**Introduction to Unite Impulse and unite step Sequence**

**LAB # 07**

****

**Spring 2021**

**CSE301L-Signal $ System**

Submitted by: **Ashfaq Ahmad**

Registration No: **19PWCSE1795**

Class Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Engr Durr-e-Nayab**

June 18, 2021

**Department of Computer Systems Engineering**

**University of Engineering and Technology, Peshawar**

**OBJECTIVES OF THE LAB**

‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

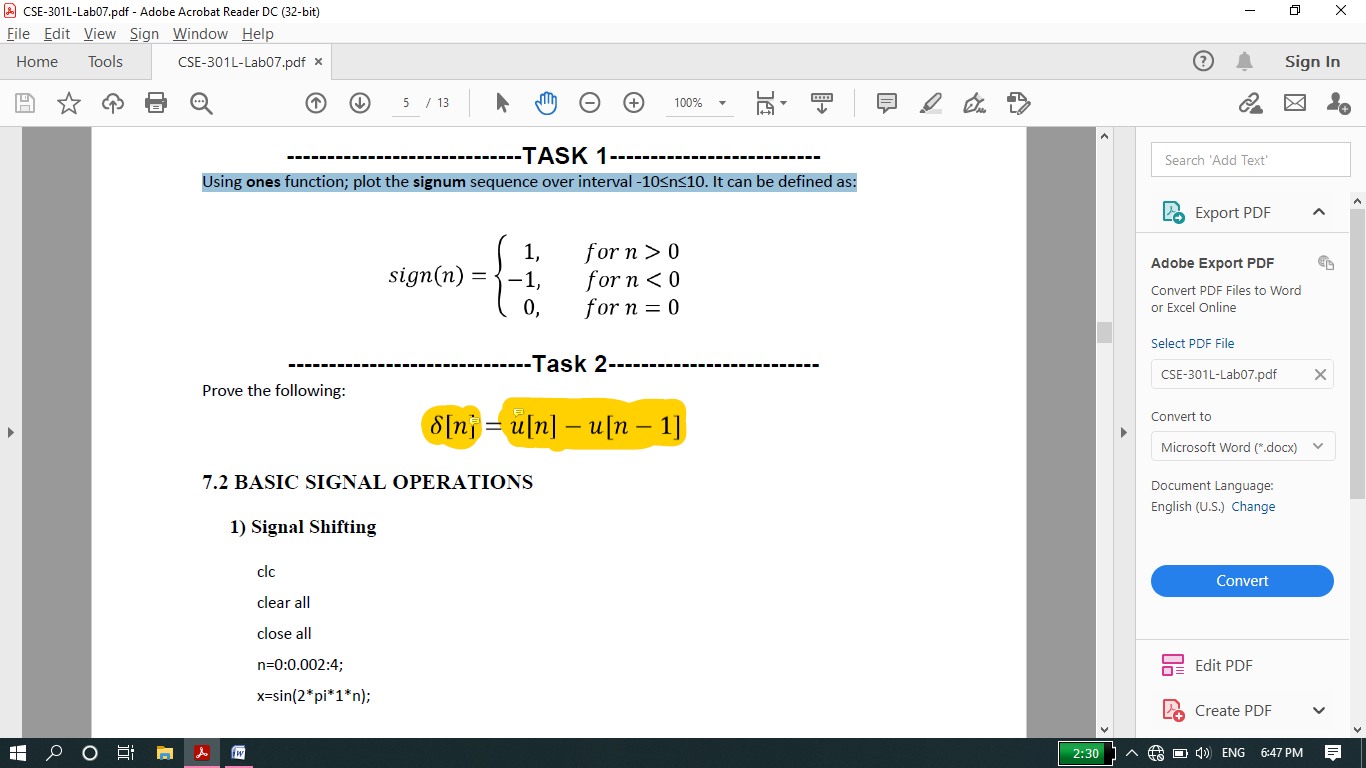
In this lab, we will cover the following topics:

* Generating unit impulse and unit step sequences
* Basic signal operations

‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐‐

**-------------------------TASK 01--------------------------**

* Using **ones** function; plot the **signum** sequence over interval ‐10≤n≤10. It can be defined as:

****

**Source code:**

function task01

disp('\*\*\*\*\*\*\*\*\*\*task no 1\*\*\*\*\*\*\*\*\*')

n=-10:10;

signal=zeros(1,21);%we can also remove line 4 it still work.

%1=row 21=coloumns.

signal(1:10)=-1;

%here automatically row=1 or we can also declear it i,e signal(1,1:10)...

signal(12:21)=1;

figure(1)

stem(n,signal,'b','linewidth',2);

%2nd method

for i=1:21;

if i<=10;

x(i)=-1;

else

if i==11;

x(i)=0;

else

if i>11;

x(i)=1;

end

end

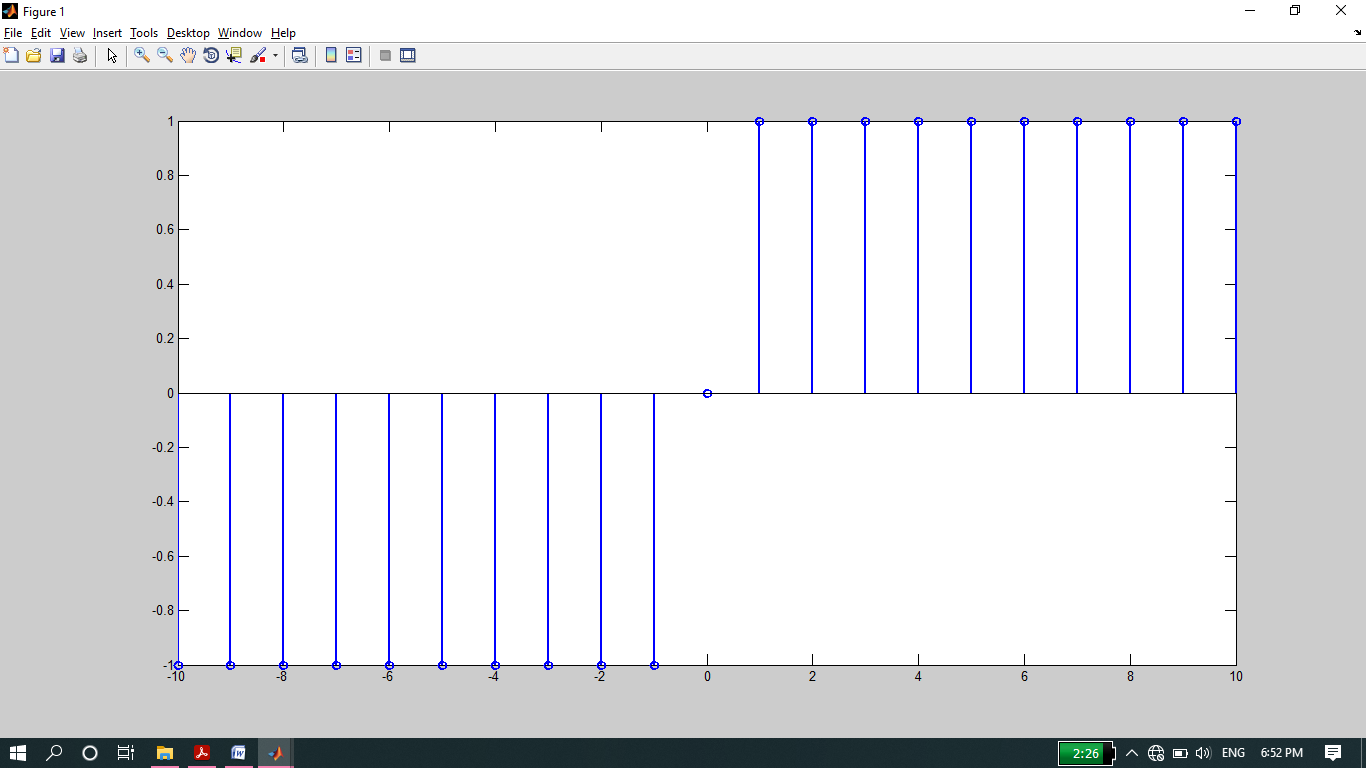
end

end

figure(2)

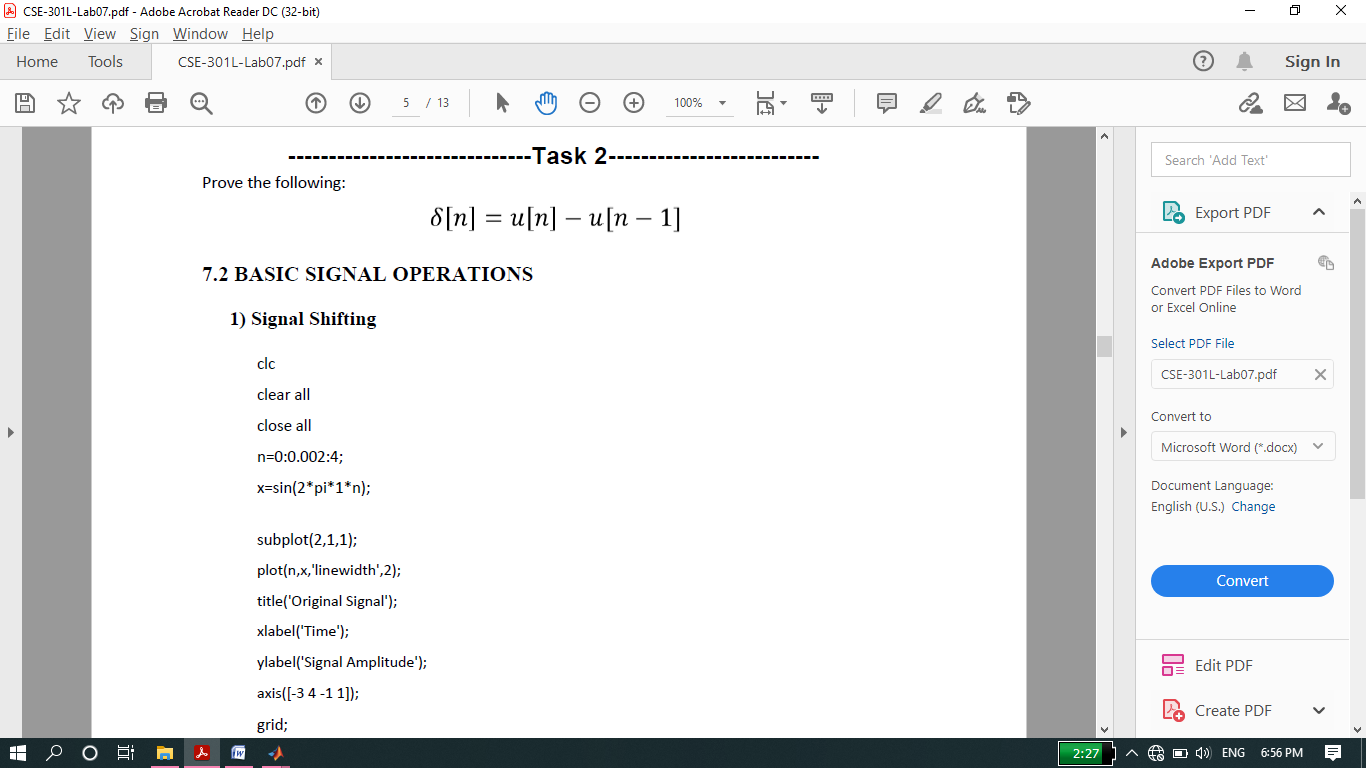
stem(n,x,'r','linewidth',2);

**OUTPUT:**



**---------------TASK 02--------------------------**

* Prove the following:

****

**Source code:**

function task02

disp('\*\*\*\*\*\*\*\*\*\*task no 2\*\*\*\*\*\*\*\*\*')

n=-10:10;

signal1(11:21)=1; %it automatically assign signal(1:10)=0

signal2(12:21)=1;

signal=signal1-signal2;

figure(1)

subplot(3,1,1)

stem(n,signal1,'filled');

title('signal 1');

subplot(3,1,2)

stem(n,signal2);

title('signal 2')

subplot(3,1,3);

stem(n,signal,'filled');

title('Unite Sample')

%2nd method

for i=1:21

if i>=11;%it automatically assign x(1:10)=0

x(i)=1;

end

end

%instead of for loop we can also write here x(11:21)=1;

x2(12:21)=1;%it automatically assign x2(1:11)=0

k=x-x2;

figure(2)

subplot(3,1,1)

stem(n,x,'filled');

title('signal 1');

subplot(3,1,2)

stem(n,x2);

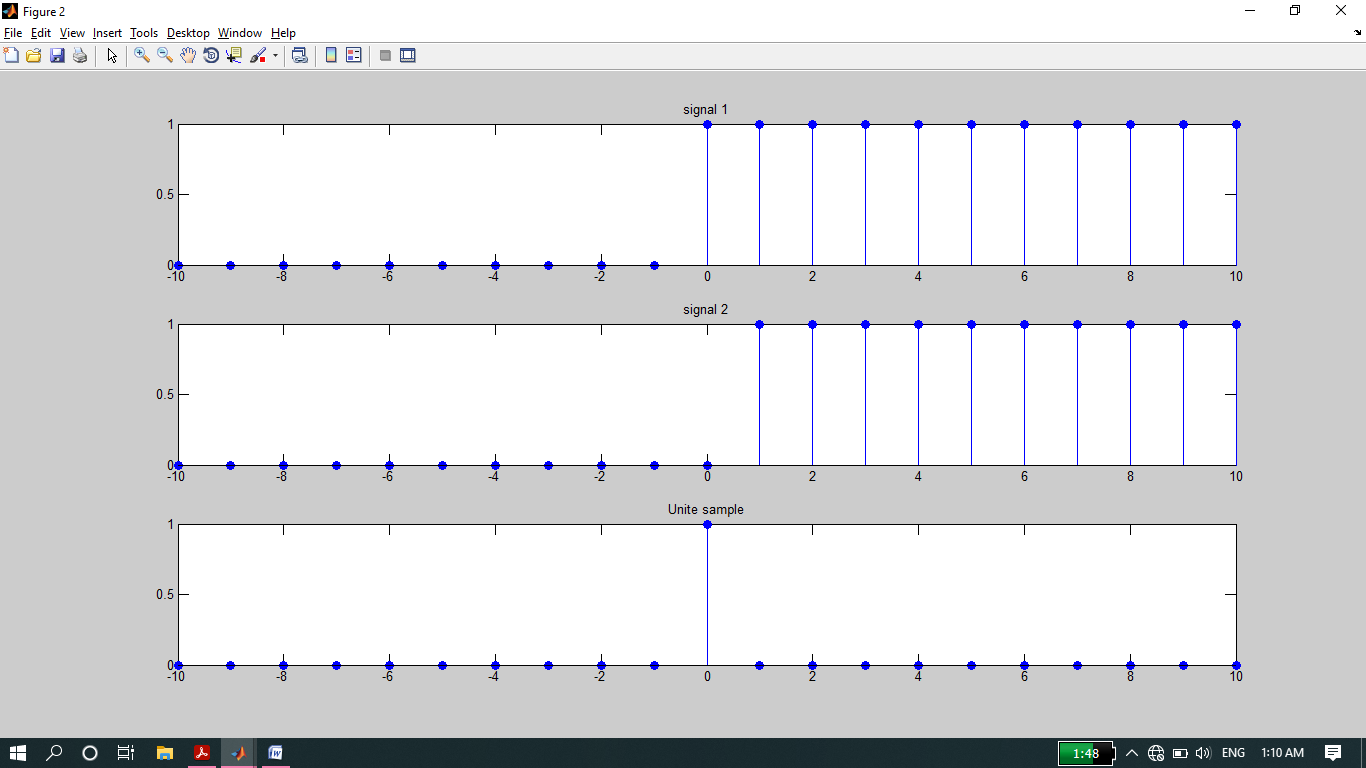
title('signal 2')

subplot(3,1,3);

stem(n,k,'filled');

title('Unite sample')

**Output:**



**-------------------------TASK 03--------------------------**

* Delay the **original signal** given in above example by 1 sec. Plot both the delayed & original signal on the same figure.

**Source code:**

clc

clear all

close all

disp('\*\*\*\*\*\*\*\*\*task no 3\*\*\*\*\*');

n=0:0.002:4;

x=sin(2\*pi\*1\*n);

%the size of n and x is same.

subplot(2,1,1)

plot(n,x);

axis([-1 5 -1 1]);

title('Original signal');

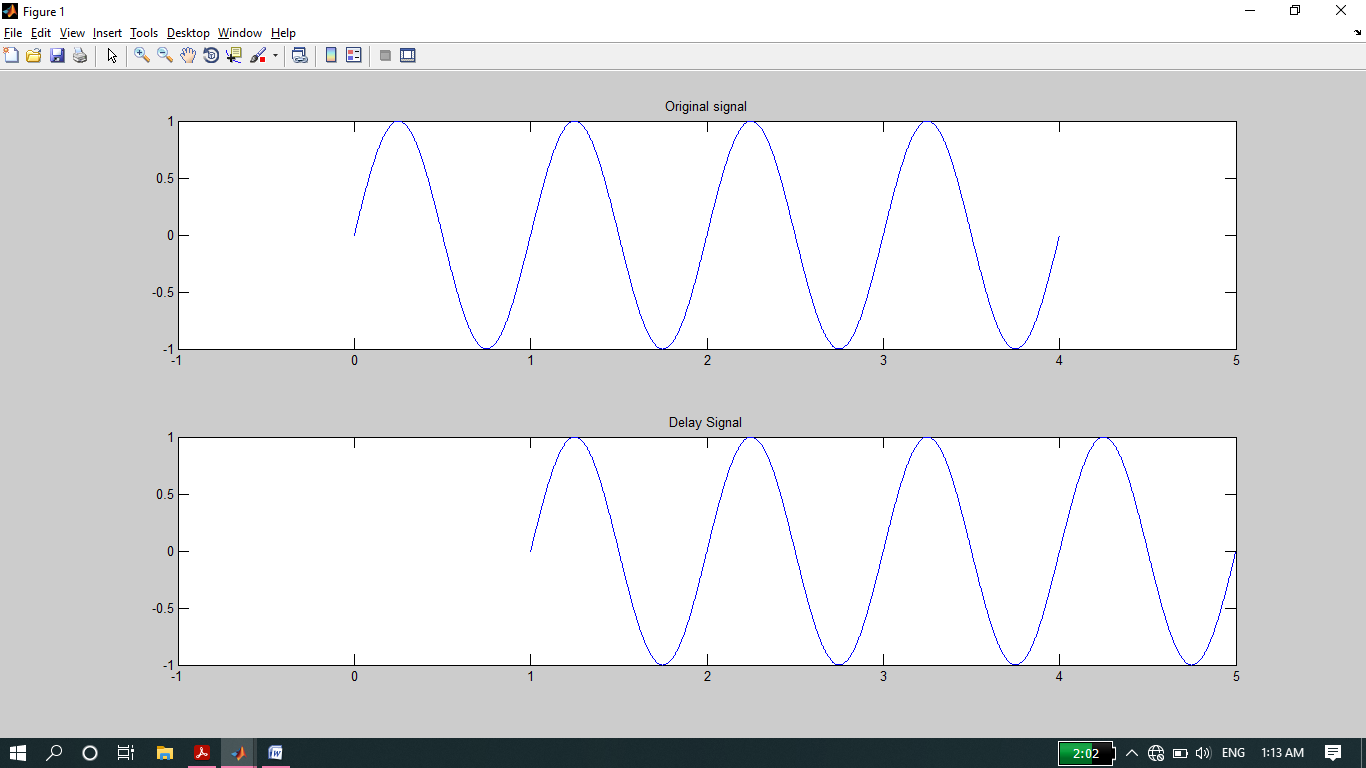
subplot(2,1,2)

plot(n+1,x);

title('Delay Signal');

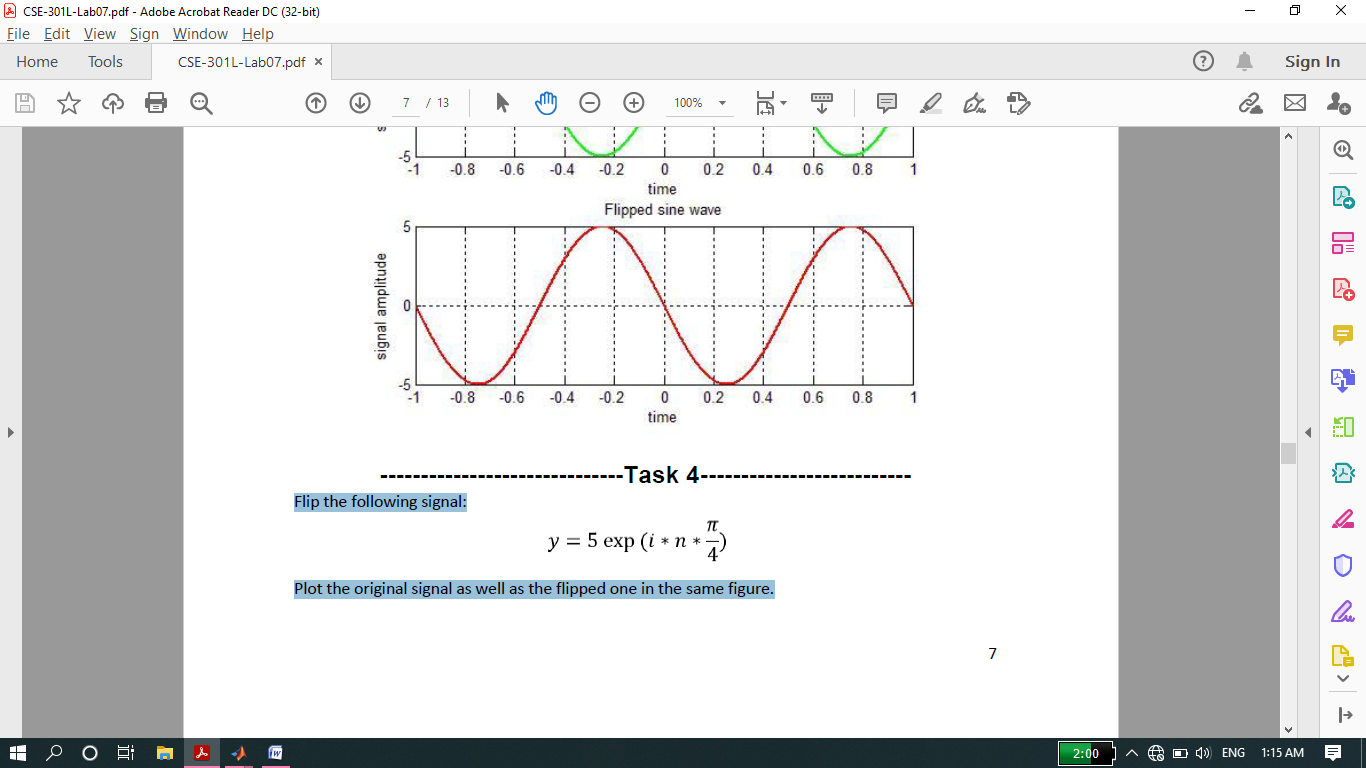
axis([-1 5 -1 1]);

**Output:**



**-------------------------TASK 04--------------------------**

* Flip the following signal:



* Plot the original signal as well as the flipped one in the same figure.

**Source Code:**

clc

clear all

close all

disp('\*\*\*\*\*\*task 04\*\*\*\*')

n=2:0.1:5;

y=5\*exp(n\*(pi/4\*i));

subplot(1,2,1)

stem(n,y,'filled')

title('original signal');

xlabel('Time');

ylabel('Amplitude');

subplot(1,2,2)

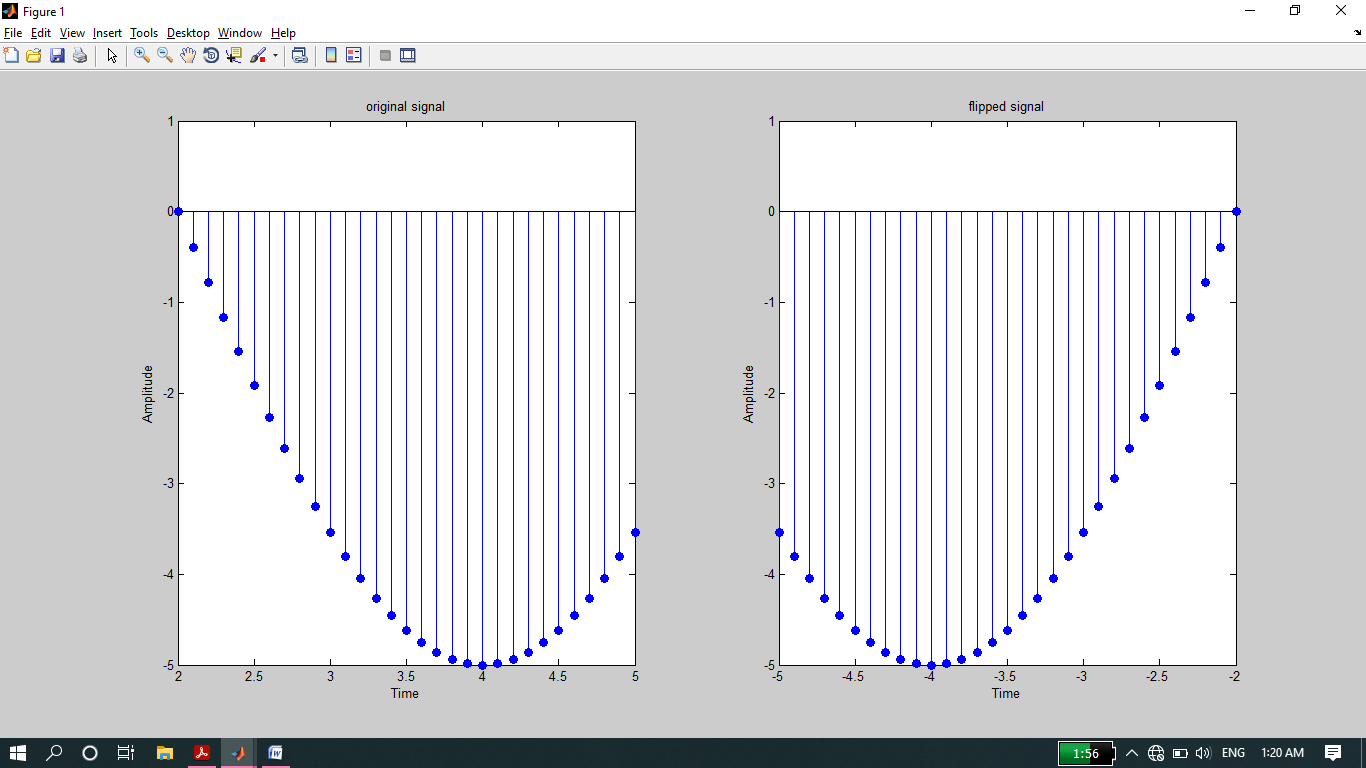
stem(-n,y,'filled');

title('flipped signal');

xlabel('Time');

ylabel('Amplitude');

**Output:**



**-------------------------TASK 05--------------------------**

* Flip the following signal:

x[n]= 2δ[n]+ 5δ[n‐1] + 8δ[n‐2] + 4δ[n‐3] + 3δ[n‐4]

* Plot the original signal as well as the flipped one in the same figure.

**Source code:**

clc

clear all

close all

disp('\*\*\*\*\*\*\*Task no 05\*\*\*\*\*\*\*\*\*');

n=-10:10;

sig\_1=2\*[zeros(1,10) 1 zeros(1,10)]

sig\_2=5\*[zeros(1,11) 1 zeros(1,9)]

sig\_3=8\*[zeros(1,12) 1 zeros(1,8)]

sig\_4=4\*[zeros(1,13) 1 zeros(1,7)]

sig\_5=3\*[zeros(1,14) 1 zeros(1,6)]

x=sig\_1+sig\_2+sig\_3+sig\_4+sig\_5

subplot(2,1,1)

stem(n,x,'r','linewidth',2);

xlabel('time');

ylabel('amplitude');

title('Original Signal');

subplot(2,1,2)

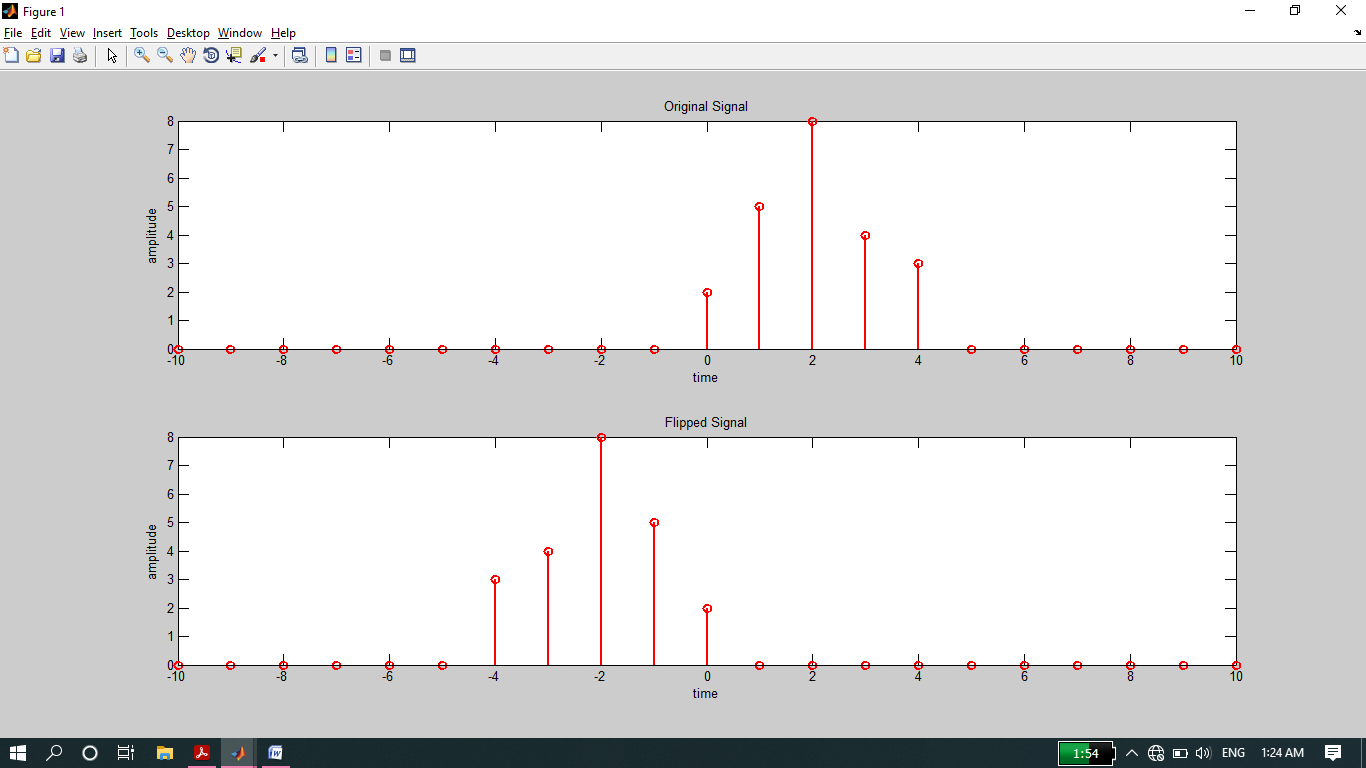
stem(-n,x,'r','linewidth',2);

xlabel('time');

ylabel('amplitude');

title('Flipped Signal');

**Output:**



**-------------------------TASK 06--------------------------**

* Scale the continuous‐time sinusoid used in signal shifting example by a factor of 2.

**Source code:**

function task06

disp('\*\*\*\*\*\*\*\*Task no 06\*\*\*\*\*\*\*');

n=1:0.002:4;

x=sin(pi\*2\*n);

subplot(2,1,1)

plot(n,x,'r','linewidth',2);

xlabel('Time');

ylabel('Amplitude');

title('Original signal');

grid on

axis([0 5 -3 3]);

subplot(2,1,2)

plot(n,2\*x,'b','linewidth',2);%Megnituded scaled by 2

xlabel('Time');

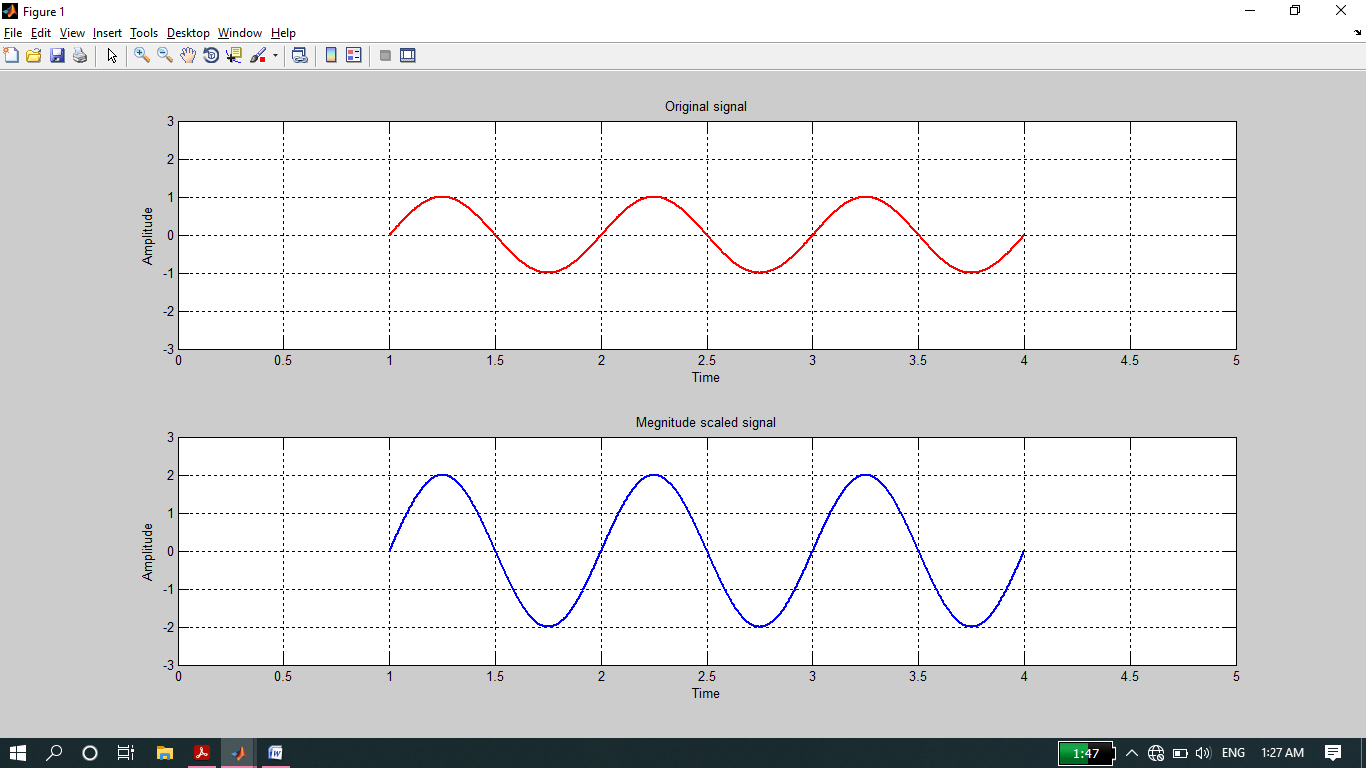
ylabel('Amplitude');

title('Megnitude scaled signal');

grid on

axis([0 5 -3 3]);

**Output:**

****

**-------------------------TASK 07--------------------------**

* Use ***interp*** command in the above program to interpolate (up‐sample) the signal by a factor of 2.

**Source code:**

clc

clear all

close all

disp('\*\*\*\*\*Task No 07\*\*\*\*\*\*');

n=-2:1/1000:2;

x1=sin(2\*pi\*2\*n);

x2=interp(x1,3); %up-sampling

%INCASE of x1 if one cycle cover 100 sample then one cycle will cover

%100 \*3 incase of x2 this is called positive time scaling or up sampling.

%for up time scaling command is inpterp(interpolate)

%for down time scaling (down sampling) command is decimate.

subplot(2,1,1)

plot(x1,'b');

xlabel('x-axis');

ylabel('y-axis');

title('original signal');

axis([0 4000 -1 1])

subplot(2,1,2)

plot(x2,'b');

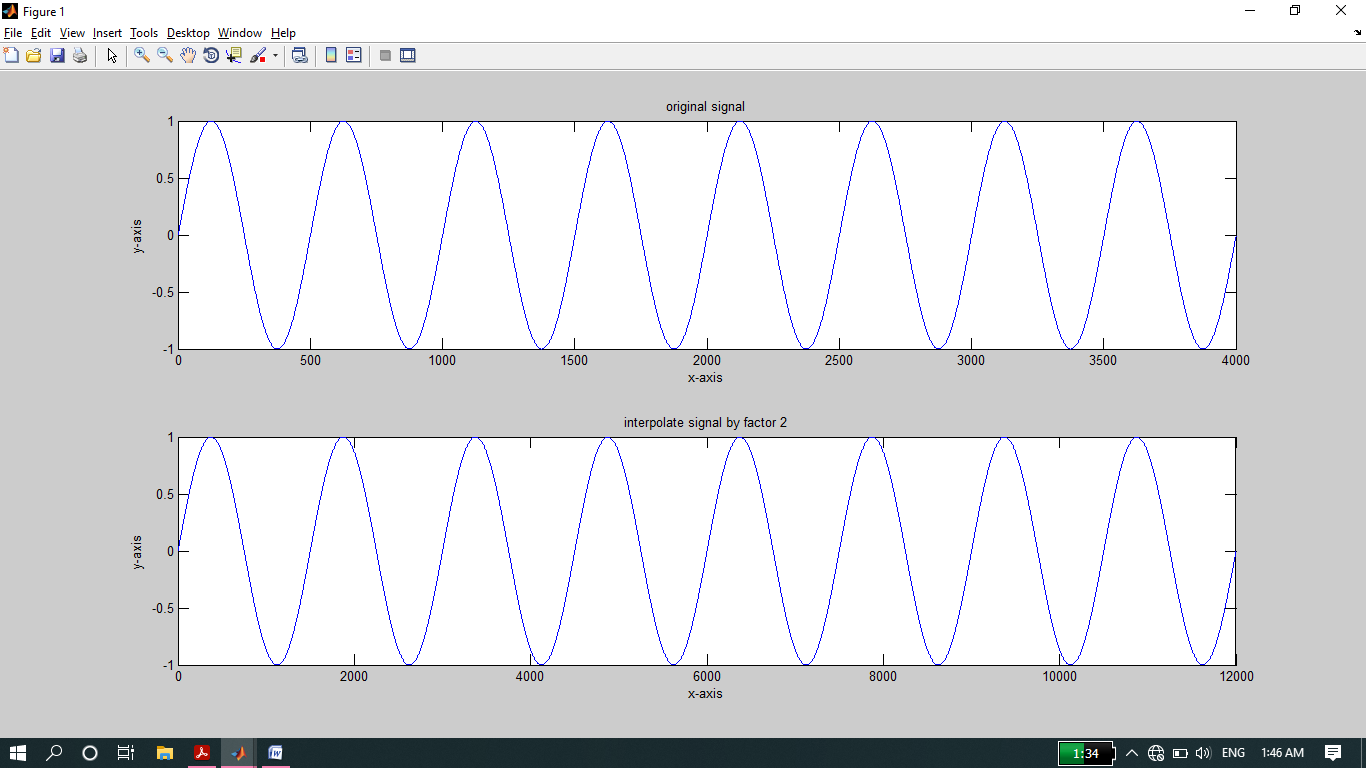
xlabel('x-axis');

ylabel('y-axis');

title('interpolate signal by factor 2');

axis([0 12000 -1 1])

**Output:**

****

**THE END**