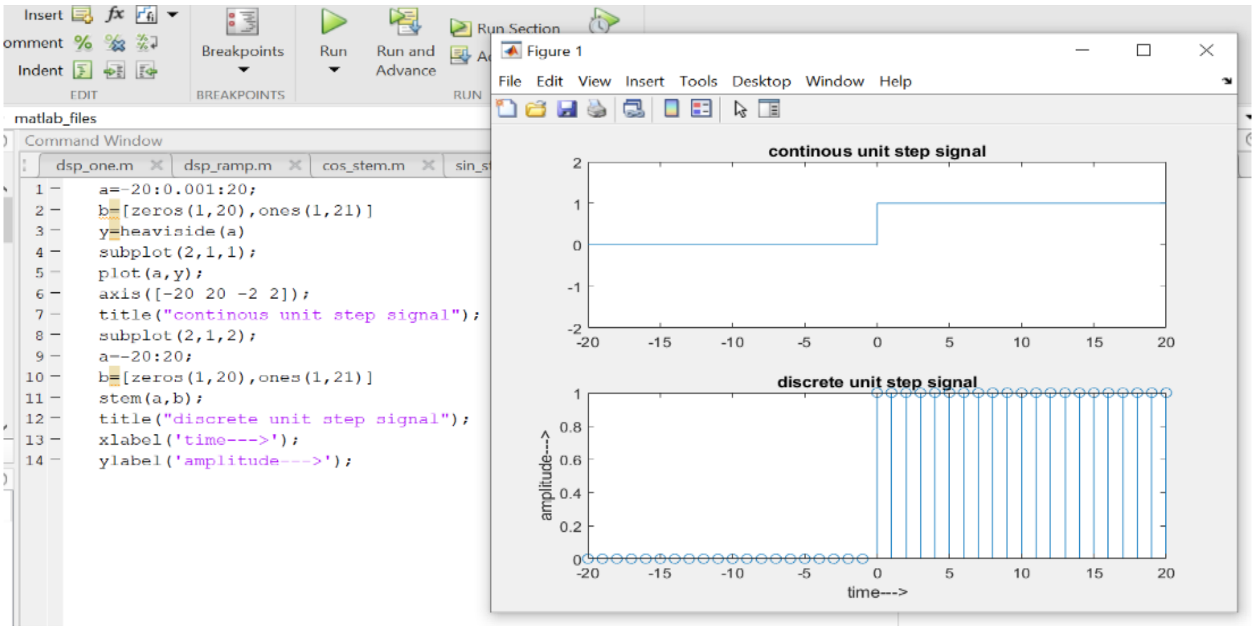
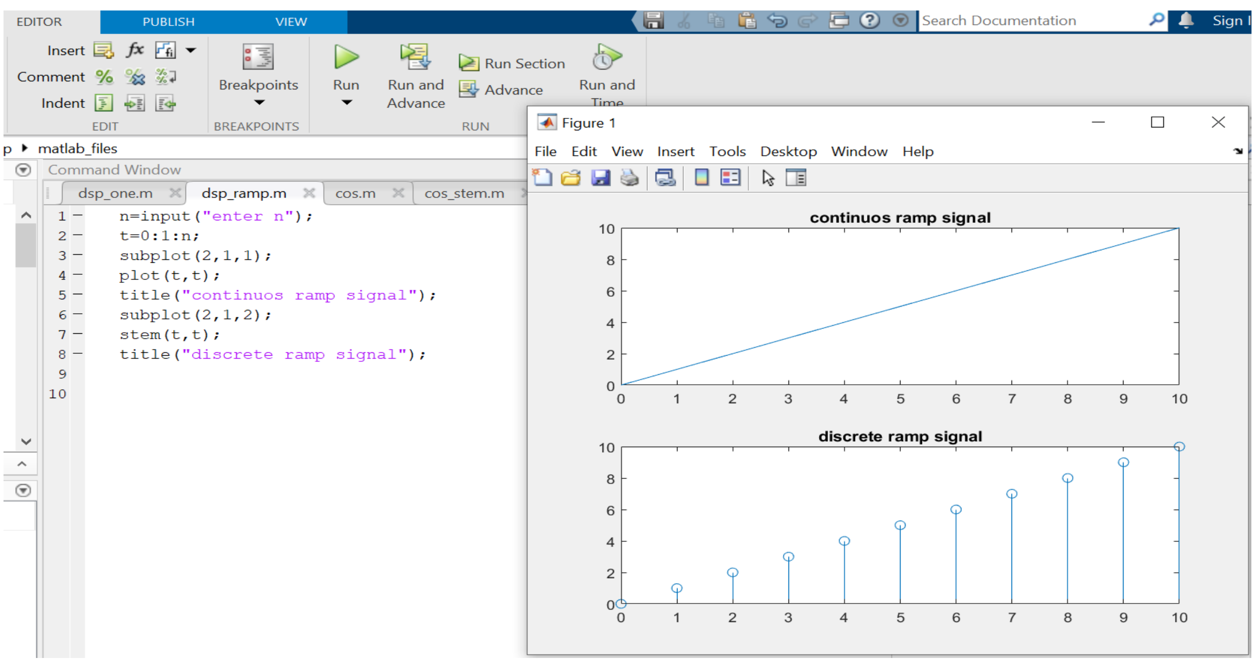
EXPERIMENT 1: Generation and operation of continuous and Discrete Time signals.

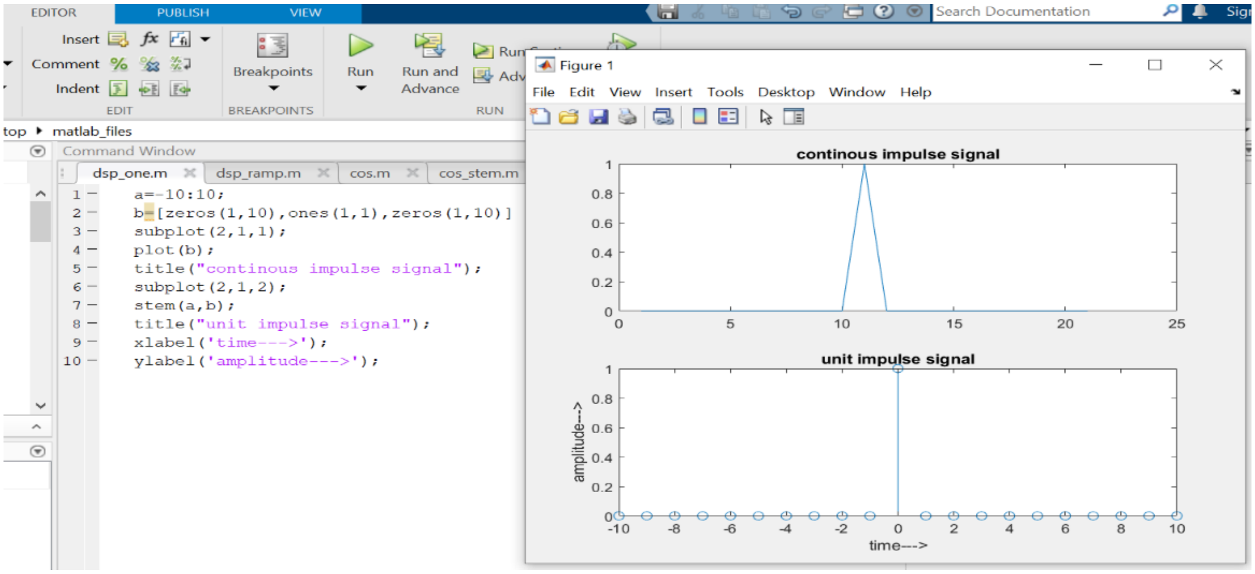
UNIT STEP SIGNAL:



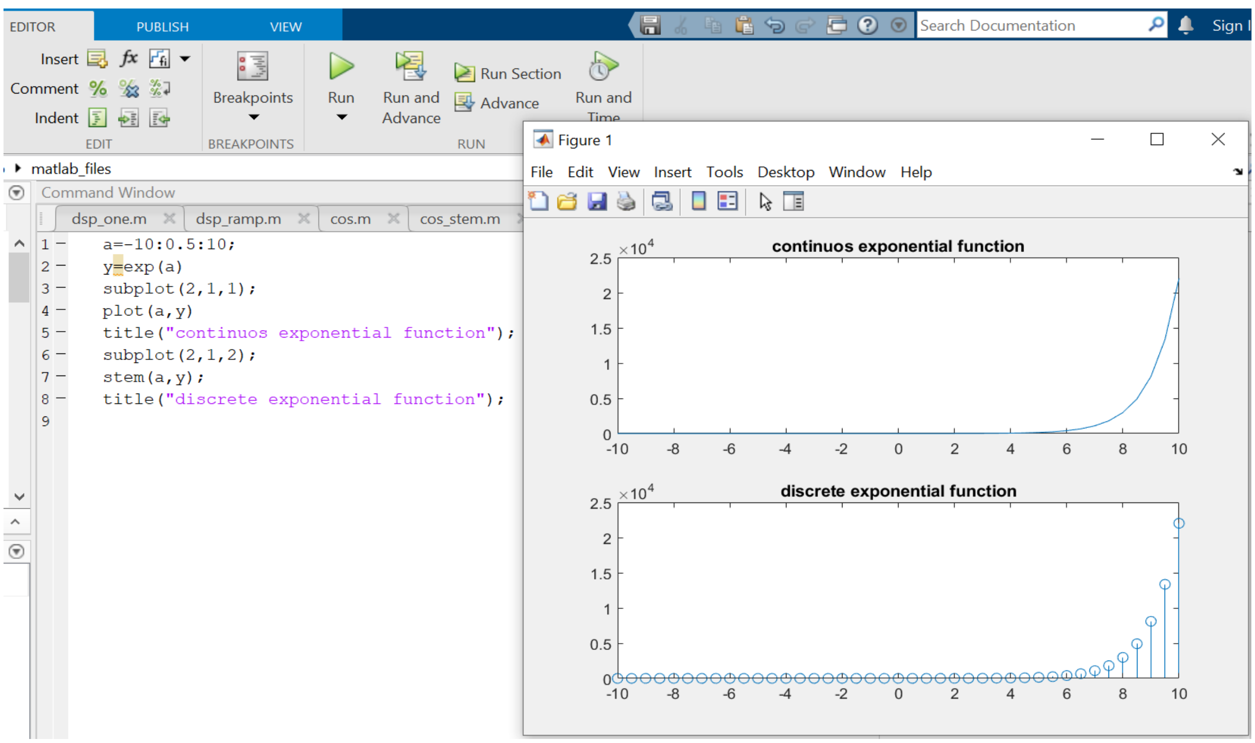
RAMP SIGNAL:



IMPULSE SIGNAL:

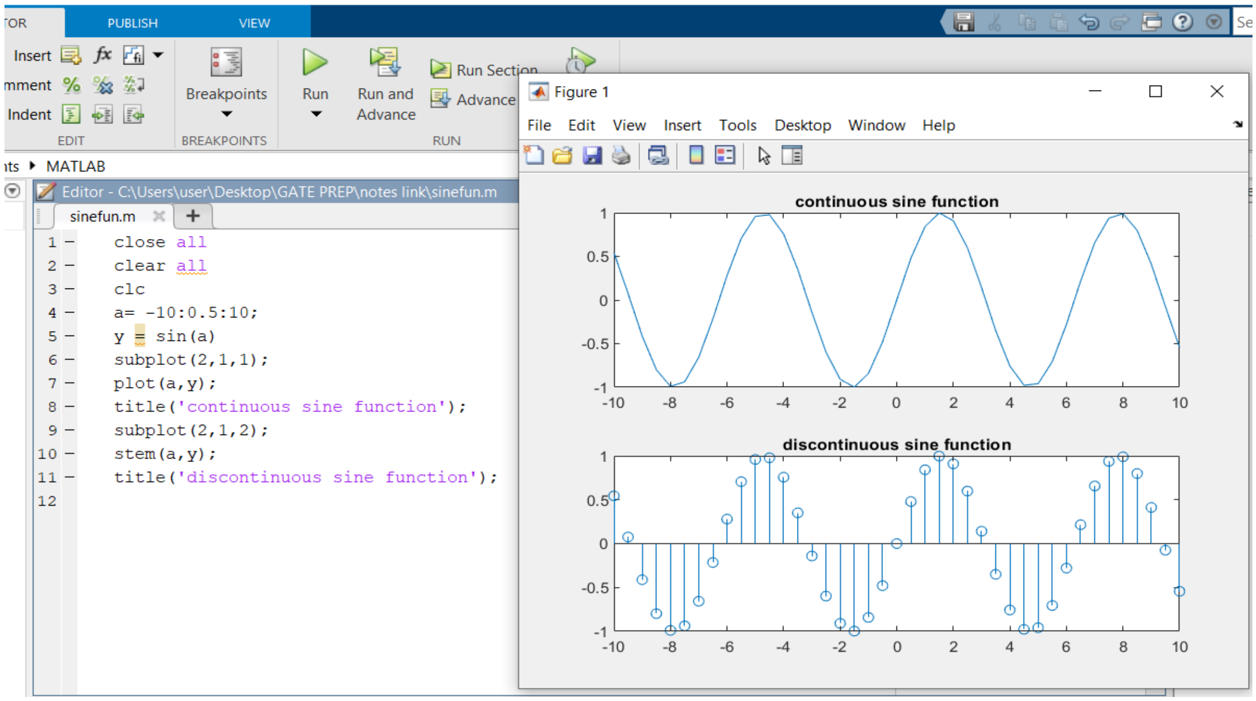


EXPONENTIAL SIGNAL:

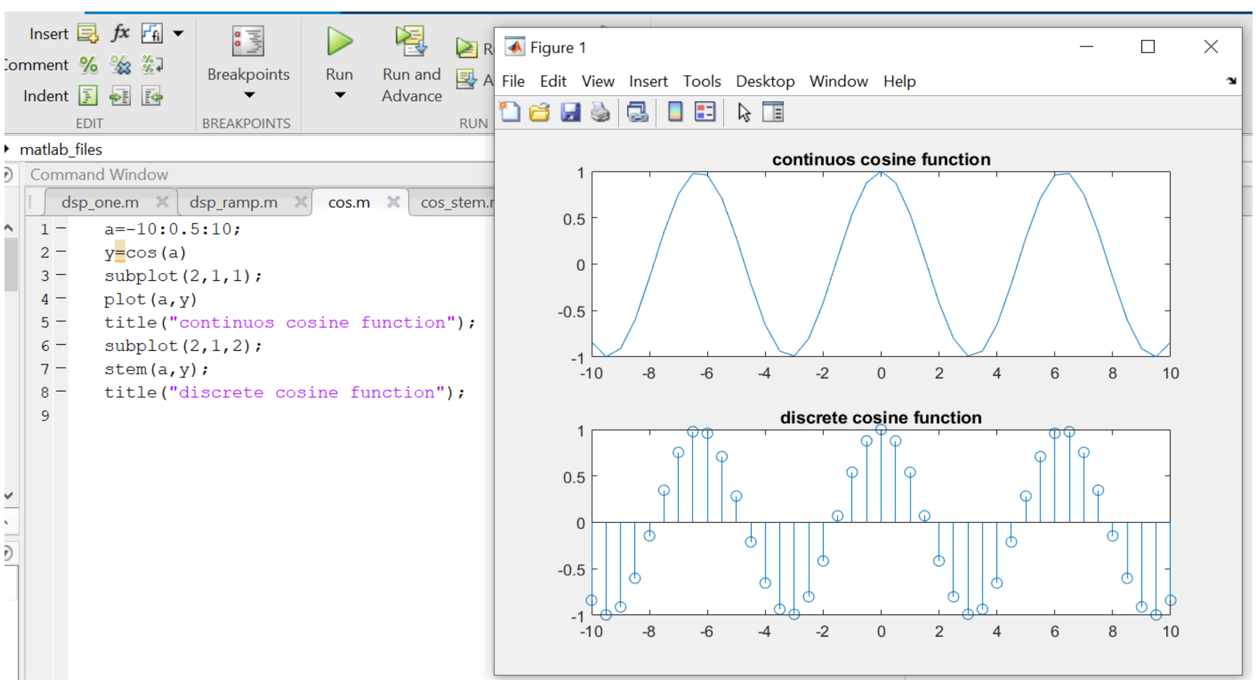


SINUSOIDAL SIGNAL:

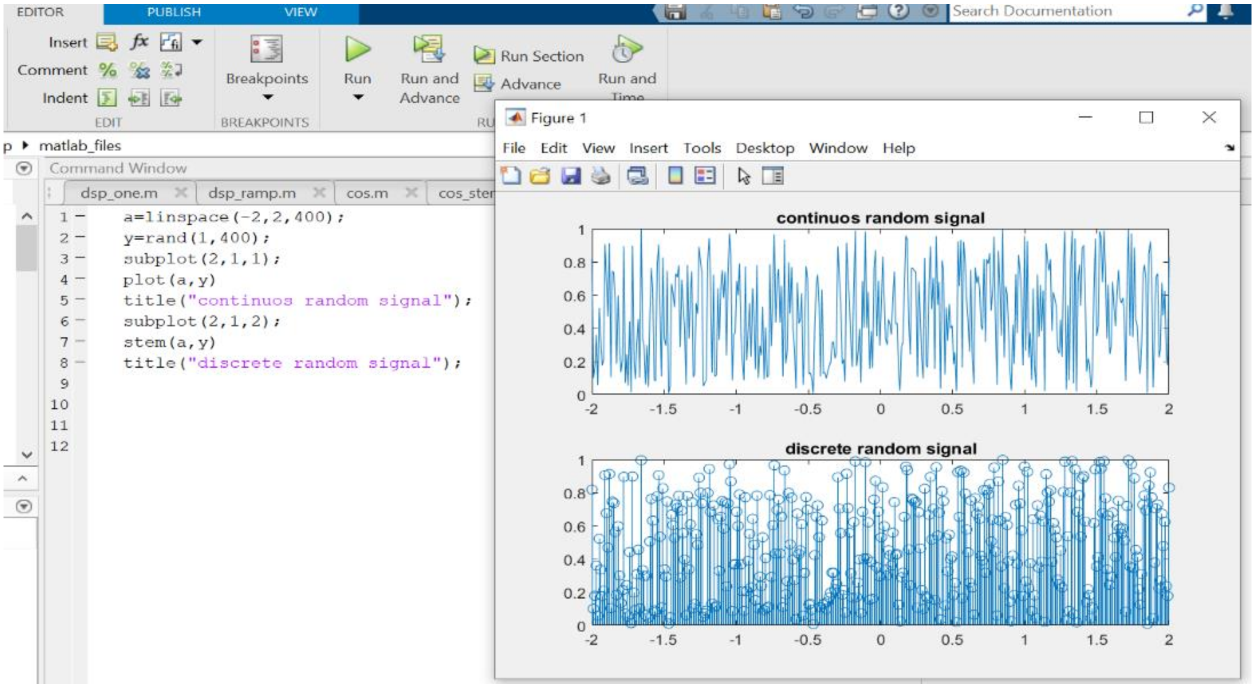
SINE FUNCTION:



COSINE FUNCTION:



RANDOM SIGNAL:



<u>Experiment 2</u>: Operation on sequences like:

a. Signal addition

b. Signal multiplication

c. Scaling

d. Shifting

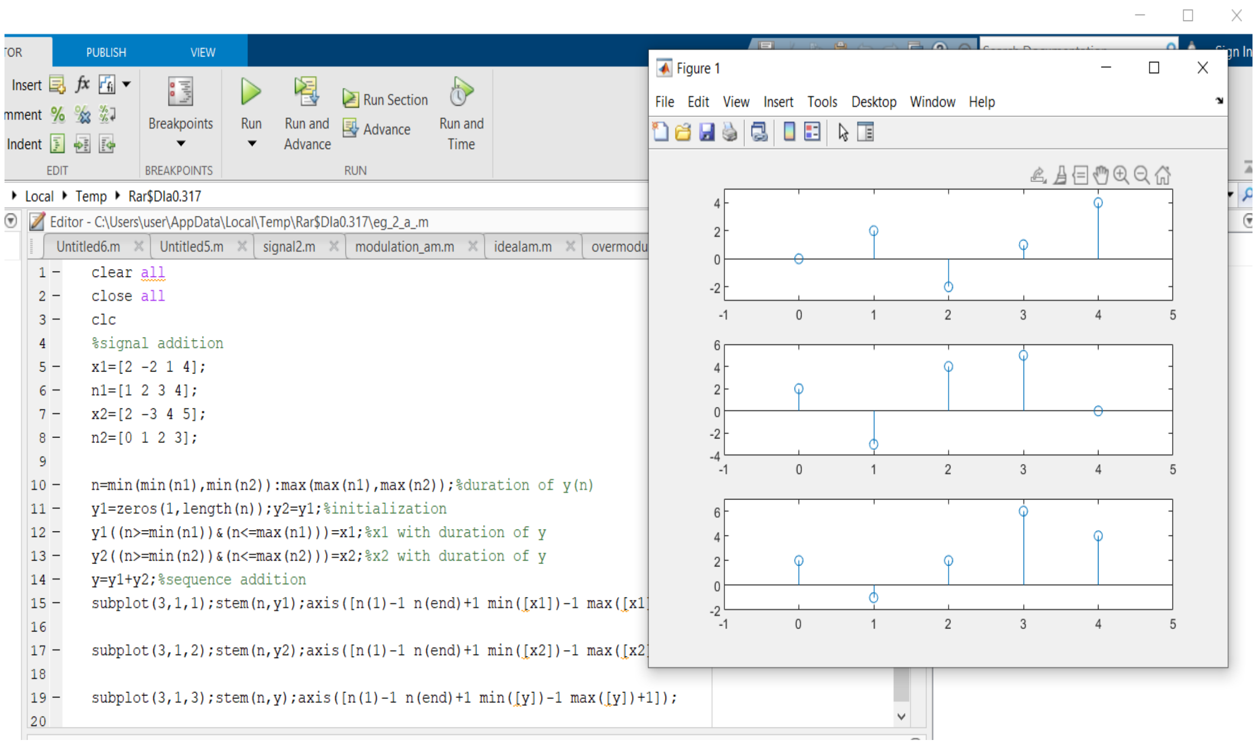
e. Folding

f. Signal Energy

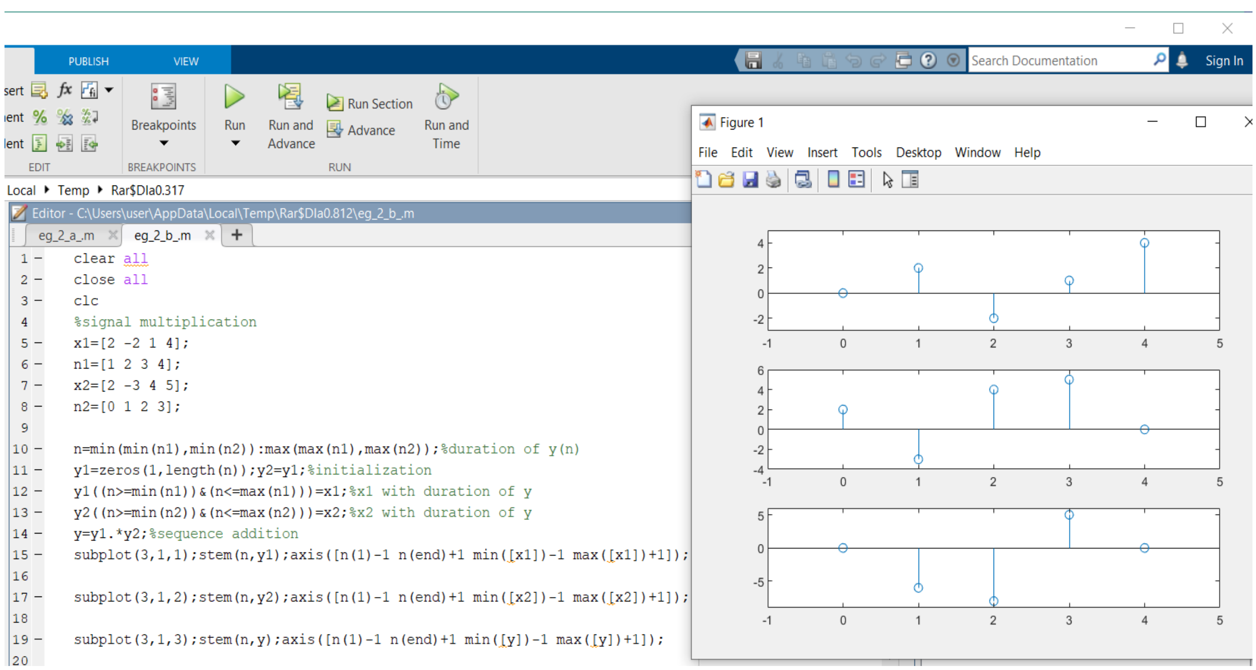
g. Signal Power

h. Even Odd component

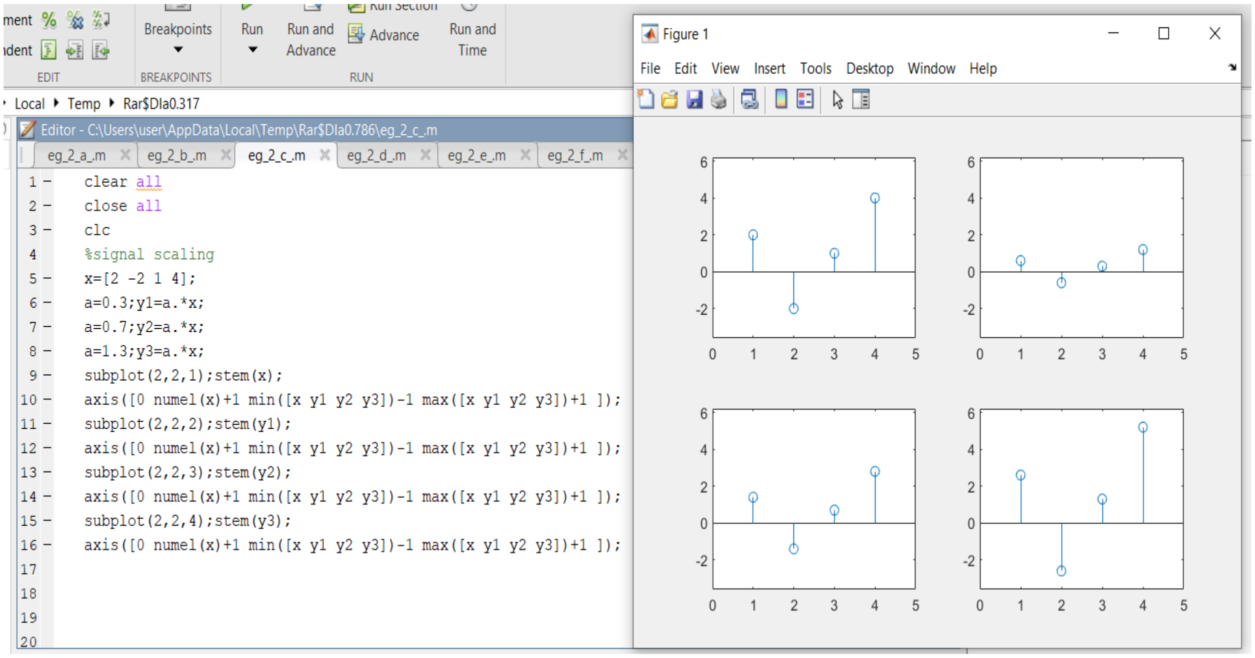
. **ADDITION:**



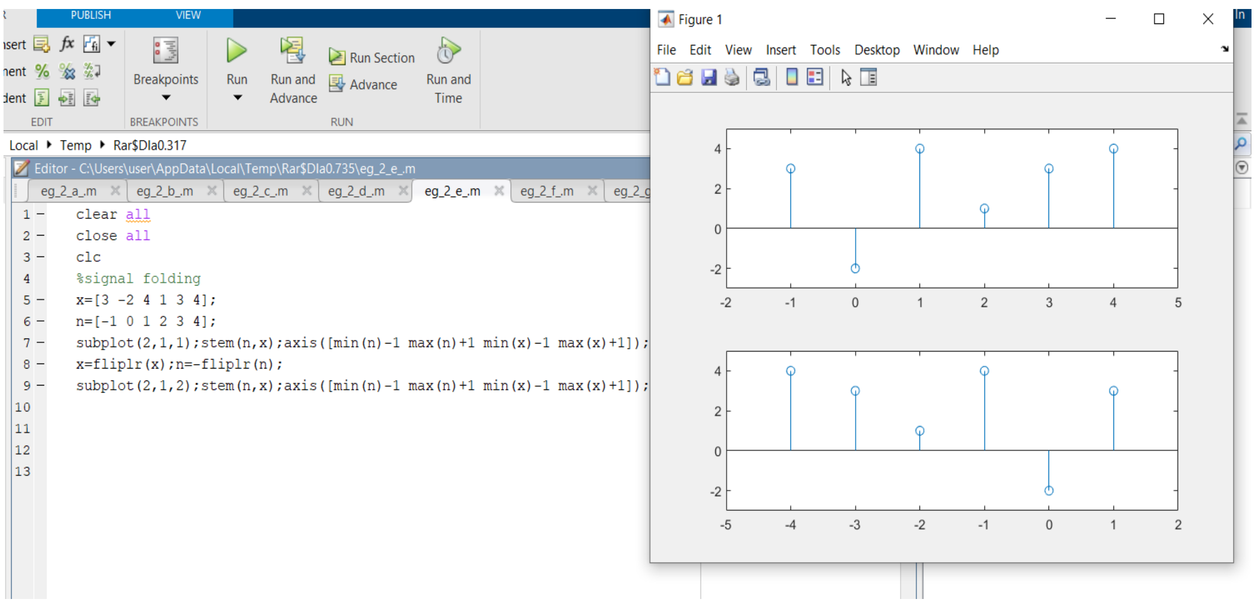
MULTIPLICATION



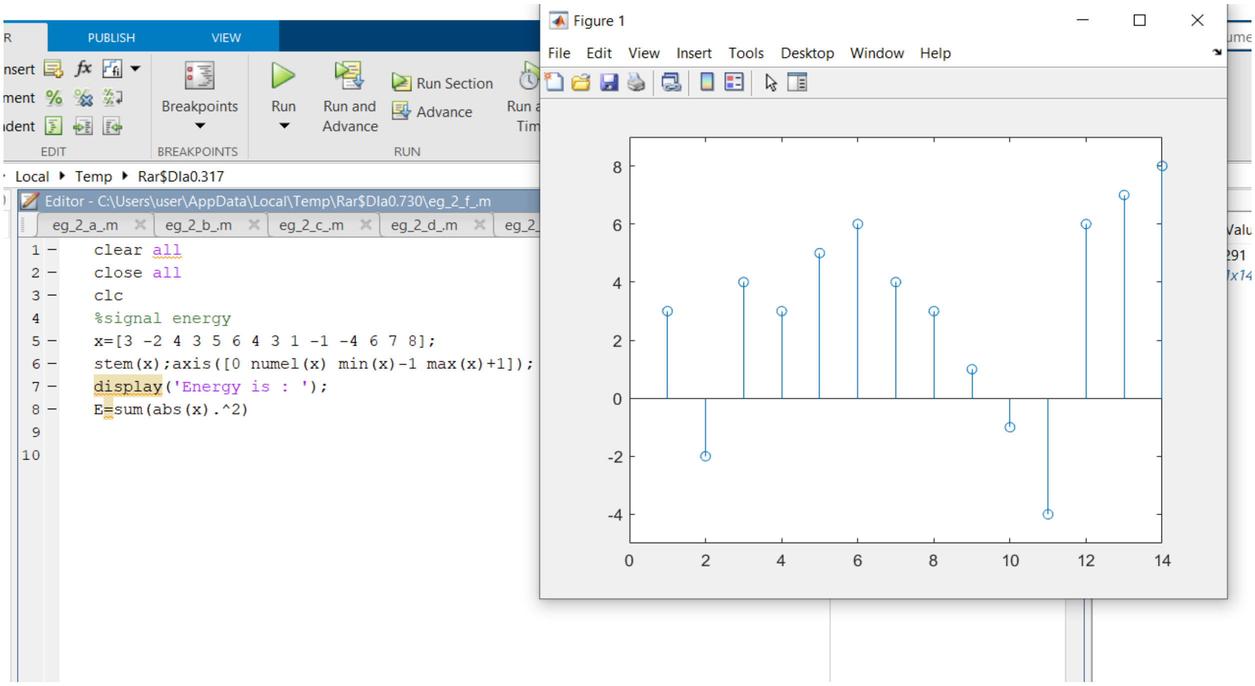
SIGNAL SCALING



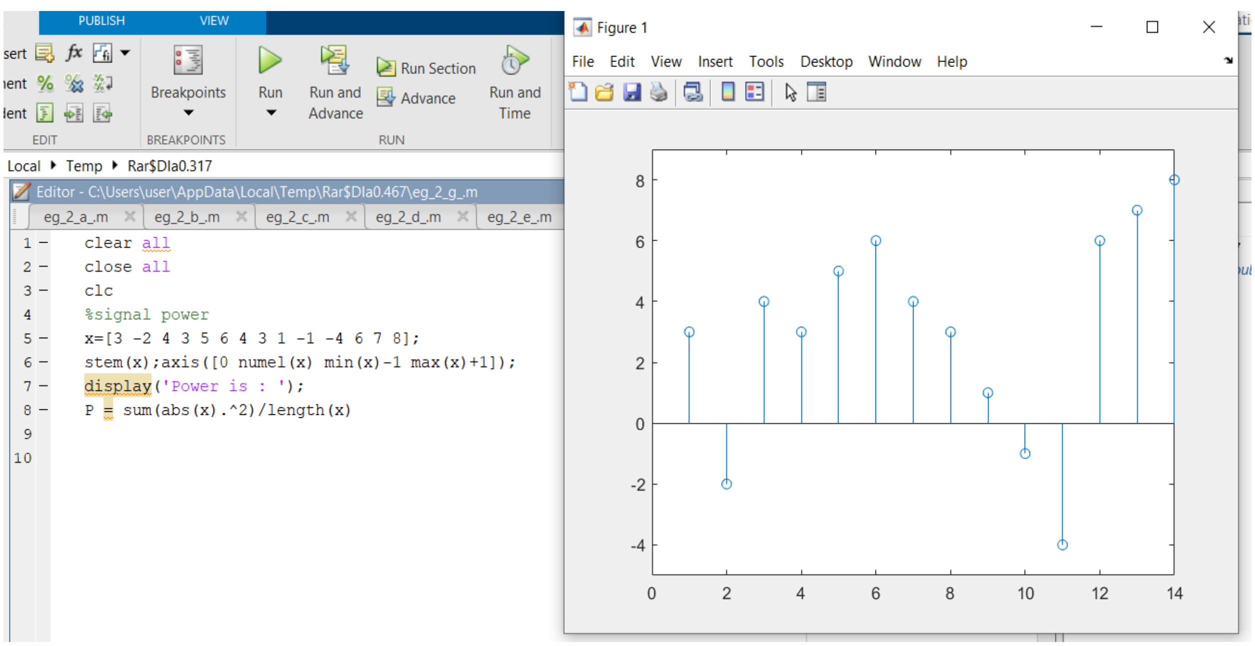
SIGNAL FOLDING



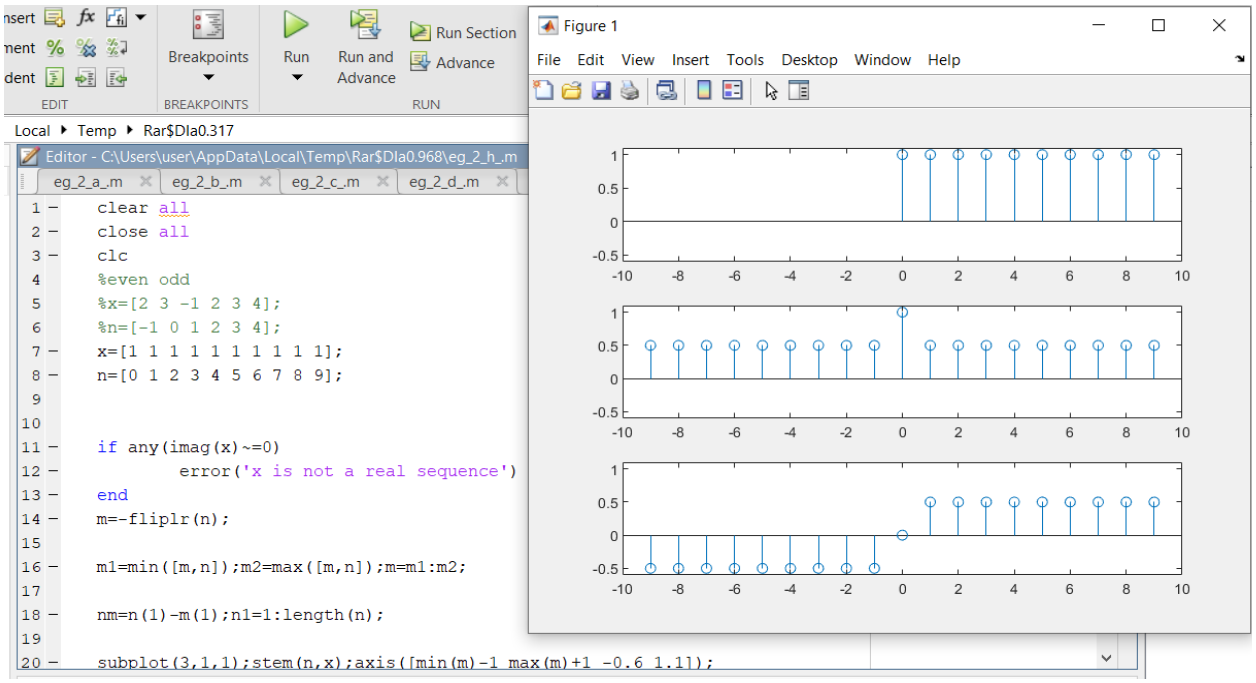
ENERGY:



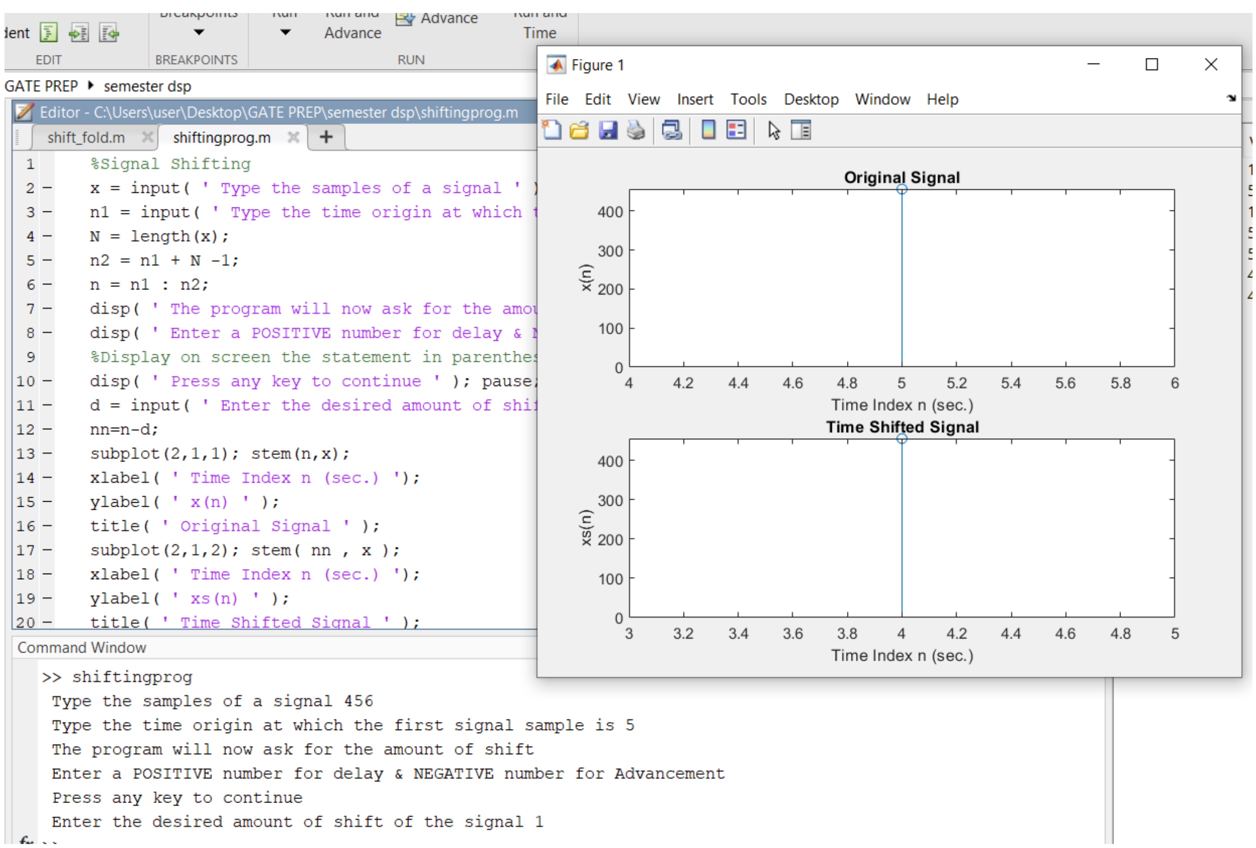
POWER:



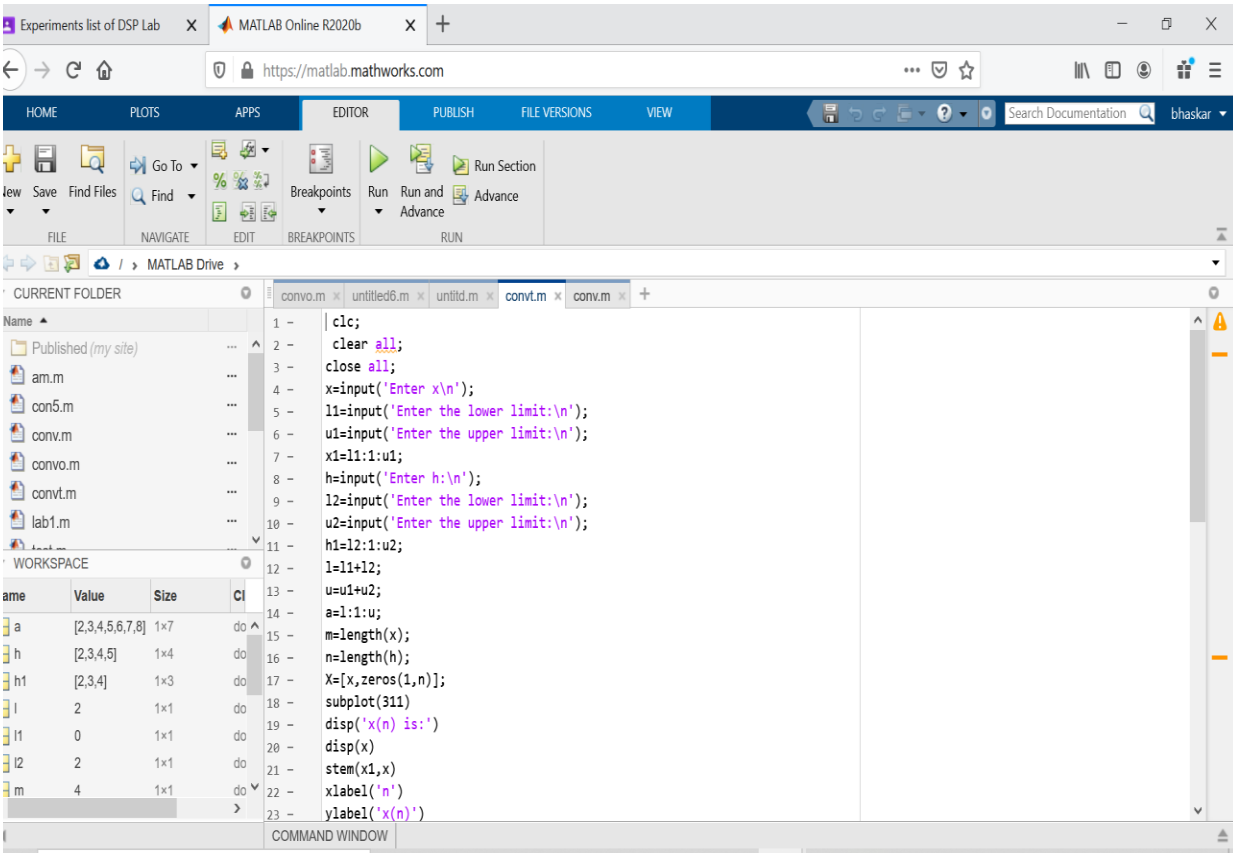
EVEN AND ODD:



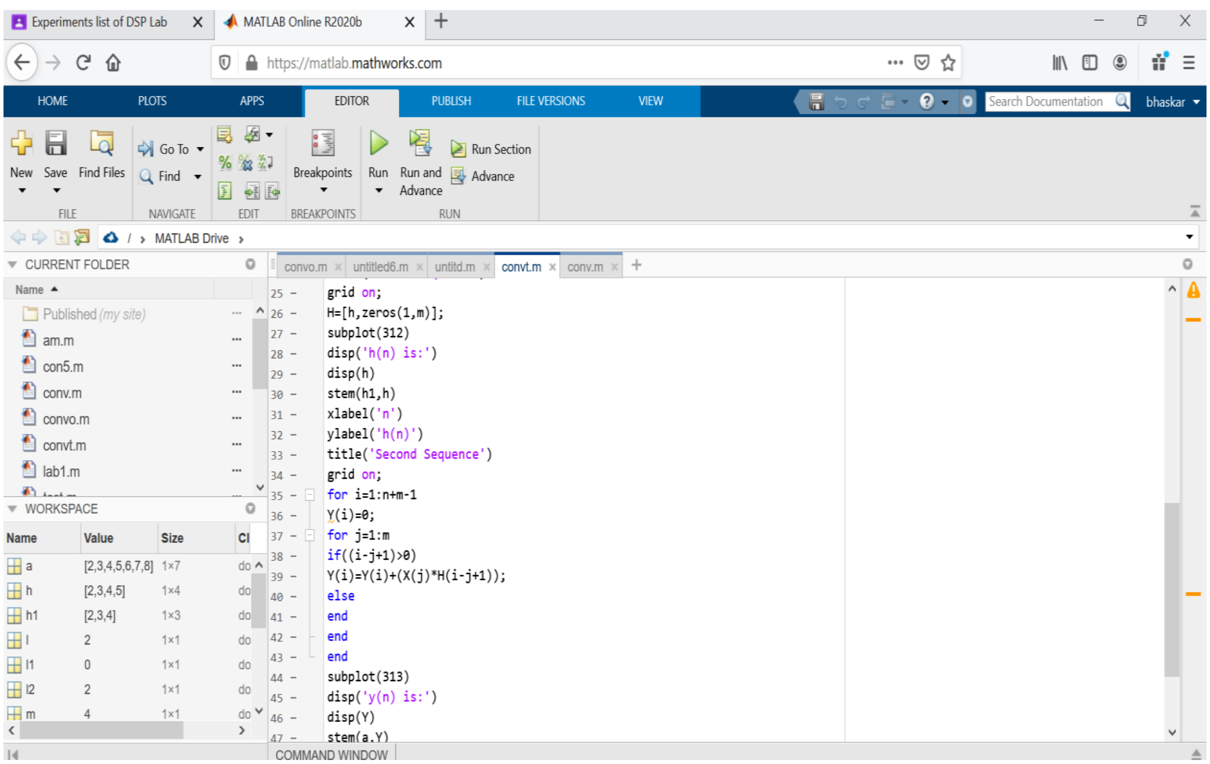
SIGNAL SHIFTING:



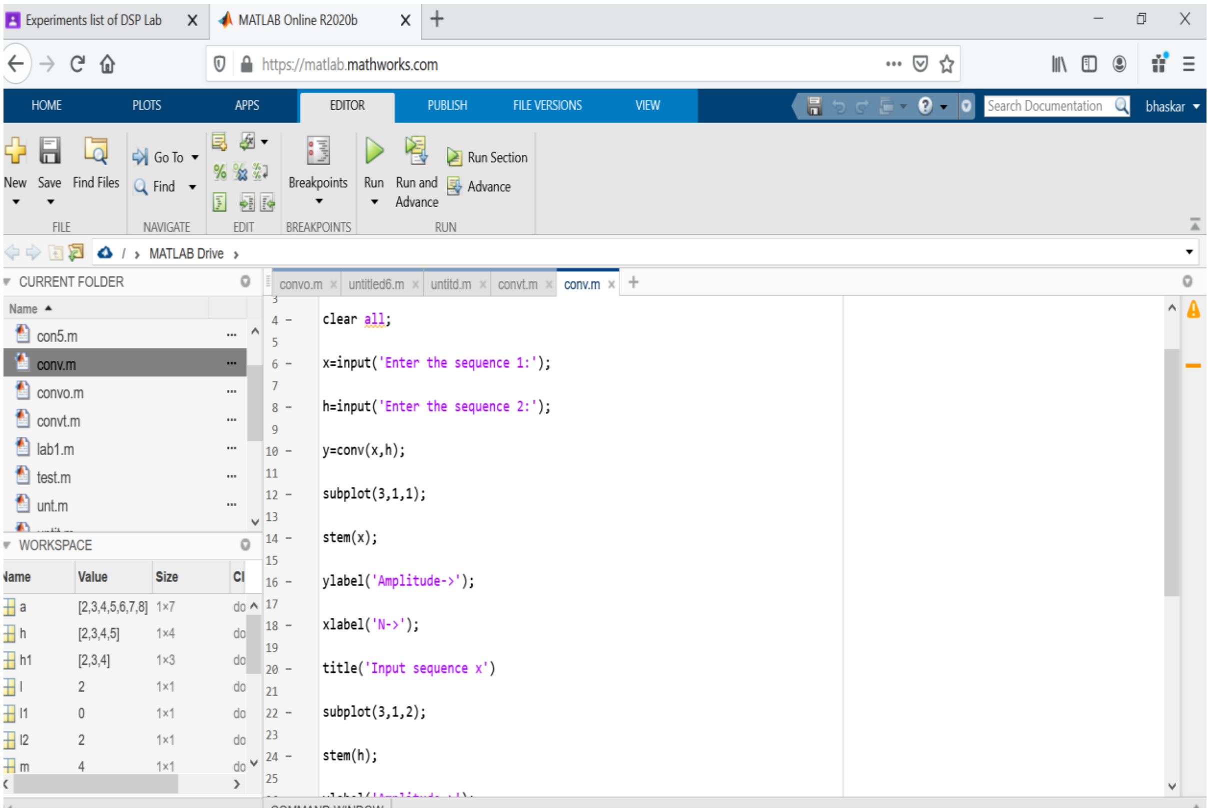
Experiment 3: Computation of linear convolution on discrete signals.

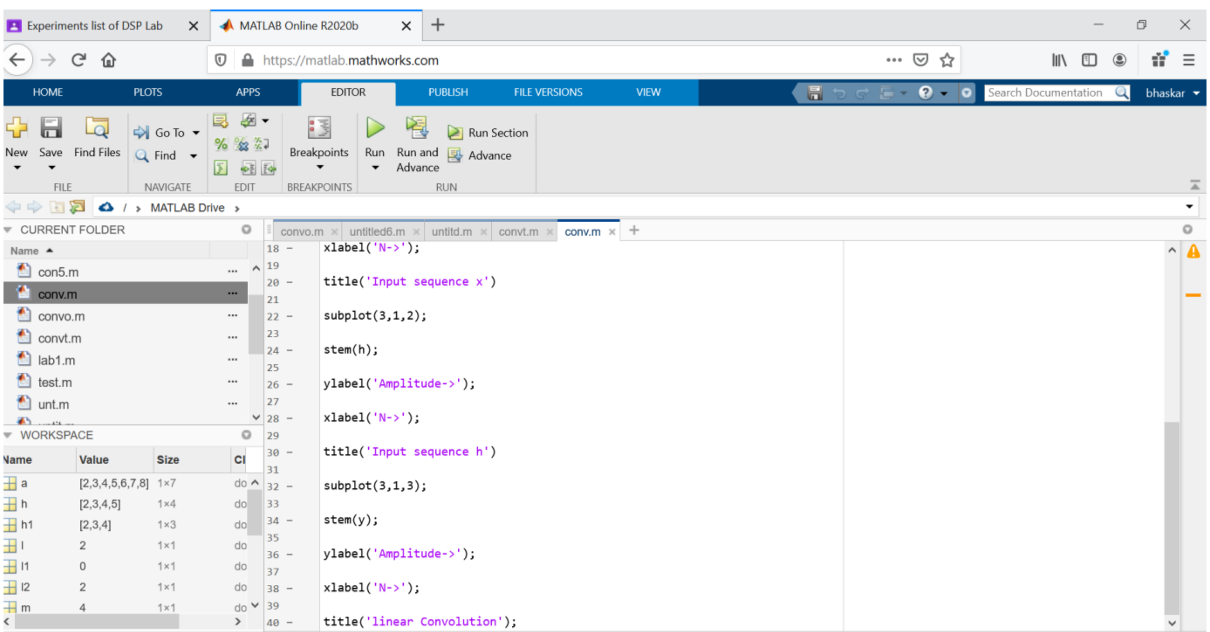


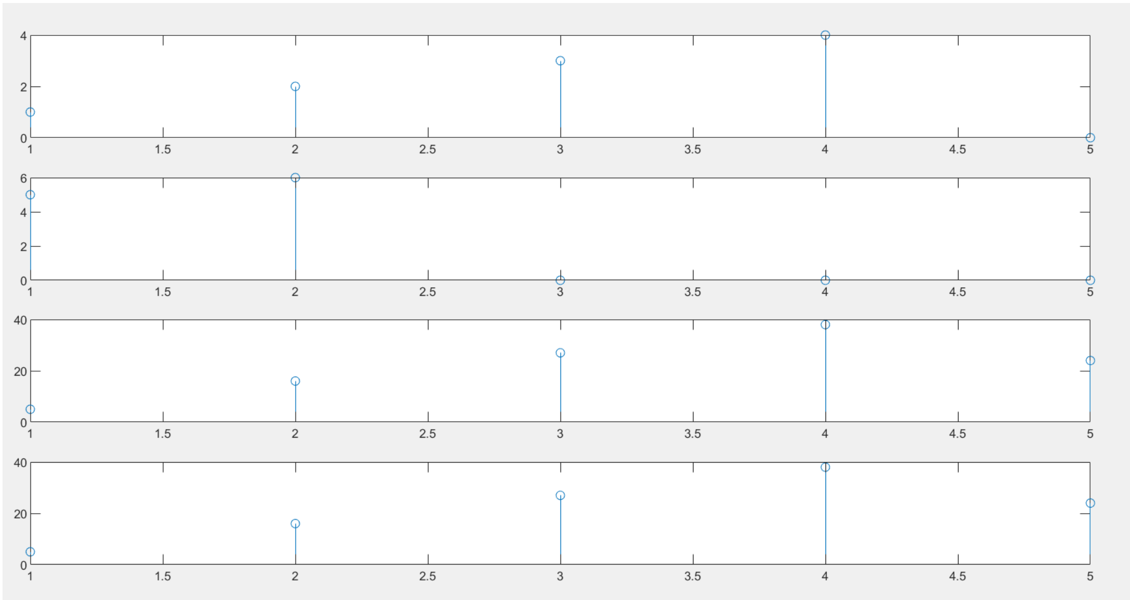
clc;
clear all;
close all;
x=input('Enter x\n');
l1=input('Enter the lower limit:\n');
u1=input('Enter the upper limit:\n');
x1=l1:1:u1;
h=input('Enter h:\n');
l2=input('Enter the lower limit:\n');
u2=input('Enter the upper limit:\n');
h1=l2:1:u2;
l=l1+l2;
u=u1+u2;
a=l:1:u;
m=length(x);
n=length(h);
X=[x,zeros(1,n)];
subplot(311)
disp('x(n) is:')
disp(x)
stem(x1,x)
xlabel('n')
ylabel('x(n)')



grid on;
H=[h,zeros(1,m)];
subplot(312)
disp('h(n) is:')
disp(h)
stem(h1,h)
xlabel('n')
ylabel('h(n)')
title('Second Sequence')
grid on;
for i=1:n+m-1
Y(i)=0;
for j=1:m
if((i-j+1)>0)
Y(i)=Y(i)+(X(j)*H(i-j+1));
else
end
end
end
subplot(313)
disp('y(n) is:')
disp(Y)
stem(a,Y)


Experiments list of DSP Lab
MATLAB Online R2020b
https://matlab.mathworks.com
HOME PLOTS APPS EDITOR PUBLISH FILE VERSIONS VIEW
MATLAB Drive
CURRENT FOLDER
con5.m
conv.m
convo.m
convt.m
lab1.m
test.m
unt.m
WORKSPACE
convo.m untitled6.m untitd.m convt.m conv.m
clear all;
x=input('Enter the sequence 1:');
h=input('Enter the sequence 2:');
y=conv(x,h);
subplot(3,1,1);
stem(x);
ylabel('Amplitude->');
xlabel('N->');
title('Input sequence x')
subplot(3,1,2);
stem(h);

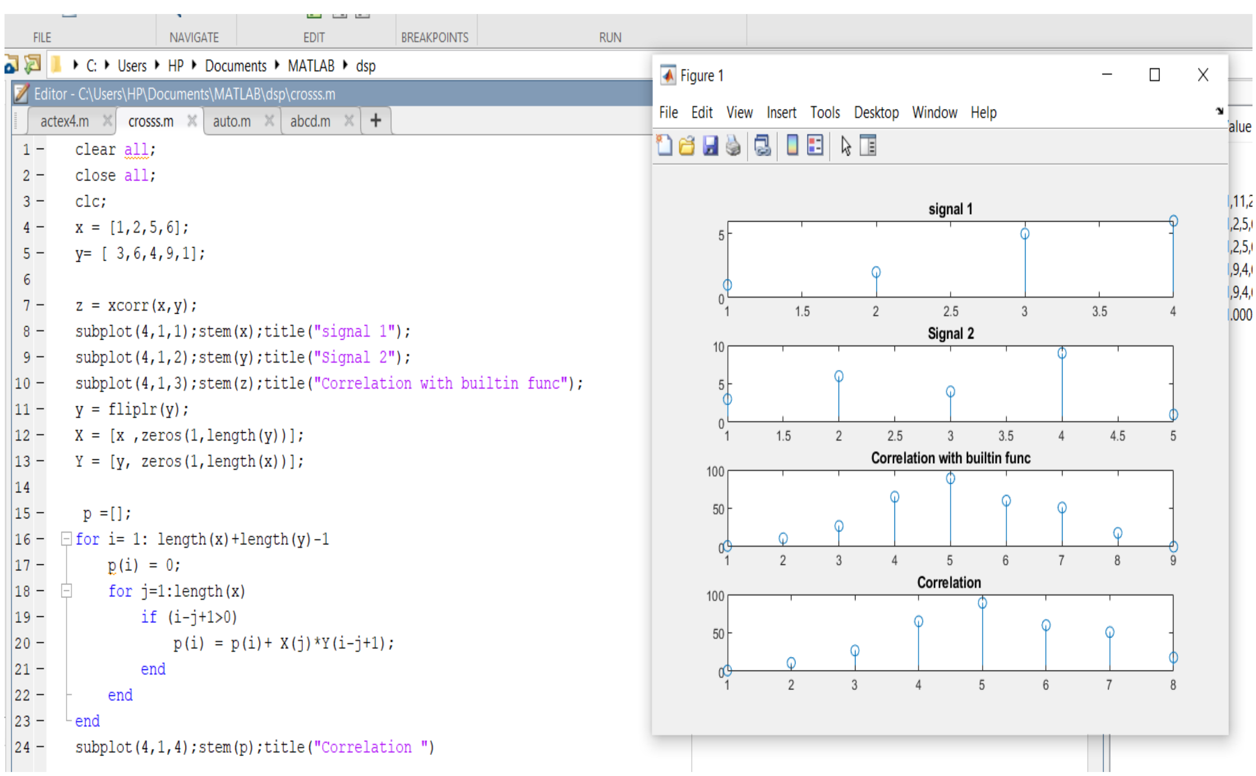

Experiments list of DSP Lab
MATLAB Online R2020b
https://matlab.mathworks.com
HOME PLOTS APPS EDITOR PUBLISH FILE VERSIONS VIEW
MATLAB Drive
convo.m untitled6.m untitd.m convt.m conv.m
xlabel('N->');
title('Input sequence x')
subplot(3,1,2);
stem(h);
ylabel('Amplitude->');
xlabel('N->');
title('Input sequence h')
subplot(3,1,3);
stem(y);
ylabel('Amplitude->');
xlabel('N->');
title('linear Convolution');



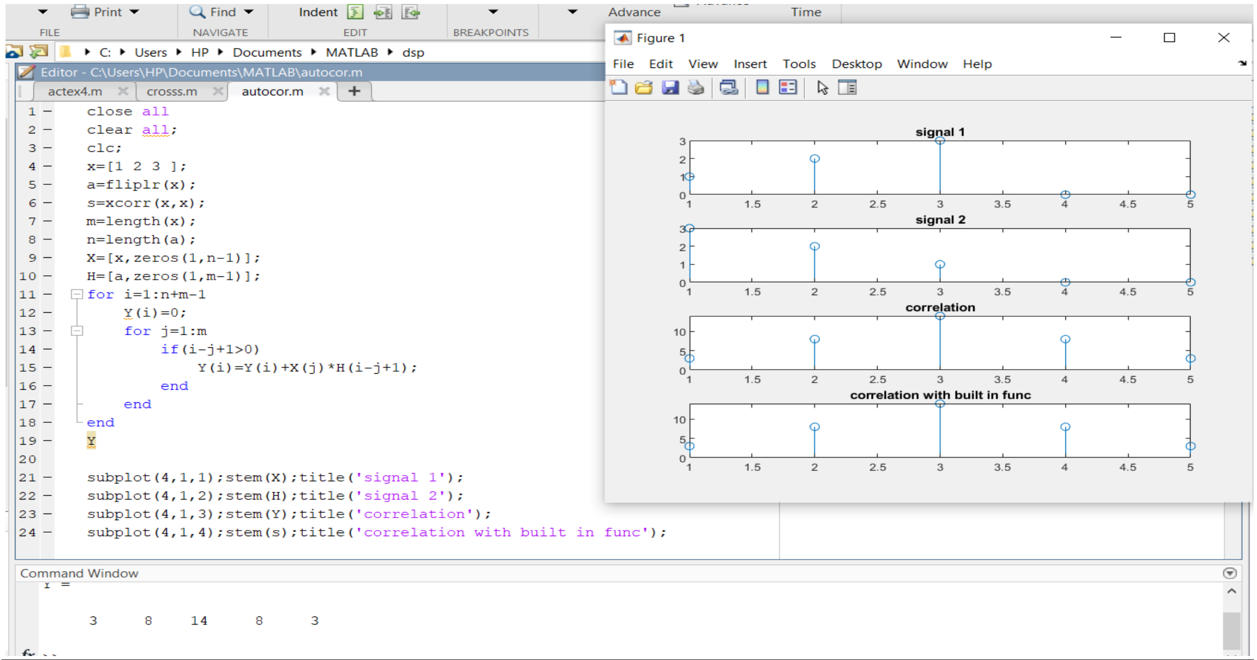
EXPERIMENT 4:

Computation of cross and auto correlation

Cross correlation:



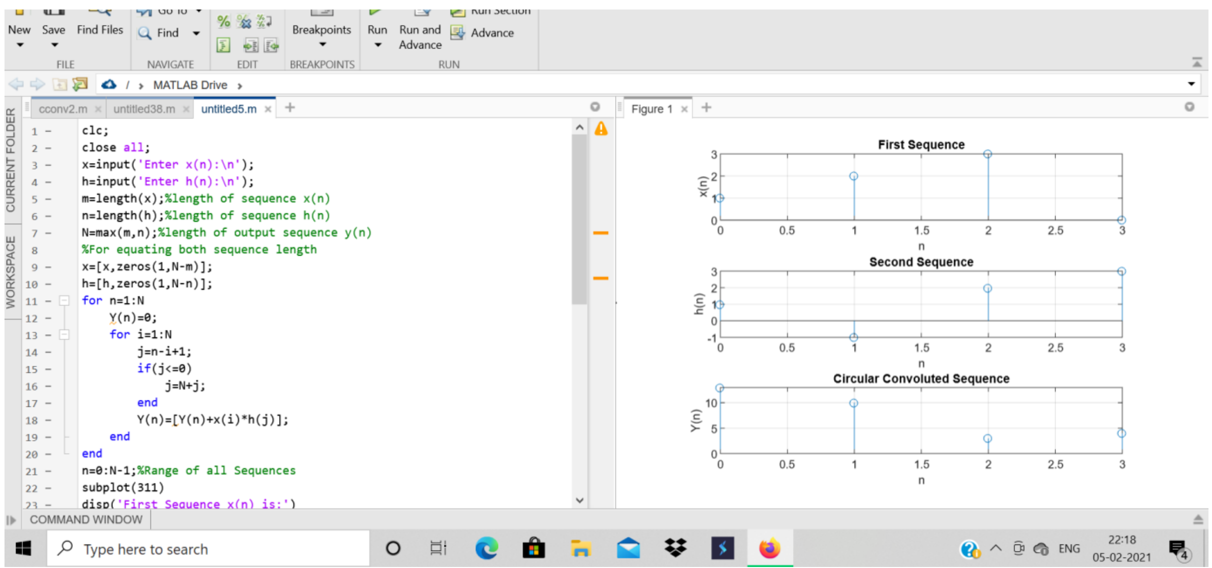
Auto correlation:



Experiment 5: Computation of circular Convolution on discrete signals.

```
clc;
close all;
x=input('Enter x(n):\n');
h=input('Enter h(n):\n');
m=length(x);%length of sequence x(n)
n=length(h);%length of sequence h(n)
N=max(m,n);%length of output sequence y(n)
%For equating both sequence length
x=[x,zeros(1,N-m)];
h=[h,zeros(1,N-n)];
for n=1:N
    Y(n)=0;
for i=1:N
        j=n-i+1;
if(j<=0)
            j=N+j;
end
        Y(n)=[Y(n)+x(i)*h(j)];
end
end
n=0:N-1;%Range of all Sequences
subplot(311)
disp('First Sequence x(n) is:')
disp(x)
stem(n,x)
xlabel('n')
```

```
ylabel('x(n)')
title('First Sequence')
grid on;
subplot(312)
disp('Second Sequence h(n) is:')
disp(h)
stem(n,h)
xlabel('n')
ylabel('h(n)')
title('Second Sequence')
grid on;
subplot(313)
disp('Convoluted Sequence Y(n) is:')
disp(Y)
stem(n,Y)
xlabel('n')
ylabel('Y(n)')
title('Circular Convoluted Sequence')
grid on
```

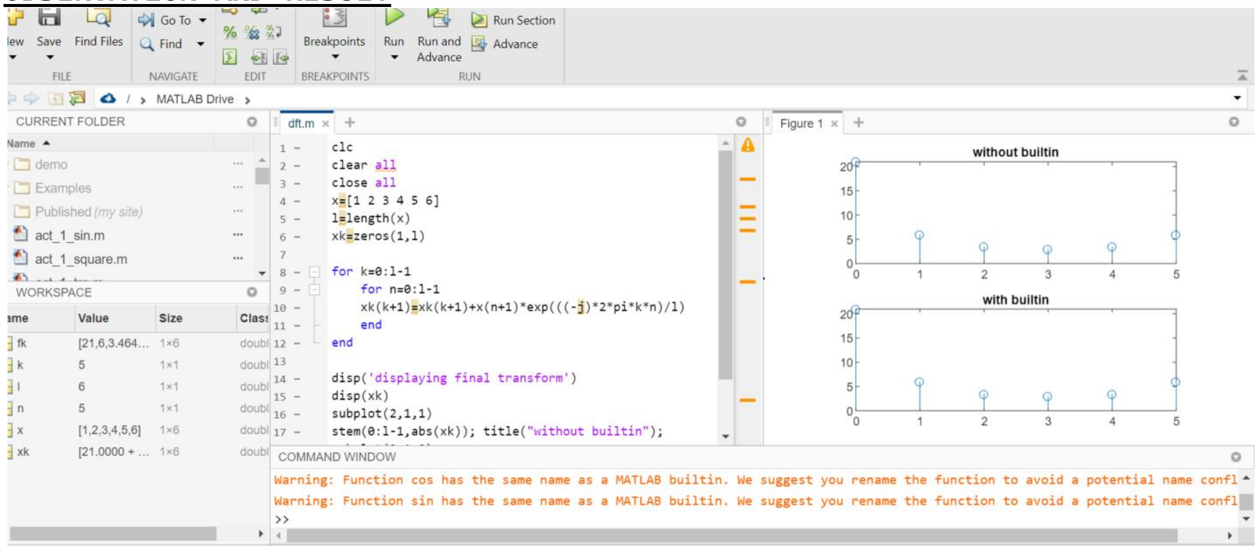


EXPERIMENT – 6: To generate DFT of a signal

```
clc
clearall
closeall
x=[1 2 3 4 5 6]
l=length(x)
xk=zeros(1,l)
fork=0:l-1
forn=0:l-1
xk(k+1)=xk(k+1)+x(n+1)*exp(((-j)*2*pi*k*n)/l)
end
```

```
end
disp('displaying final transform')
disp(xk)
subplot(2,1,1)
stem(0:l-1,abs(xk)); title("without builtin");
subplot(2,1,2)
fk=abs(fft(x))
stem(0:l-1,fk); title("with builtin")
```

OBSERVATION AND RESULT



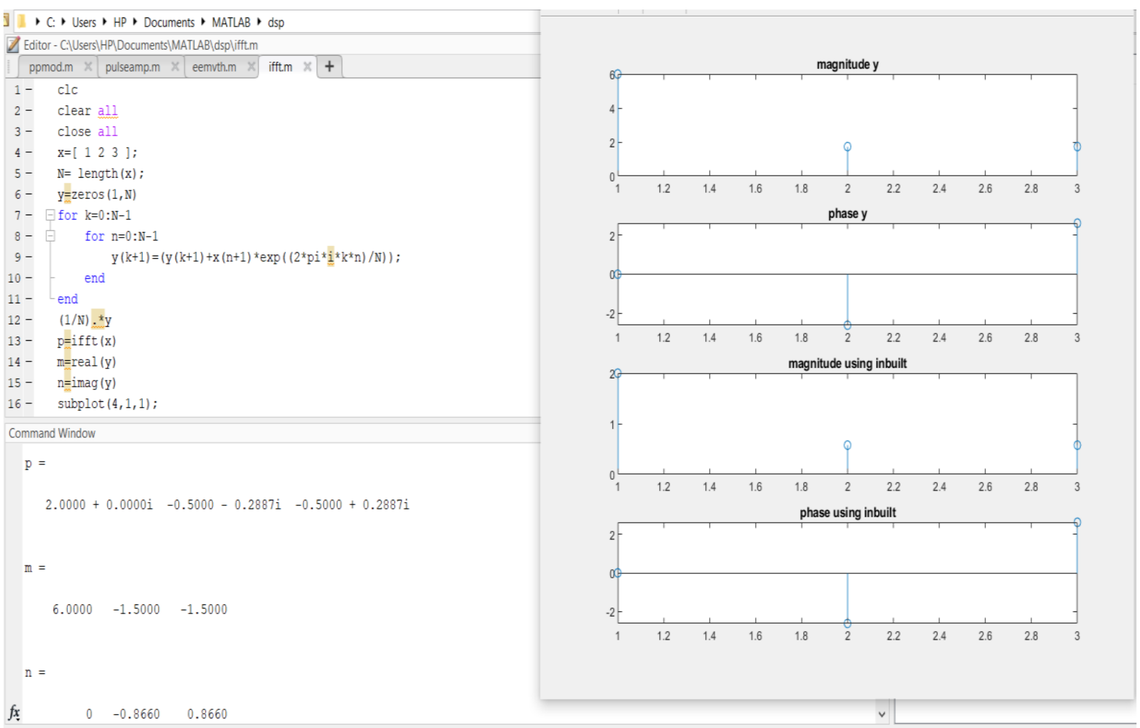
EXPERIMENT-7

Computation of Inverse Discrete Fourier Transform (IDFT)

```
clc
clear all
close all
x=[ 1 2 3];
N= length(x);
y=zeros(1,N)
for k=0:N-1
    for n=0:N-1
        y(k+1)=(y(k+1)+x(n+1)*exp((2*pi*i*k*n)/N));
end
end
(1/N).*y
p=ifft(x)
m=real(y)
n=imag(y)
```

```
subplot(4,1,1);
stem(abs(y));
title ('magnitude y');
z=atan2(n, m)
subplot(4,1,2);
stem(z);
title('phase y');
subplot(4,1,3);
stem(abs(p));
title('magnitude using inbuilt');
subplot(4,1,4);
o=atan2(imag(p),real(p))
stem(o);
title ('phase using inbuilt');
```

Observation in MATLAB:

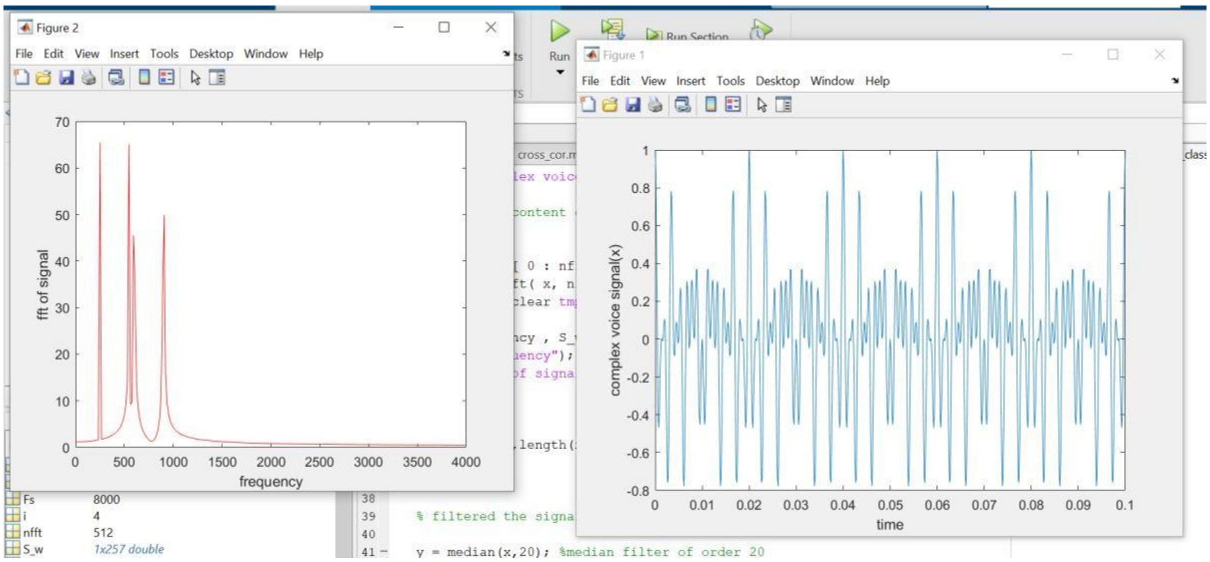


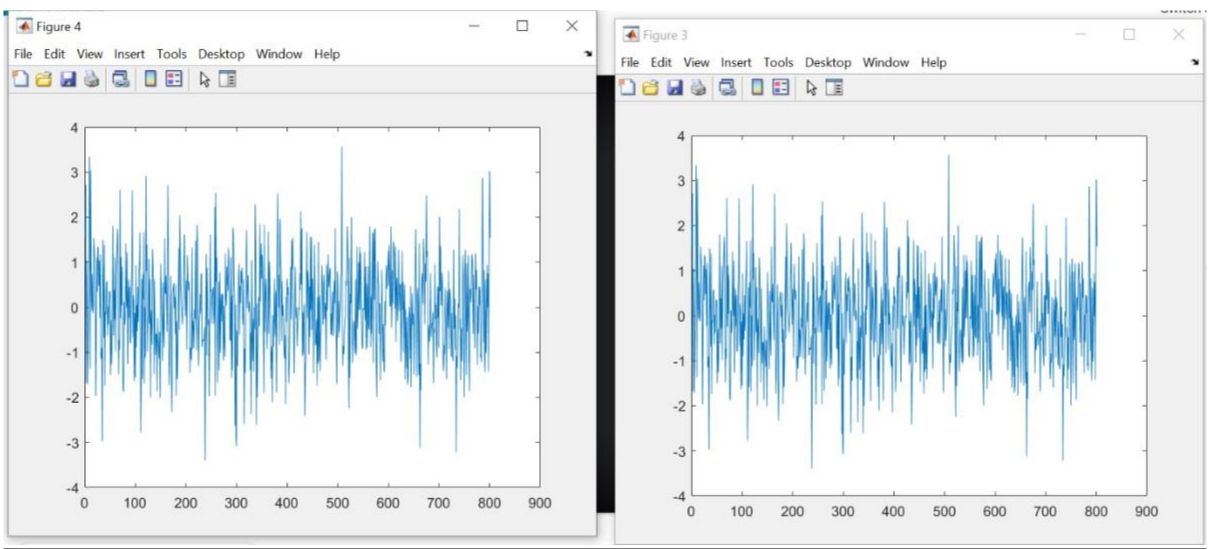
Experiment 8: Analysis of speech signal.

```
clear all;
close all;
clc;
```

```
% Generation of synthetic voice signal;

Fs=8000;
Ts=1/Fs;
time=0:Ts:0.1;
Freqs=[250 550 900 600];
Xs=zeros(length(Freqs),length(t
ime)); fori=1:length(Freqs)
Xs(i,:)=cos(2*pi*Freqs(i)*time);
end
x=sum(Xs);
x=x./max(abs(
x)); figure(1);
plot(time,x);
xlabel("time");
ylabel("complex voice signal(x)");

% frequency content of audio

nfft = 512;
frequency = [ 0 :nfft/2 ]' * (Fs/nfft) ;
tmp = abs( fft( x, nfft )); tmp = tmp( 1 : (nfft/2+1) ) ;
S_w = tmp ; clear tmp ;
figure(2);
plot( frequency , S_w , 'r' ) ;
xlabel("frequency");
ylabel("fft of signal");

% add noise

x=x+ randn(1,length(x));
figure(3);
plot(x);

% filtered the signal using median filtering and plots it
```

y = median(x,20); %median filter of order 20figure(4); plot(y)
The observation is plotted below.



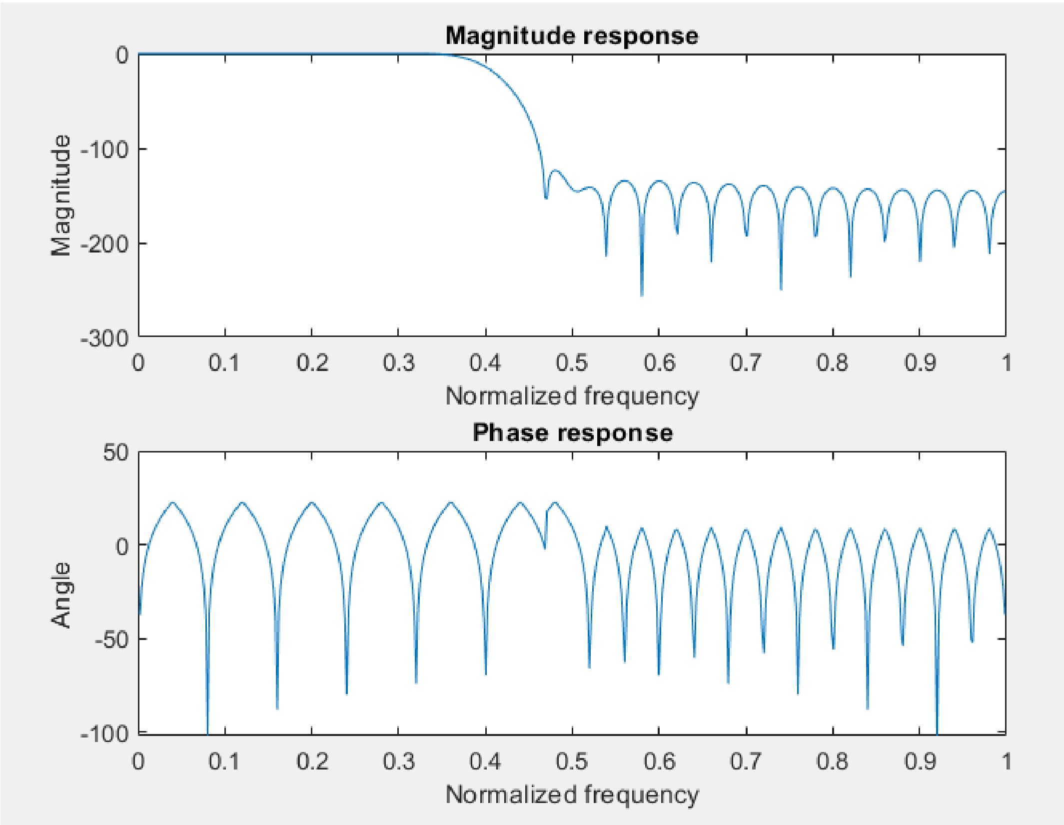


EXPERIMENT- 9

Analysis of filters using different frequency

%FIR filter design by WINDOW method
n=50;
fp=200;
f=1000;
wn=2*(fp/f);

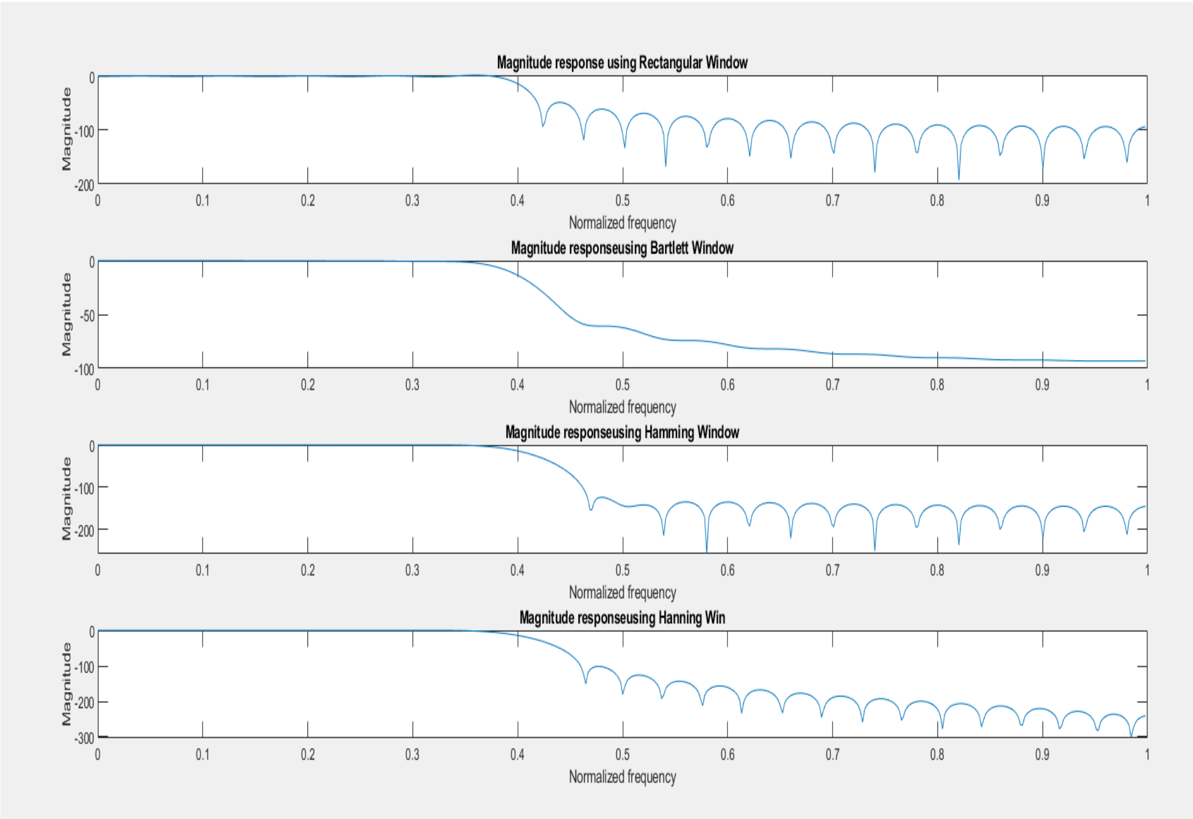
```
b=fir1(n,wn, 'low');
[H,w]=freqz(b,1);
subplot(2,1,1)
plot(w/pi,20*log(abs(H)));
xlabel('Normalized frequency');
ylabel('Magnitude');
title('Magnitude response');
subplot(2,1,2)
plot(w/pi,20*log(angle(H)));
xlabel('Normalized frequency');
ylabel('Angle');
title('Phase response');
```



2)

```
%FIR filter design by various WINDOW methods and comparison
n=50;
fp=200;
f=1000;
window=boxcar(n+1);
```

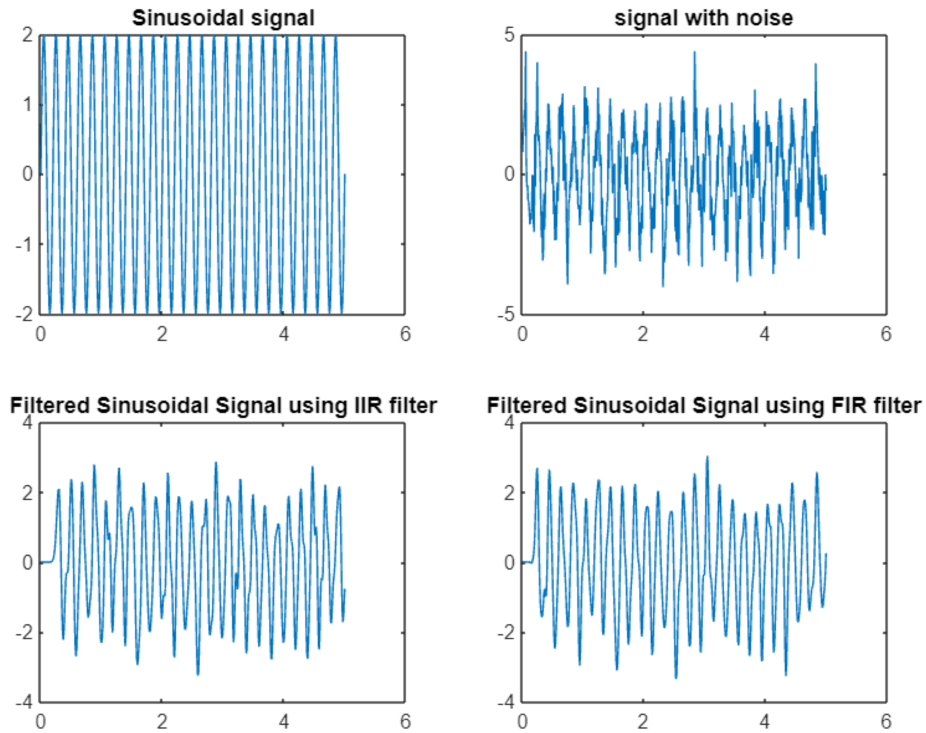
```
window1=bartlett(n+1);
window2=hamming(n+1);
window3=hanning(n+1);
window4=kaiser(n+1);
wn=2*(fp/f);
b1=fir1(n,wn,window);
[H1,w]=freqz(b1,1);
subplot(5,1,1)
plot(w/pi,20*log(abs(H1)));
xlabel('Normalized frequency');
ylabel('Magnitude');
title('Magnitude response using Rectangular Window');
b2=fir1(n,wn,window1);
[H2,w]=freqz(b2,1);
subplot(5,1,2)
plot(w/pi,20*log(abs(H2)));
xlabel('Normalized frequency');
ylabel('Magnitude');
title ('Magnitude response using Bartlett Window');
b3=fir1(n,wn,window2);
[H3,w]=freqz(b3,1);
Subplot (5,1,3)
plot(w/pi,20*log(abs(H3)));
xlabel ('Normalized frequency');
ylabel('Magnitude');
title ('Magnitude response using Hamming Window');
b4=fir1(n,wn,window3);
[H4,w]=freqz(b4,1);
Subplot (5,1,4)
plot(w/pi,20*log(abs(H4)));
xlabel ('Normalized frequency');
ylabel('Magnitude');
title ('Magnitude response using Hanning Win')
```



3. % This file design the LPF and plots its frequency response

```
close all; clear all;

% generate signal
fs = 100; % sampling frequency
f = 5;     % signal frequency
t = 5;      % time duration
n = [0:1/fs:t]; % sample vector

x = 2*sin(2*pi*f*n); % single tone signal

z = awgn(x,1);  % noisy signal

subplot (2,2,1); plot(n,x); title ('Sinusoidal signal');
subplot (2,2,2); plot(n,z); title ('signal with noise');

% IIR Butterworth LPF filter Design
o = 40;  % order of the filter
```

```
Wc = 2*pi*f/fs;     % n
[b,a] = butter(o,Wc,'low'); % Nr and Dr coeff. of IIR butterworth filter

%fvtool(b,a); % filter frequency response

% filter the signal
x_f_iir = filter(b,a,z);

%figure;
Subplot (2,2,3);
plot(n,x_f_iir); title('Filtered Sinusoidal Signal using IIR filter');

%% FIR Filter Design
o = 40;
b = fir1(o,Wc); % FIR low pass filter

%figure;
%freqz(b,1,512); % frequency response of FIR LPF filter

% filter the signal
x_f_fir = filter(b,1,z);

%figure;
Subplot (2,2,4);
plot(n,x_f_fir);
title ('Filtered Sinusoidal Signal using FIR filter');
```

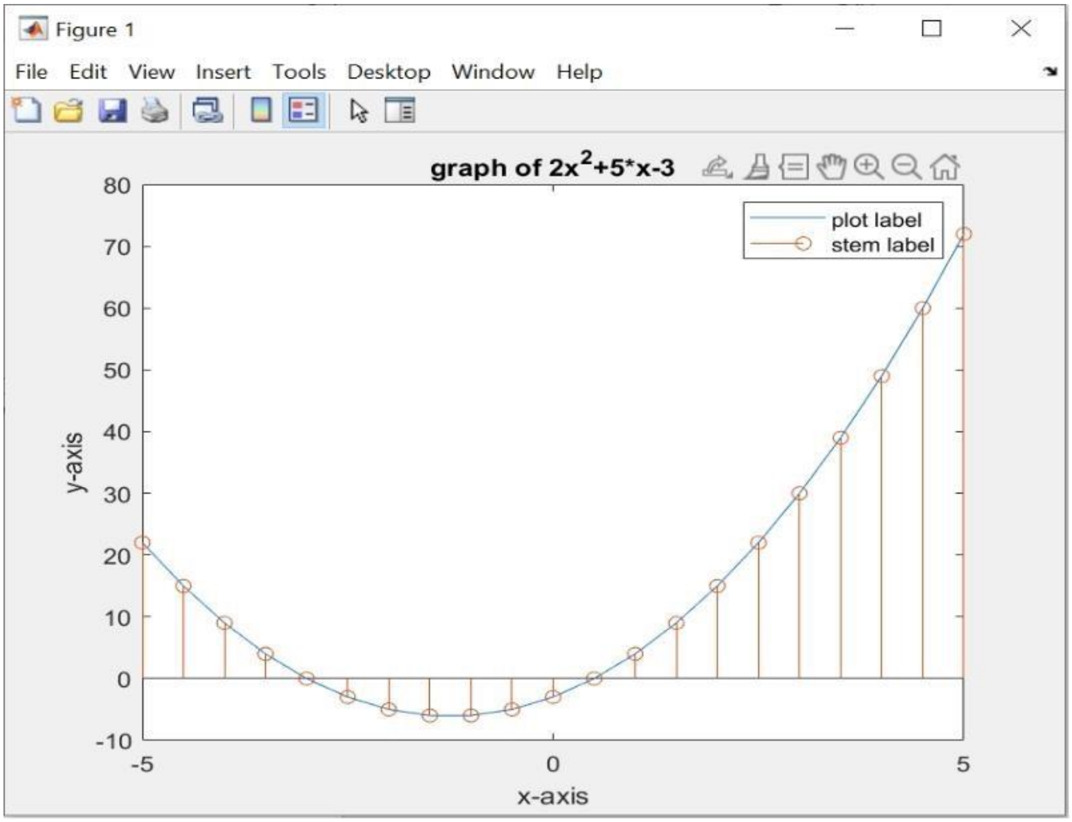


MATLAB

Q- write a MATLAB script to plot and visualize the following function y=2x^2+5*x-3 and -5<=x<=5. Use plot() and stem() to create continuous and discrete plot in a single figure using hold on and hold off command. Provide the legends, title, xlabel and ylabels.

```
x=-5:0.5:5
y=2*x.^2+5*x-3 plot(x,y) hold
on; stem(x,y); hold off;
legend('plot label','stem label')
title("graph of 2x^2+5*x-3");
xlabel("x-axis") ylabel("y-axis")
```

<u>output of Exercise-01</u>



EXERCISE-02

The equation for a 10HZ sine wave with an amplitude of 3 is 3sin(2*pi(10)t). A frequency of 10HZ means the sine wave completes 10 cycles in 1 second. Answer each of the following questions.

1. What does this command do in MATLAB >>t=0:0.001:0.6

Ans-> Assigns array of numbers to t ranging between 0 to 0.6 at an interval of 0.001.

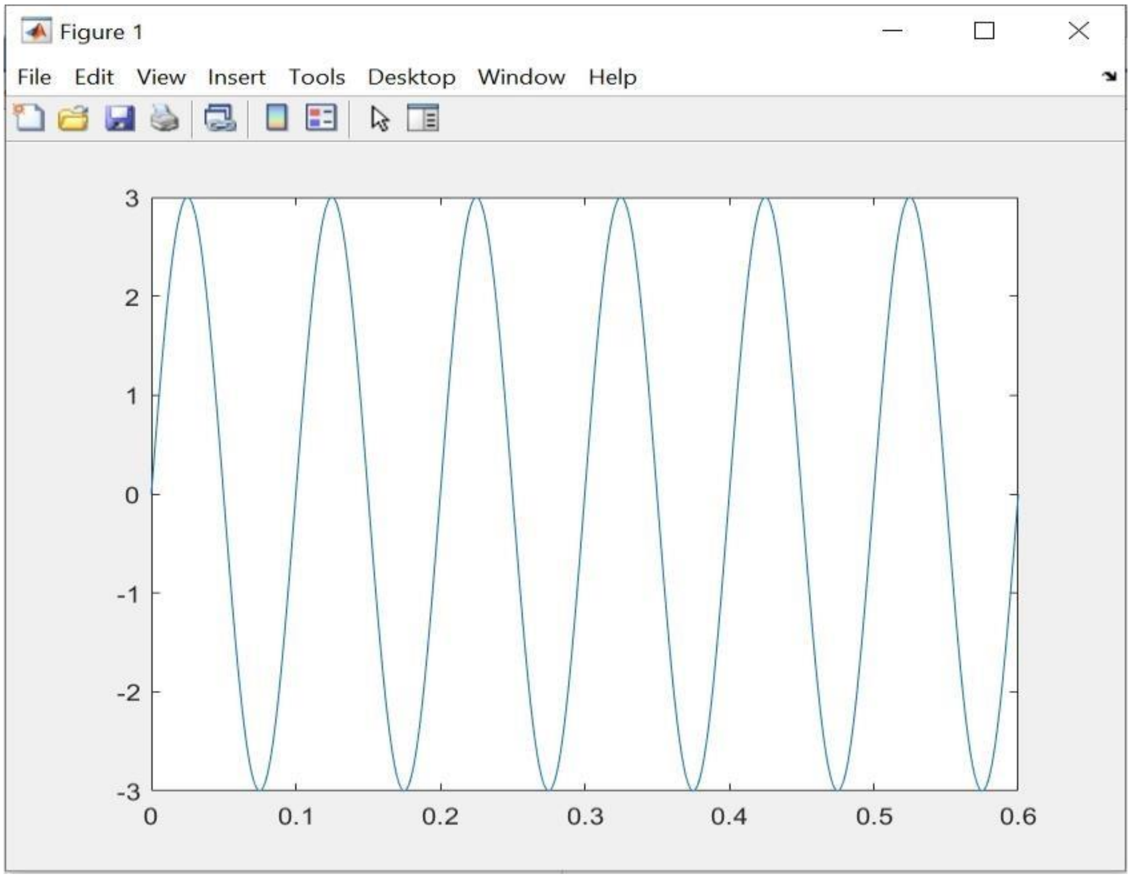
2. What does this command do in MATLAB? >>y=3*sin(2*pi*10*t)

Ans-> This defines how y is going to depend on t. Here y is initialized with a sign curve with amplitude 3 and frequency 10. 3. What does this command do in MATLAB >>plot(t,y)?

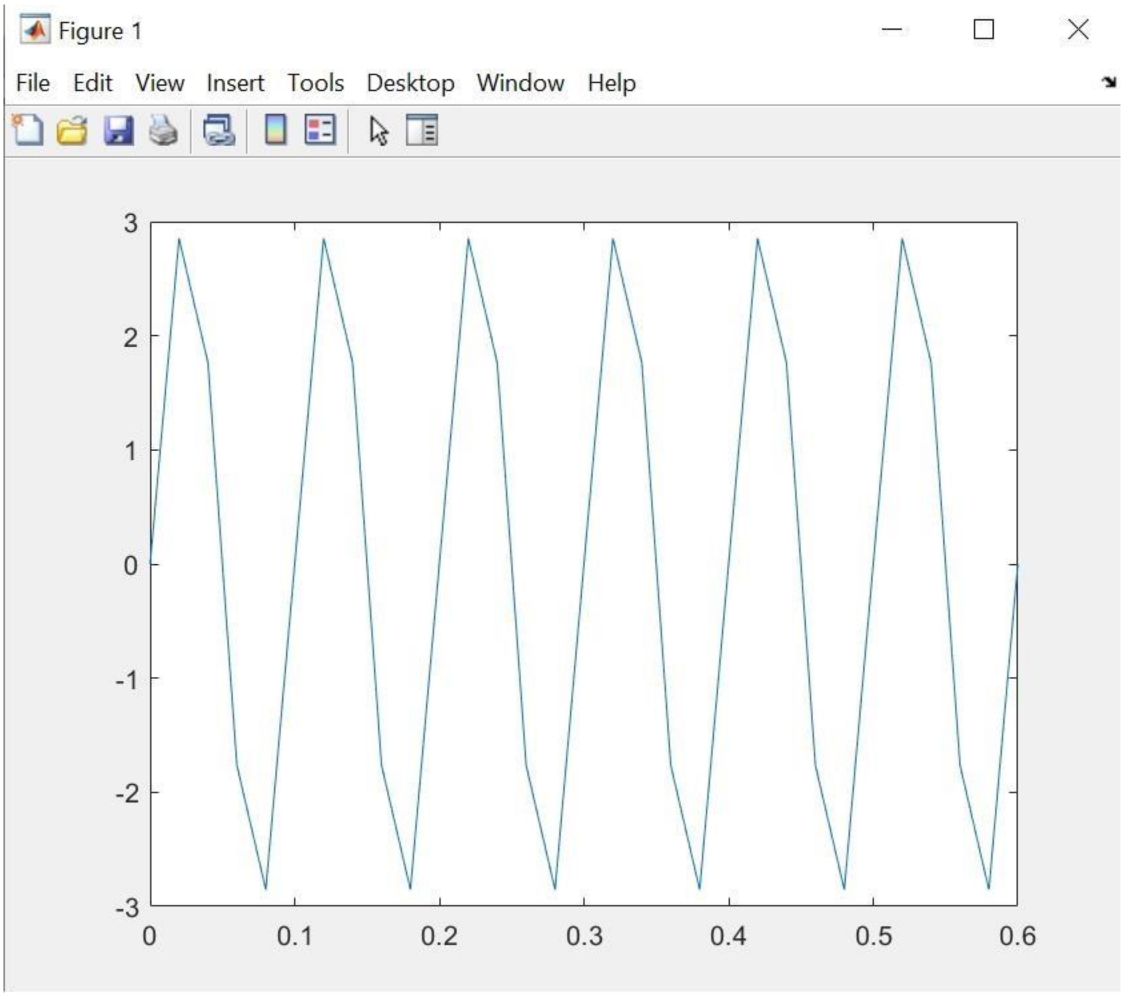
Ans-> plots a curve between t and y taking t in x axis and y in y-axis.

The code and corresponding output is given below:

• f=10;
a=3; t=0:0.001:0.6
y=3*(sin(2*pi*10*t));
plot(t,y);
<u>Output1:</u>



If we write the below code:
t=0:0.02:0.6 y=3*sin(2*pi*10*t);
plot(t,y); output2:



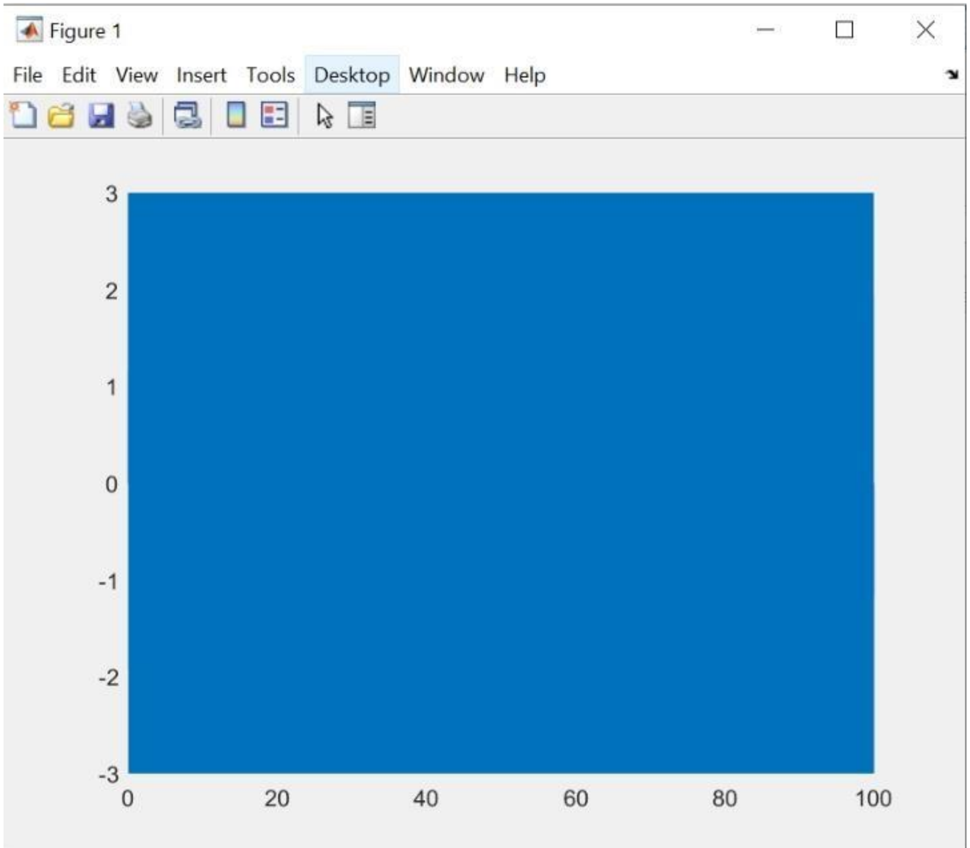
This curve does not look like exact sine curve as the interval is quite large with respect to the range, due to which very few points are plotted and graph is not exact.

If we write the below code:

t=0:0.001:100;

y=3*sin(2*pi*10*t) plot(t,y);

output:



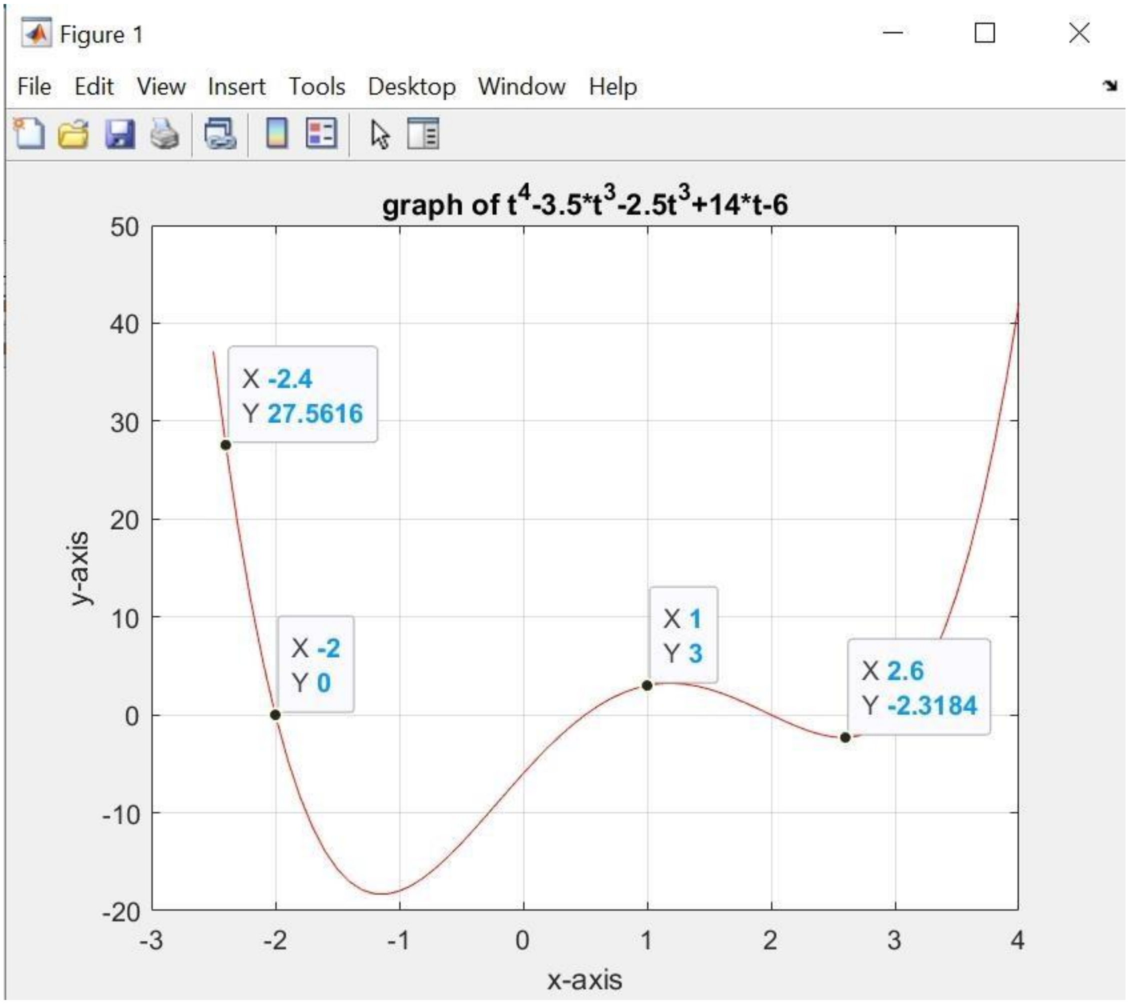
This curve does not look like exact sine curve as the interval is very small with respect to the range, due to which large number of points are plotted and graph appear as solid but the sine nature can be observed on zooming in the graph.

EXERCISE-03

Q- Plot the polynomial f(t)=t^4-3.5*t^3-2.5*t^2+14*t-6 from t=-2.5 to 4. Add a label on the x-axis , a label on the y axis, atitle, and a grid. Also change the colour of the graph to something other than blue. Use the data cursor to estimate all of the roots of the polynomial.

~~Answer:~~

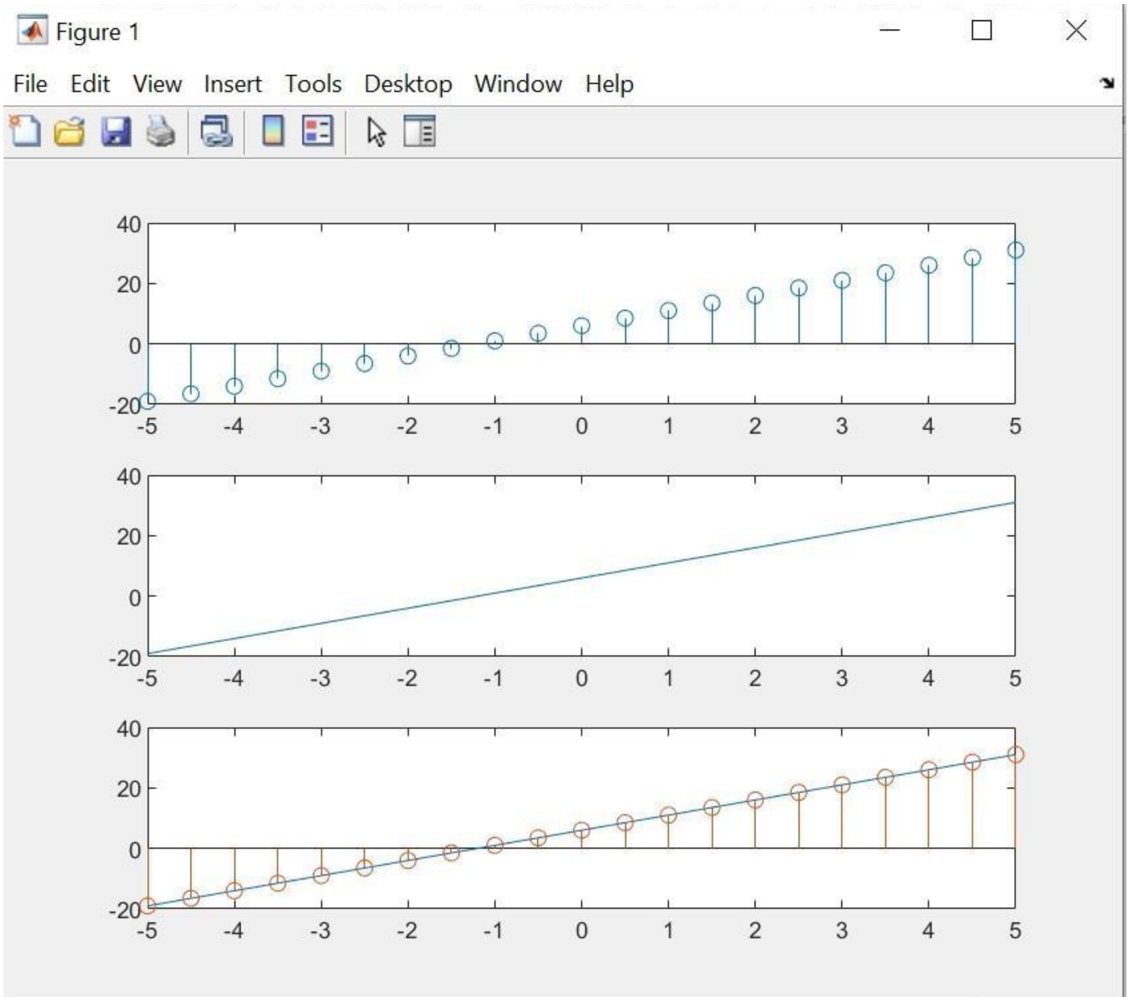
```
t=-2.5:0.1:4;
y=t.^4-(3.5*t.^3)-(2.5*t.^2)+14*t-6; plot(t,y,'r'); xlabel("x-
axis"); ylabel("y-axis");
title("graph of t^4-3.5*t^3-2.5*t^2+14*t-6"); grid();
```



EXERCISE-04

Q- Write a MATLAB to plot straight line y=5*x+6. Use subplot command and plot discrete straight line plot in 1st subplot,continuous straight line plot in 2nd plot and combined straight line plots in 3rd subplot window.

Answer-> x=-5:0.5:5; y=5*x+6;
subplot(3,1,1); stem(x,y);
subplot(3,1,2); plot(x,y);
subplot(3,1,3); plot(x,y);hold on;
stem(x,y); hold off; **OUTPUT:**



1.Do the following in a script file. . (a) Write a set of input statements that prompt the user for the year, month (***a string***), and the day that he/she was born.

(b) Write a single fprintf statement that will display the user's birthday using the variables generated from your input statements.

Solution->

variable=input("Enter your birth year,month,day",'s'); fprintf("your birthday is on %s",variable)

2. **.** Do the following in a script file. (a) Write a set of input statements that prompt the user for his/her favorite band, favorite song, favorite restaurant, and favorite food.

(b) Write two fprintf statements. The first statement should display the user's favorite band and favorite song. The second statement should display the user's favorite restaurant and favorite food on a new line.

Solution->

```
band=input("enter your favourite band ",'s'); song=input("enter your favourite song
",'s'); restaurant=input("enter your favourite restaurant  ",'s'); food=input("Enter your
favourite food
",'s'); fprintf("The entered band and song is
%s %s respectively\n",band,song);
fprintf("your favourite restaurant and food is %s
%s respectively",restaurant,food);
```

3. **.** In this problem, you will be writing a MATLAB script that will determine a person's blood pressure category. The user will have to input his/her systolic and diastolic blood pressure readings. The program will determine the user's category and then output the statement Your Blood Pressure Reading Indicates: followed by the correct category. The following table will be of use.

Category	Systolic	Diastolic
Hypotension Low Blood Pressure	50-89	35-59
Normal	90-119	60-79
Pre-hypertension	120-139	80-89
Mild Hypertension (Stage 1)	140-159	90-99
Moderate Hypertension (Stage 2)	160-179	100-109
Severe Hypertension (Stage 3)	180-209	110-119
Very Severe Hypertension (Stage 4)	210-239	120-135

Solution->

```
systolic=input("Enter your systolic boold pressure
reading"); diastolic=input("Enter your diastolic boold pressure
reading");
if(systolic>=50&&systolic<=8&&diastolic>=35&&diastolic<=59)
   fprintf("Your Blood Pressure Reading Indicates Hypotension Low Blood
Pressure");  elseif(systolic>=90&&systolic<=119&&diastolic>=60&&diastolic<=79)
fprintf("Your Blood Pressure Reading Indicates Normal");
elseif(systolic>=120&&systolic<=139&&diastolic>=80&&diastolic<=89)
fprintf("Your Blood Pressure Reading Indicates Pre-hypertension");
elseif(systolic>=140&&systolic<=159&&diastolic>=90&&diastolic<=99)
fprintf("Your Blood Pressure Reading Indicates Mild hypertension(stage
1)");  elseif(systolic>=160&&systolic<=179&&diastolic>=100&&diastolic<=109)
fprintf("Your Blood Pressure Reading Indicates Moderate hypertension(stage
2)");  elseif(systolic>=180&&systolic<=209&&diastolic>=110&&diastolic<=119)
fprintf("Your Blood Pressure Reading Indicates severe hypertension(stage 3)");
elseif(systolic>=210&&systolic<=239&&diastolic>=120&&diastolic<=135)
fprintf("Your Blood Pressure Reading Indicates very severe hypertension(stage 4)");
else
```

fprintf("Please Enter correct boold presuure reading\n"); end

4. How many times will the ***for loop*** in the following code run and what will the output be? Work this out by hand and just use MATLAB to check your answer. x = 2; sum = 1; for k = 1:5 sum = 1 + 1/x*sum;

end disp('sum ='); disp(sum)

Solution->

The for loop will be executed 5 times. The output will be 1.9688.

Step-1:

When x=2 and sum=1 and k=1 Sum value is 1.5 Step-2:

When x=2 and sum=1.5 and k=2 Sum value is 1.75 Step-3:

When x=2 and sum=1.75 and k=3 Sum value is 1.875 Step-4:

When x=2 and sum=1.875 and k=4 Sum value is 1.9375 Step-5:

When x=2 and sum=1.9375 and k=5

Sum value will be 1.96875

5. How many times will the ***while loop*** in the following code run and what will the ouput be? Again, work this out by hand and just use MATLAB to check your answer.

sum = 0; while sum <=10
sum = sum + 3;

end disp('sum =');disp(sum)

solution->

The while Loop will be executed 4 times and the output will be sum= **12.**

Step-1:

In 1st step sum is less than or equal to 10. So we will go into the loop and the sum will be 3.

Step-2:

Then 3<=10. Sum=3+3=6 Step-3:

Then 6<=10 sum=6+3=9

Step-4:

Then 9<=10 sum=9+3=12

Step-5:

12 is not less than or equal to 10 so,it will come out of the loop. And sum will be **12.**

6. **What value will the following program print?**

```
count = 0;  for d
=  1:7    for  h  =
1:24     for  m  =
1:60      for  s  =
1:60     count  =
count + 1;   end
end    end   end
count
```

What is a simpler way to achieve the same results?

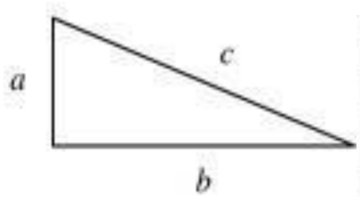
Solution->

The output of the above program is 604800.

The simpler way to get the same result is given below:

```
count=0; for k=1:604800
count=count+1;       end
disp(count)
```

7 **.** Consider the right triangle shown in figure. Suppose you wish to find the length of the hypotenuse **c** of this triangle for several combinations of side lengths **a** and **b**; the specific combinations of **a** and **b** are given in Table. Write an m-file to do this using **for** loop.



a	b
1	1
1	2
2	3
4	1
2	2

Solution->

```
 variable=input('How many values of which you  want to find the hypertenuse')
for i=1:variable     a=input("enter value of a")     b=input("Enter value of b")
if(a==1)        c=sqrt(a^2+b^2)        disp("for a=1 the value of
hypertenuse");disp(c)     elseif(a==2)        c=sqrt(a^2+b^2)        disp("for a=2
the value of hypertenuse is");disp(c)     else          c=sqrt(a^2+b^2)
       disp("for a=4 the value of hypertenuse is");disp(c);     end end
```

8 **.** Write a script that will display each of the following shapes using asterisks *.

The user should be able to specify the size of each shape.

Solution-> **<u>For solid square:</u>**

```
n=input("Enter value of n"); for i=1:n
   for j=1:n        fprintf('*')     end     fprintf('\n') end
```

<u>For Open Square:</u>

```
n=input("Enter the value of n") for i=1:n     for j=1:n
       if(i==1||i==n||j==1||j==n)           fprintf('*')        else             fprintf(" ")
end     end     fprintf('\n') end
```

For Triangle:

```
n=input('enter value of n') for i=1:n     for j=1:(n-
i)        fprintf(' ')     end     for j=1:(2*i-1)
fprintf('*')     end     fprintf('\n') end
```

NAME: SANTOSHINI PRADHAN

ID: B219049

BRANCH: ETC