BIRZEIT UNIVERSITY

Department of Electrical and Computer Engineering

ENCS4310

DIGITAL SIGNAL PROCESSING (DSP)

Assignment No (1)

Prepared by:

Yafa Naji-1200708

Section: 3

Supervised by:

Dr. Qadri Mayyala

Date: 12/1/2023

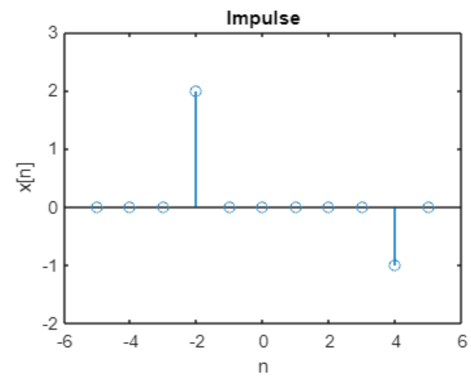
Part1

a.x1(n) = 2δ(n+2)-δ(n-4), -5≤ n ≤5

```
clc;clear all;clf;
n=-5:5;
y=2*delta_me(-2,-5,5)-delta_me(4,-5,5);
stem(n,y);
axis([-6 6 -2 3])
title("Impulse");
xlabel('n');
ylabel('x[n]');
```

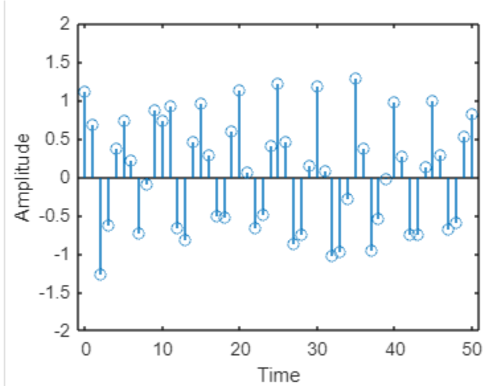
```
function [x,n]=delta_me(n0,n1,n2)
n=n1:n2;
x=[(n-n0)==0]
end
```

```
x = 1×11 logical array
    0   0   0   1   0   0   0   0   0   0   0

x = 1×11 logical array
    0   0   0   0   0   0   0   0   0   1   0
```



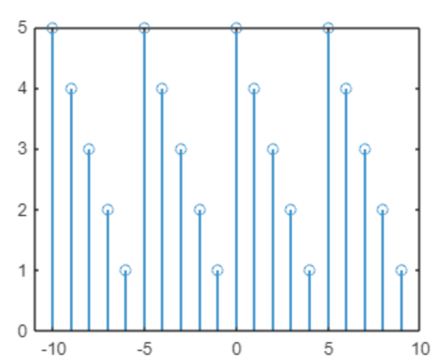
b.x3(n) = cos(0.4πn)+0.2w(n), 0≤ n ≤50, where w(n) is a Guassian random sequence with zero mean and unit variance.

```
n=[0:50];
r1=cos(0.4*pi*n);
r2=0.2*randn(1,51);
x3=r1+r2;
stem(n,x3);
axis([-1 51 -2 2])
xlabel('Time')
ylabel('Amplitude')
```



c.z [n]= [., 5,4,3,2,1,5,4,3,2,1,5,4,3,2,1, ..] ; -10<=n<=9

```
z=[5 4 3 2 1];
z=[z z z z];
n=[-10:1:9];
stem(n,z)
xlim([-11 10])
```



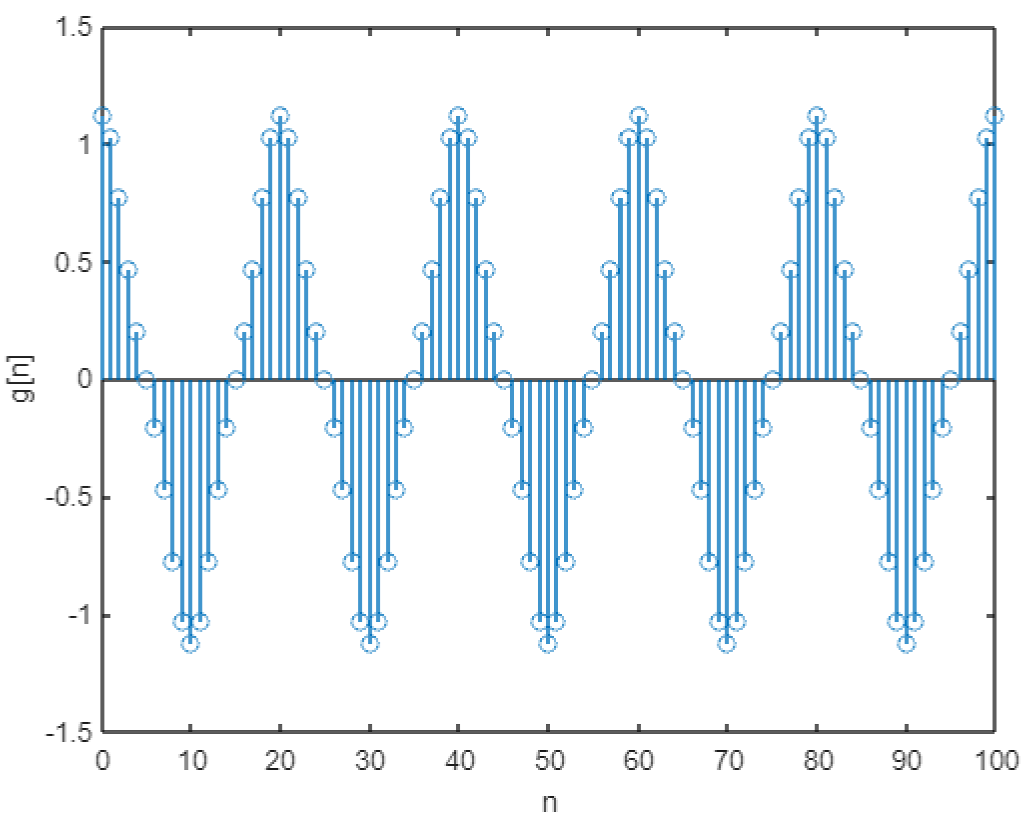
Q2. Generate and plot each of the following sequences over the indicated interval.

$g(t) = cos(2\pi F_1 t) + 0.125cos(2\pi F_2 t), F_1 = 5Hz, F_2 = 15,$, plot g[n] for one second.

A) For $Fs = 50Hz$
B) For $Fs = 30Hz$
C) For $Fs = 20Hz$

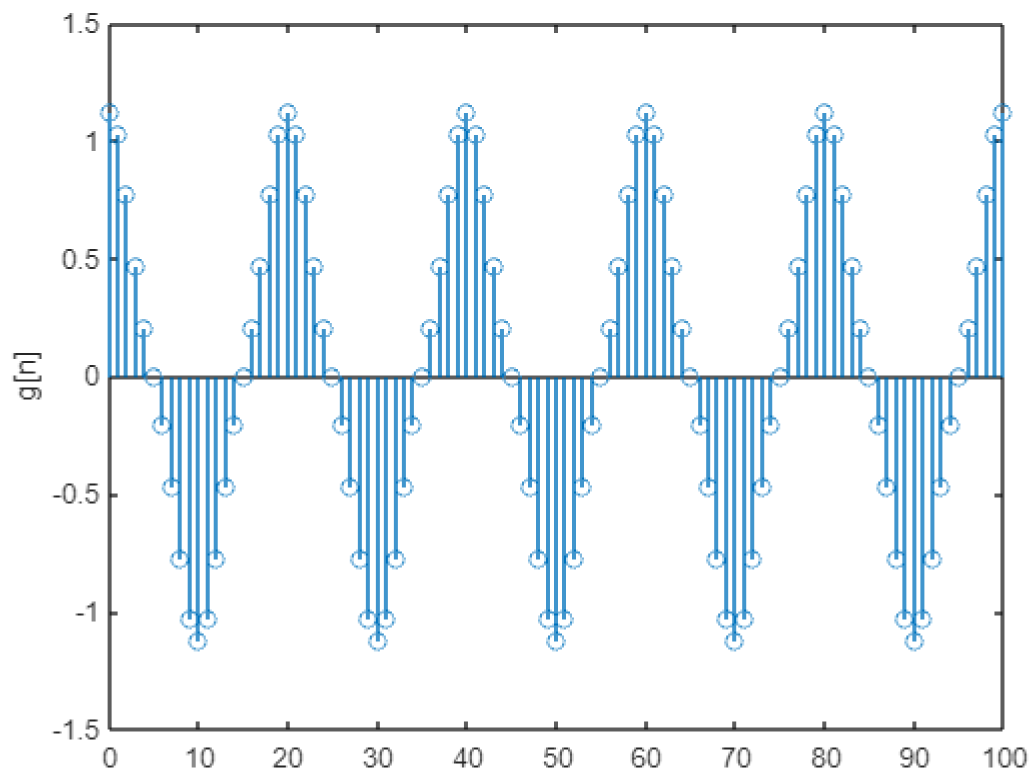
A)

```
t=0:1/100:1;
fs=50;
f1=5;
f2=15;
n=0:length(t)-1;
g=cos(2*pi*f1*t) + 0.125*cos(2*pi*f2*t);
stem(n,g);
xlabel('n');
ylabel('g[n]');
```



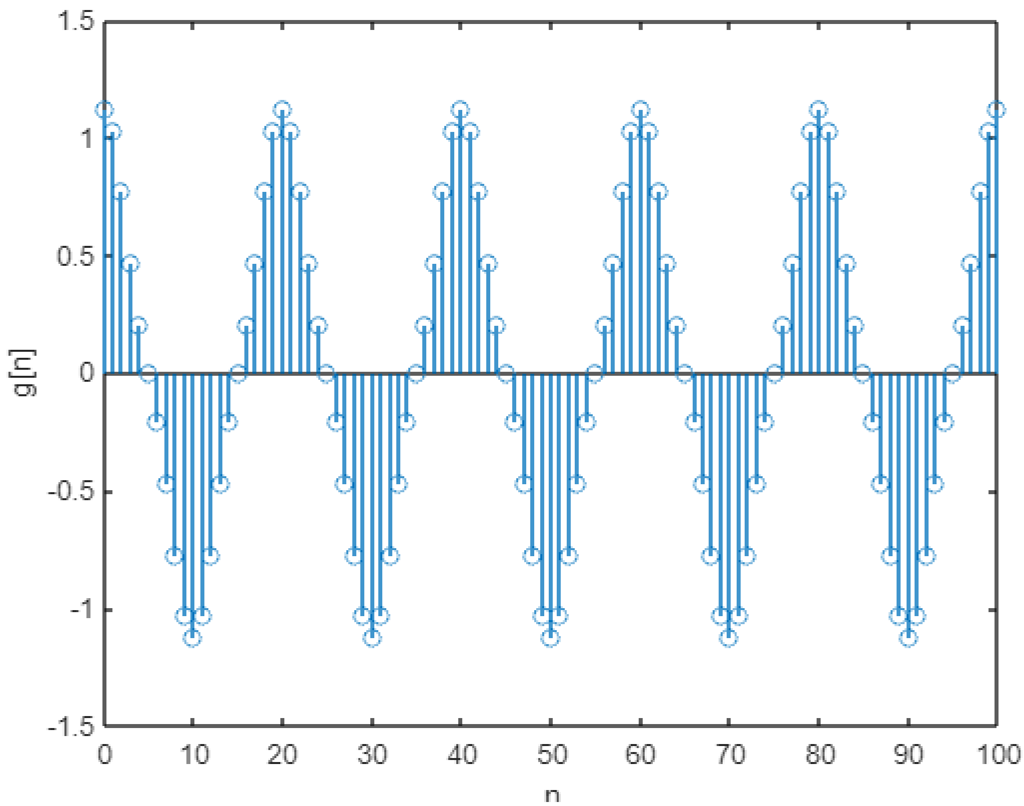
B)

```
t=0:1/100:1;
fs=30;
f1=5;
f2=15;
n=0:length(t)-1;
g=cos(2*pi*f1*t) + 0.125*cos(2*pi*f2*t);
stem(n,g);
xlabel('n');
ylabel('g[n]');
```



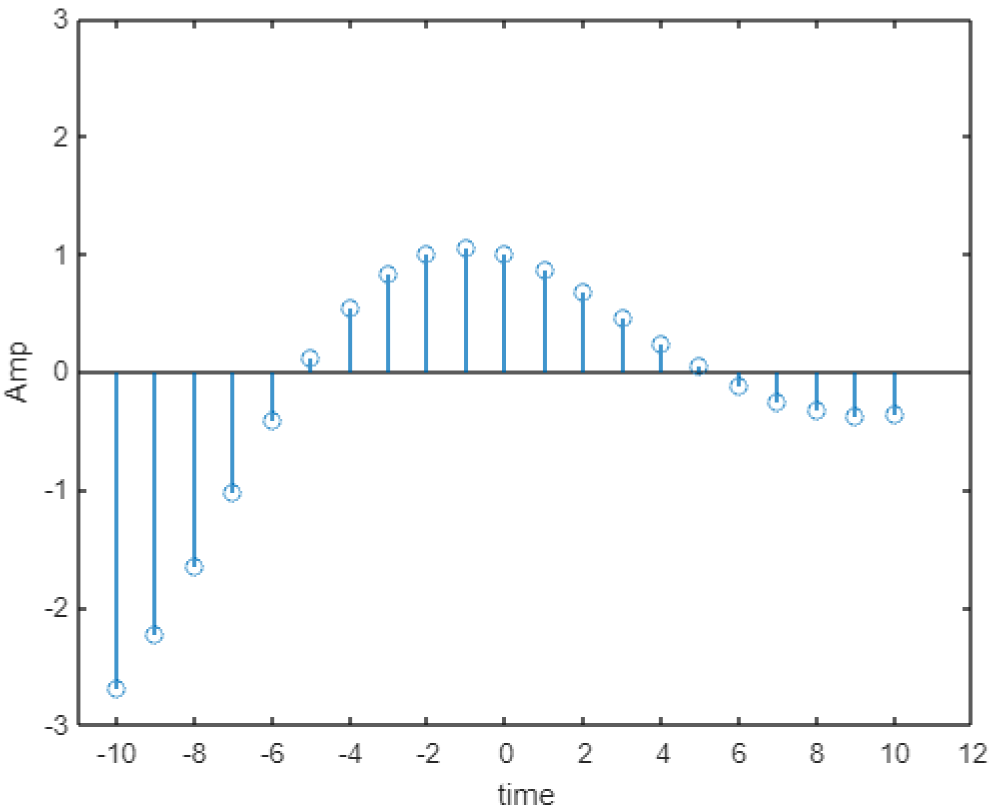
C)

```
t=0:1/100:1;
fs=20;
f1=5;
f2=15;
n=0:length(t)-1;
g=cos(2*pi*f1*t) + 0.125*cos(2*pi*f2*t);
stem(n,g);
xlabel('n');
ylabel('g[n]');
```



Q3 x[n] = exp(-0.1+j0.3)n, -10≤ n ≤10

```
n=[-10:10];
x=exp(-0.1+0.3i).^n;
stem(n,x);
axis([-11 12 -3 3]);
xlabel('time');
ylabel('Amp');
```



Part2

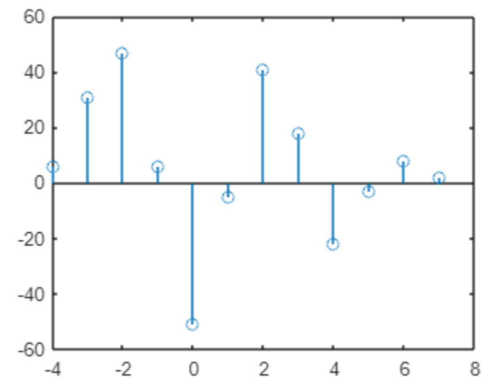
Q5. For

$$x[n] = [3, 11, 7, 0, -1, 4, 2], \ -3 \le n \le 3;$$

$$h[n] = [2, 3, 0, -5, 2, 1], \ -1 \le n \le 4$$

Find and plot y[n].

```
n=[-4:1:7];
x=[3, 11, 7, 0 , -1, 4, 2];
h=[2, 3 , 0, -5, 2, 1];
y=conv(x,h);
stem(n,y)
```



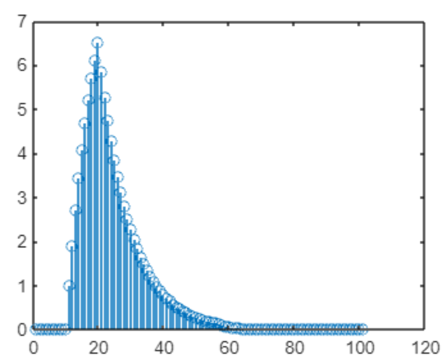
Q6. Let the rectangular pulse x(n) = u(n) − u(n − 10) be an input to an LTI system with impulse response $h[n] = (0.9)^n u(n)$

Plot x{n], h[n], Find and plot the output y(n). Consider the interval [−5, 45].

```
[a,n]=stepseq(0,-5,45);
[b,n]=stepseq(10,-5,45);
x=a-b;
h=(0.9).^n .*a;
y=conv(x,h)
stem(y)
```

```
function [x,n] = stepseq(n0,n1,n2)


if ((n0 < n1) | (n0 > n2) | (n1 > n2))
    error('arguments must satisfy n1 <= n0 <= n2')
else
n = [n1:n2];

x = [(n-n0) >= 0];
end
end
```

```
y = 1×101
       0        0        0        0        0 ...
```



Q7. To demonstrate one application of the crosscorrelation sequence.

Let $x[n] = [3, 11, 7, \underline{0}, -1, 4, 2]$ be a prototype sequence,

A) let y(n) be its noise-corrupted-and-shifted version

$$y[n]=x[n-2]+w[n]$$

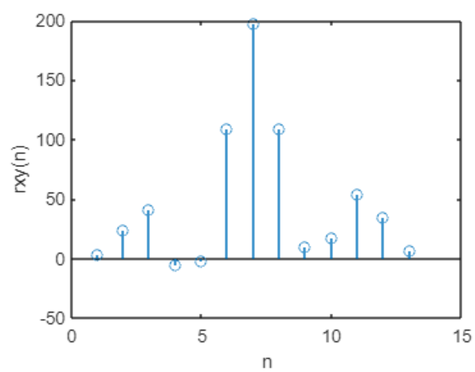
where w[n] is Gaussian sequence with mean 0 and variance 1. Compute the crosscorrelation between y[n] and x[n] and comment on the results.

B) Repeat part (a) for $y[n]=x[n-4]+w[n]$

```
n=[-3,-2,-1,0,1,2,3]
x=[3, 11, 7, 0 ,-1, 4, 2];
y=sigshift(x,n,2)+randn(1,7);
rxy=xcorr(x,y)
stem(rxy);
xlabel('n');
ylabel('rxy(n)');
```

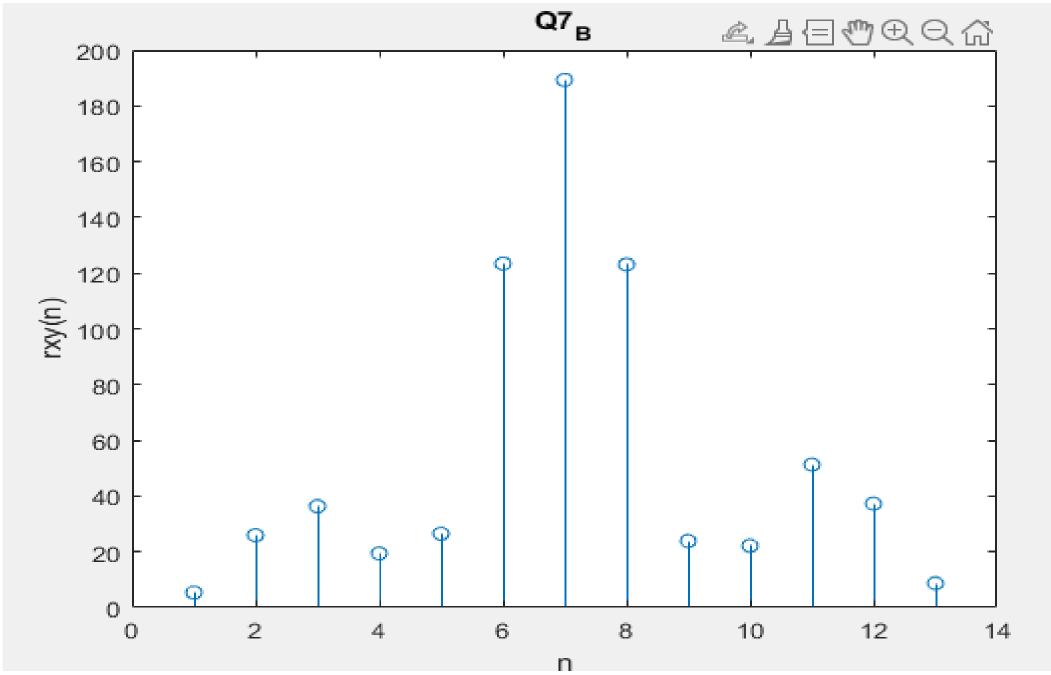
```
function [y,n] = sigshift(x,m,n0)
% implements y(n) = x(n-n0)
% -----------------------
% [y,n] = sigshift(x,m,n0)
%
n = m+n0; y = x;
end
```

```
n = 1×7
    -3    -2    -1     0     1     2     3

rxy = 1×13
    3.5059   23.9980   40.7876   -5.0379   -2.2334 ...
```



B)

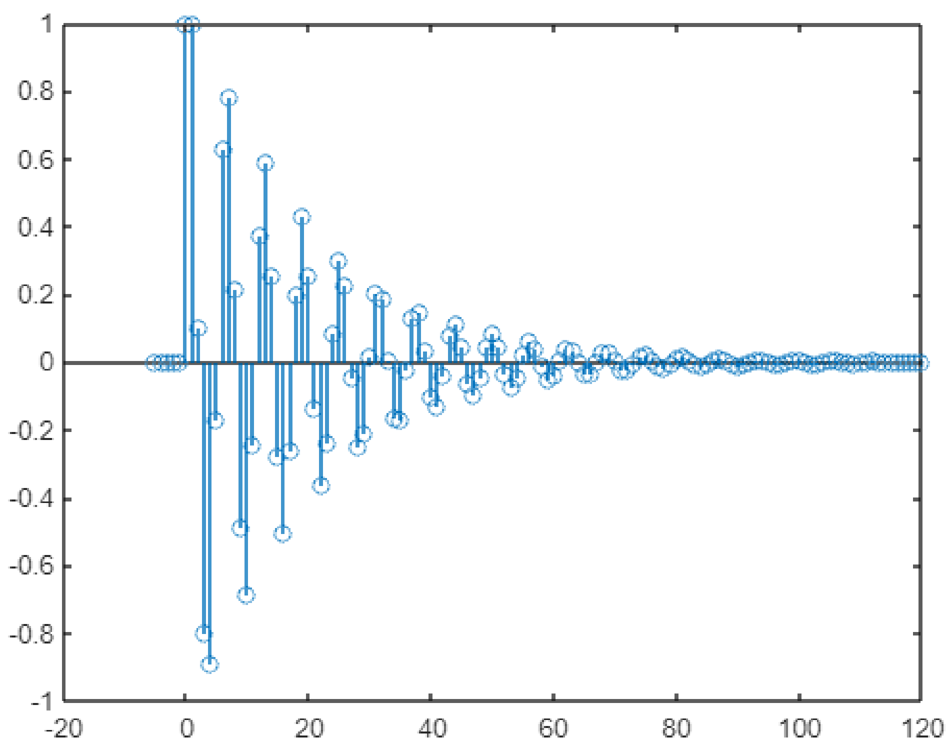
```
n = [-3, -2 ,-1, 0, 1 ,2, 3]
x = [3 ,11, 7, 0, -1, 4, 2]
y = sigshift(x,n,4) + randn(1, 7);
rxy = xcorr(x, y);

stem(rxy);
xlabel('n');
ylabel('rxy(n)');
title('Q7_B');
function [y,n]= sigshift(x,m,n0);
n=m+n0;
y=x;
end
```



Q7$_B$
200
180
160
140
120
100
80
60
40
20
0
rxy(n)
0
2
4
6
8
10
12
14
n

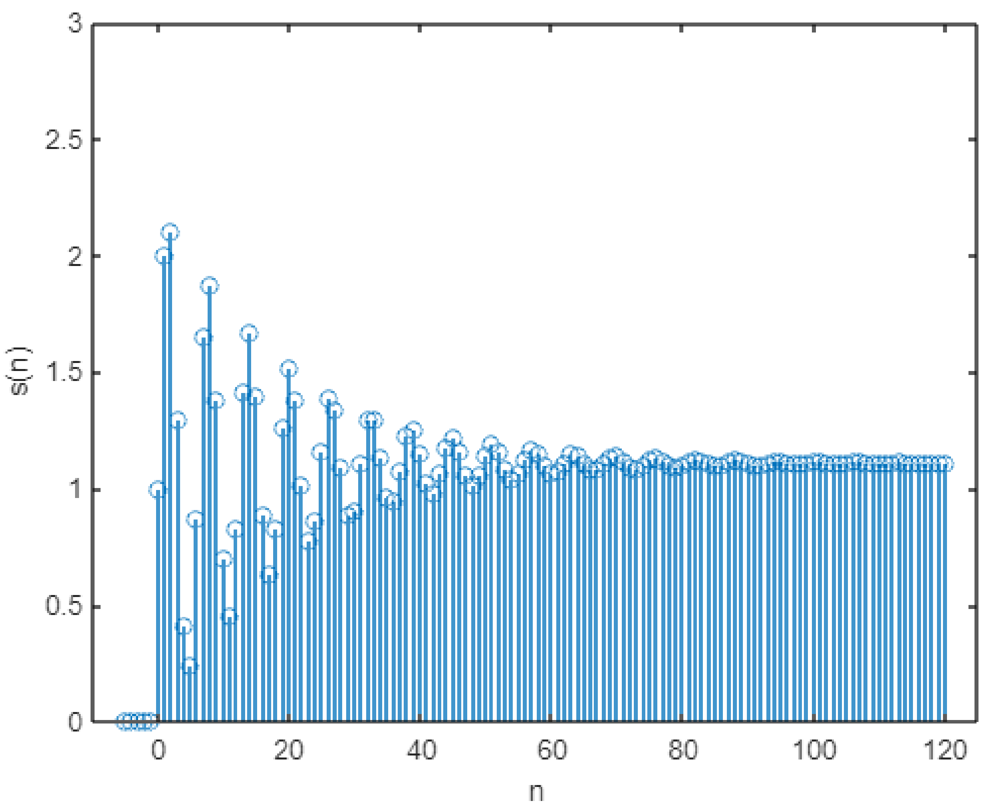
Q8:A

```
[x,n]=impseq(0,-5,120);
b=1;
a=[1 -1 0.9];
h=filter(1,a,x)
stem(n,h);
axis([ -1    0 1 ]);
xlabel('n');
ylabel('h(n)');
function [x, n] = impseq(delta, n_start, n_end)
n=n_start: n_end;
x=[(n-delta)==0]
end
```



B

```
[x,n]=stepseq(0,-5,120);
b=1;
a=[1 -1 0.9];
s=filter(1,a,x)
stem(n,s);
axis([-10 125 0 3]);
xlabel('n');
ylabel('s(n)');
sum_abs_s = sum(abs(s));
function [x,n] = stepseq(n0,n1,n2)
if ((n0 < n1) | (n0 > n2) | (n1 > n2))
      error('arguments must satisfy n1 <= n0 <= n2')
end
n = [n1:n2];
x = [(n-n0) >= 0];
end
```



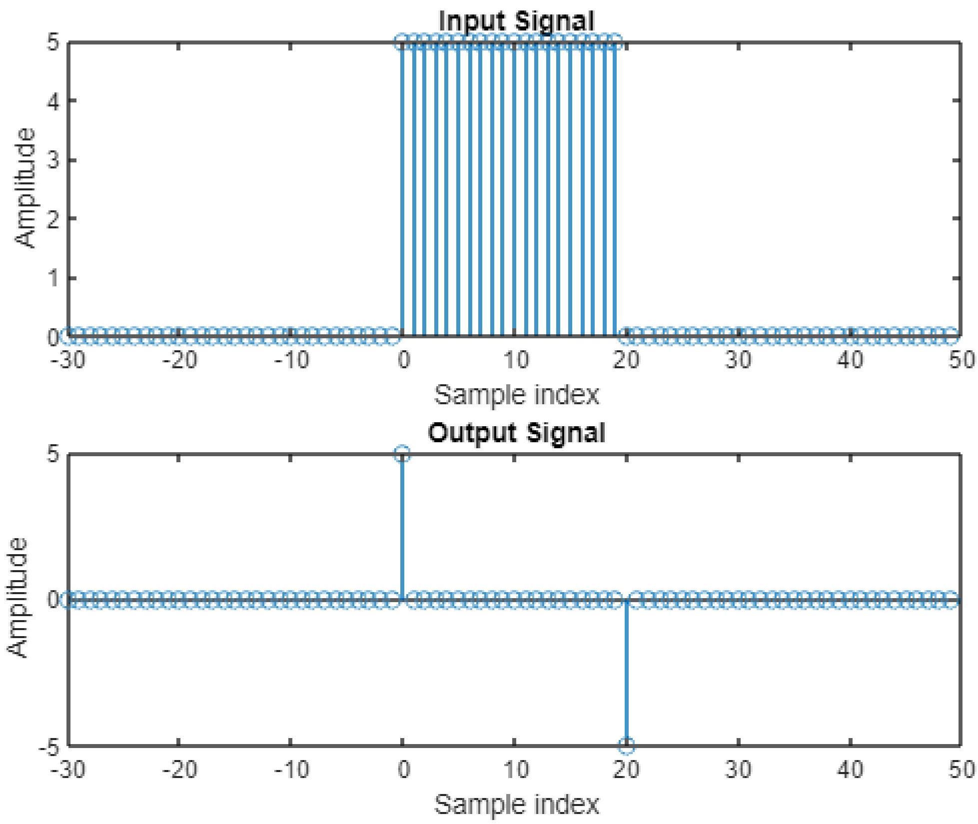
C) Is the system specified by h(n) stable?

Ans:

yes, and from the above graph , it has final value ≈0 as n goes to infinite

Q9.A

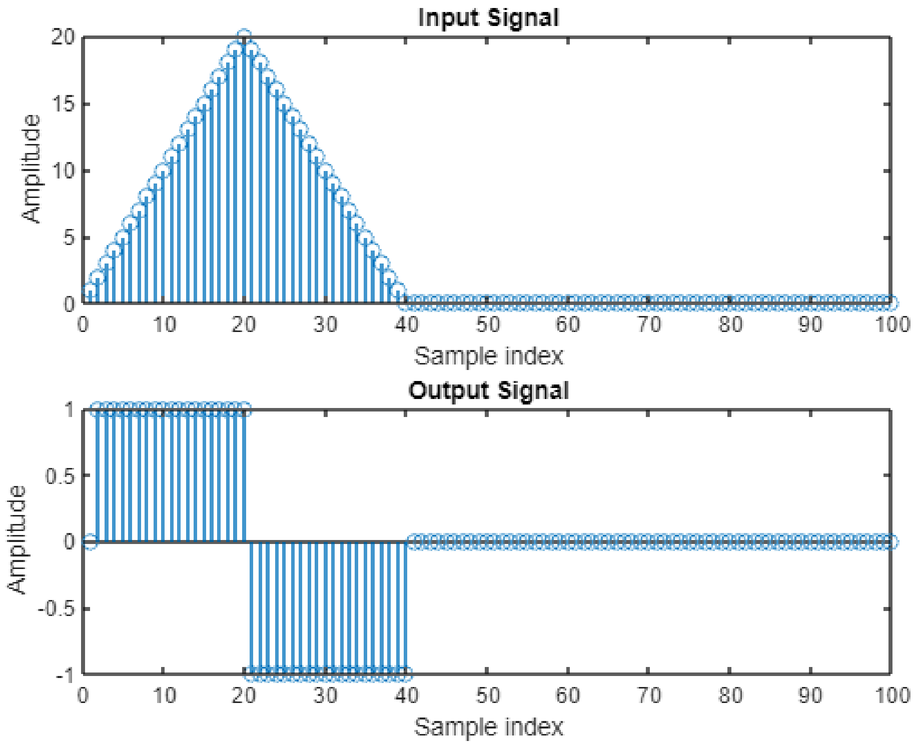
```
n=-30:49;
x = 5*((n>=0)-(n>=20));
y = zeros(size(n));
for i = 2:length(n)
y(i) = x(i) - x(i-1);
end
subplot(2,1,1);
stem(n,x);
title('Input Signal');
xlabel('Sample index');
ylabel('Amplitude');
subplot(2,1,2);
stem(n,y);
title('Output Signal');
xlabel('Sample index');
ylabel('Amplitude');
```



Q9 .B

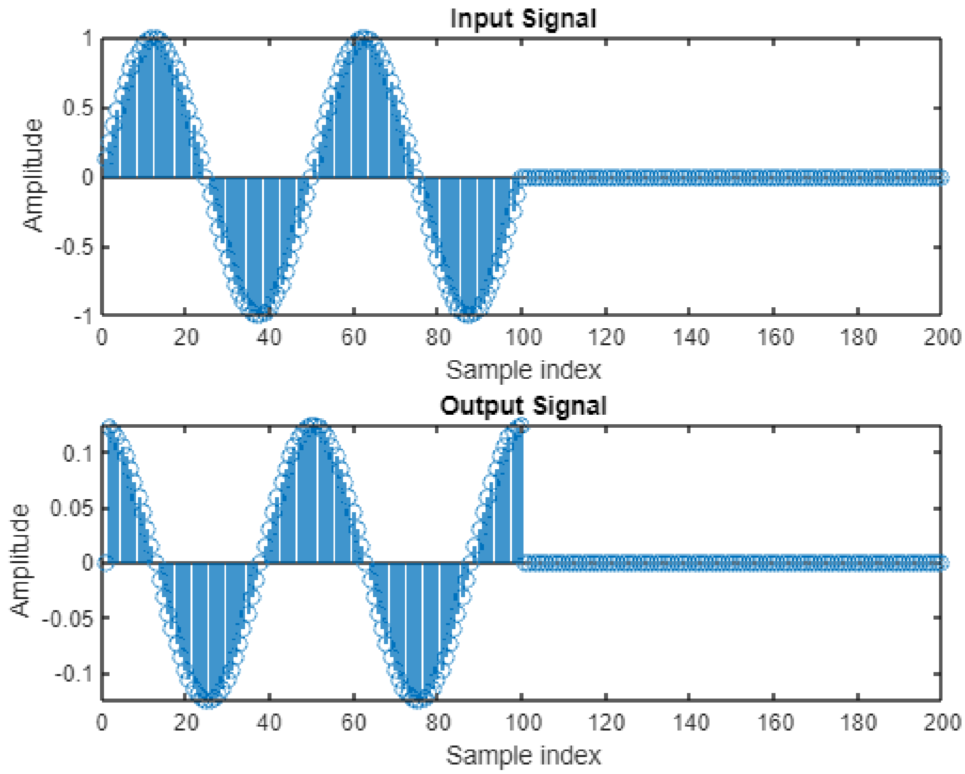
```
L = 100;
x = zeros(1, L);
for n = 1:20
    x(n) = n;
end
for n = 21:40
    x(n) = 40 - n;
end
y = zeros(1, L);
for n = 2:L
    y(n) = x(n) - x(n-1);
end
subplot(2,1,1);
stem(x);
title('Input Signal');
xlabel('Sample index');
ylabel('Amplitude');
subplot(2,1,2);
stem(y);
title('Output Signal');
```

```
xlabel('Sample index');
ylabel('Amplitude');
```



Q9.C

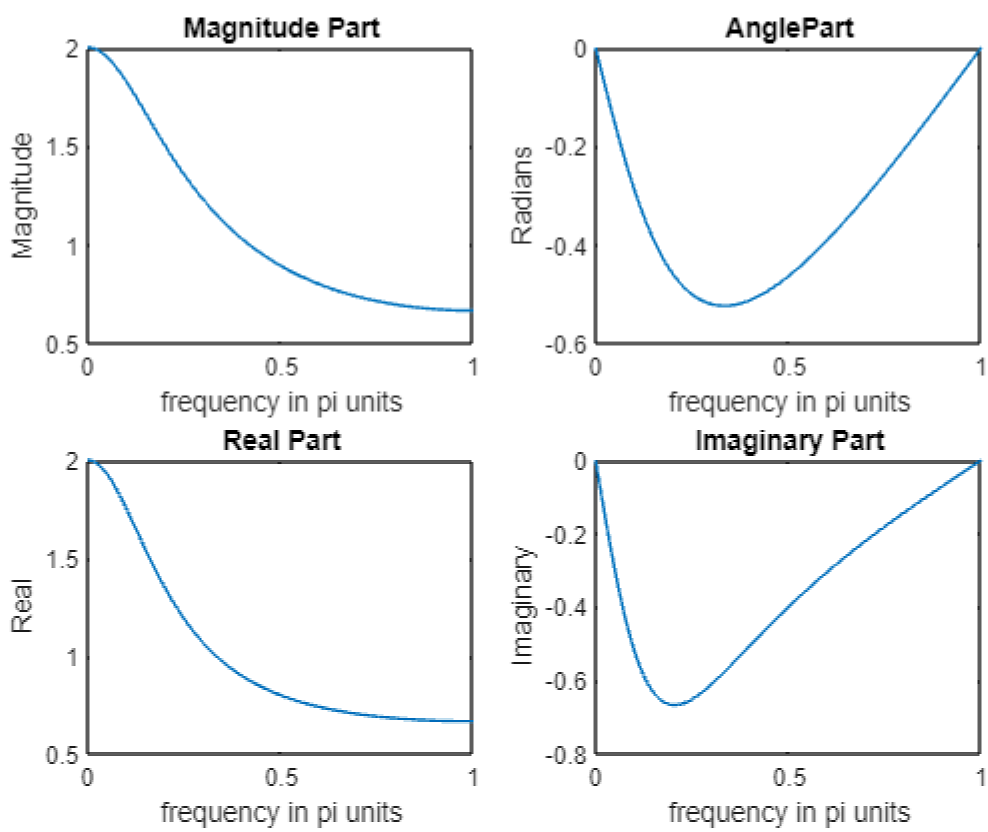
```
L = 200;
x = zeros(1, L);
for n = 1:100
    x(n) = sin(pi*n/25);
end
y = zeros(1, L);
for n = 2:L
    y(n) = x(n) - x(n-1);
end
subplot(2,1,1);
stem(x);
title('Input Signal');
xlabel('Sample index');
ylabel('Amplitude');
subplot(2,1,2);
stem(y);
title('Output Signal');
xlabel('Sample index');
ylabel('Amplitude');
```



Q10

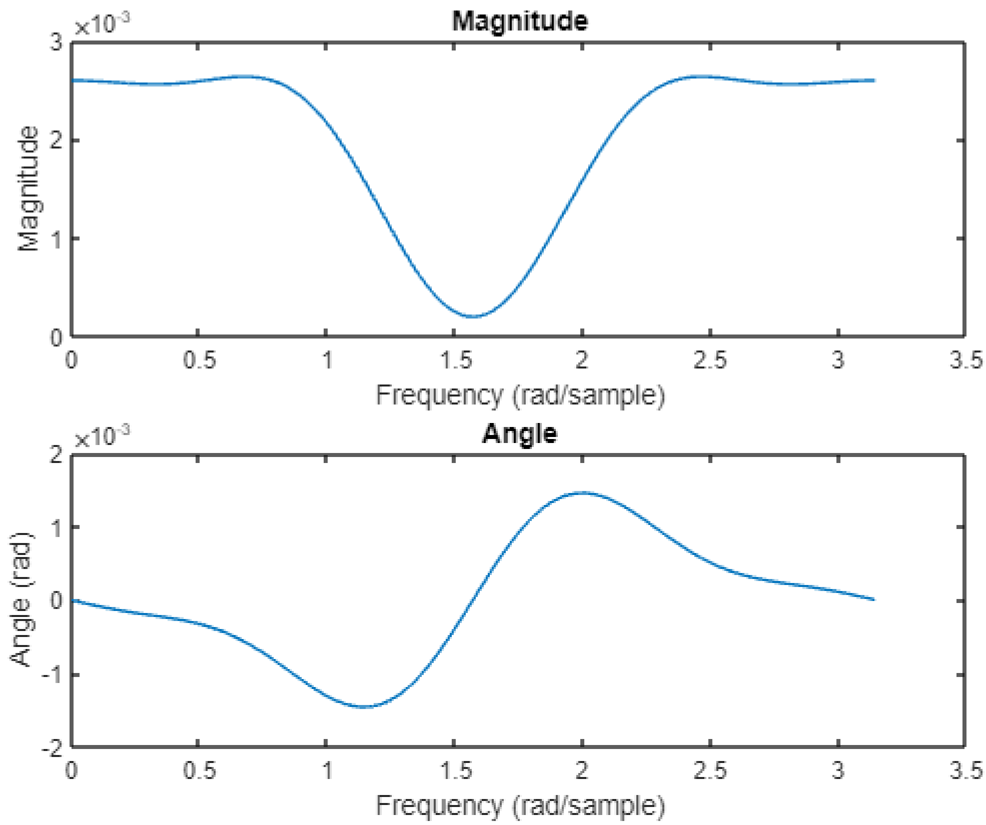
```
w = [0:1:500]*pi/500;
X = exp(j*w) ./ (exp(j*w) - 0.5*ones(1,501));
magX = abs (X);
angX = angle (X) ;
realX = real(X);
imagX = imag(X);
subplot(2,2,1);
plot(w/pi,magX);
xlabel('frequency in pi units');
title('Magnitude Part');
ylabel('Magnitude')
subplot(2,2,2);
plot(w/pi,angX);
xlabel('frequency in pi units');
title('AnglePart');
ylabel('Radians')
subplot(2,2,3);
plot(w/pi,realX);
xlabel('frequency in pi units');
```

```
title('Real Part');
ylabel('Real')
subplot(2,2,4);
plot(w/pi,imagX);
xlabel('frequency in pi units');
title('Imaginary Part');
ylabel('Imaginary');
```



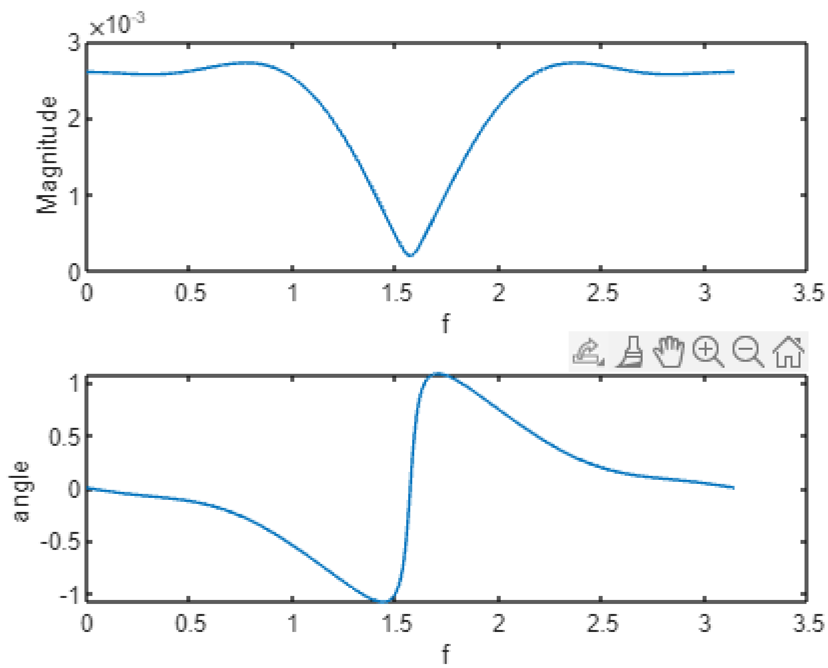
Q11.A

```
x=[1,-0.5,-0.3,-0.1];
omega = linspace(0, pi, 501);
X = fftshift(fft(x, length(omega)))/length(omega);
subplot(2,1,1);
plot(omega, real(X));
title('Magnitude');
xlabel('Frequency (rad/sample)');
ylabel('Magnitude');
subplot(2,1,2);
plot(omega, imag(X));
title('Angle');
xlabel('Frequency (rad/sample)');
```



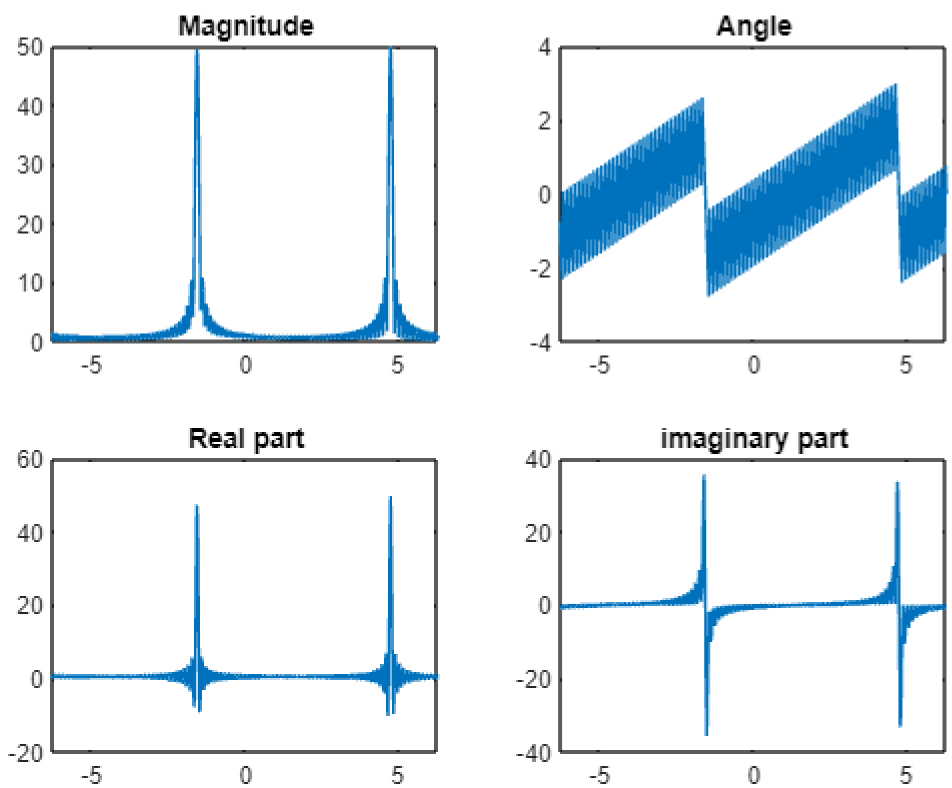
B)

```
x=[1,-0.5,-0.3,-0.1];
omega = linspace(0, pi, 501);
X = fftshift(fft(x, length(omega)))/length(omega);
subplot(2,1,1);
plot(omega, abs(X));
title('Magnitude');
xlabel('Frequency (rad/sample)');
ylabel('Magnitude');
subplot(2,1,2);
plot(omega, angle(X));
title('Angle');
xlabel('Frequency (rad/sample)');
ylabel('Angle (rad)');
```



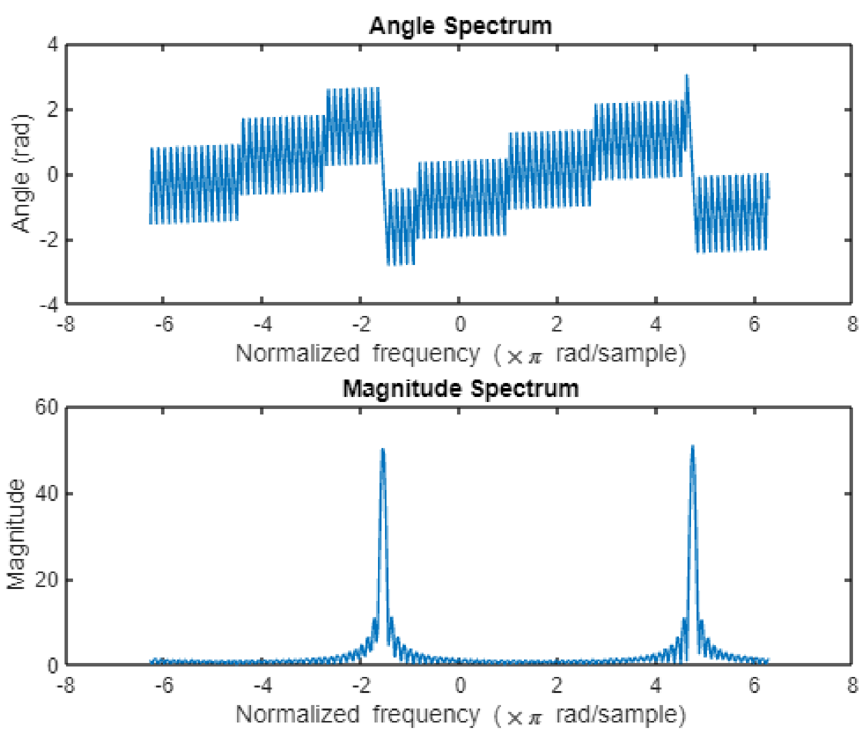
Q12.A

```
L=100;
x = cos(pi*(0:L-1)/2);
y = exp(j*pi*(0:L-1)/4) .* x;
N = 401;
frequencies = linspace(-2*pi,2* pi, N);
H = fft(y, length(frequencies));
subplot(2,2,1);
plot(frequencies, abs(H));
title('Magnitude');
subplot(2,2,2);
plot(frequencies, angle(H));
title('Angle');
subplot(2,2,3);
plot(frequencies, real(H));
title('Real part');
subplot(2,2,4);
plot(frequencies, imag(H));
title('imaginary part');
```



Q12.B

```
n = 0:100;
x = cos(pi*n/2);
% Generate the signal y[n]
y = exp(j*pi*n/4) .* x;
N = 401;
frequencies = linspace(-2*pi,2* pi, N);
H = fft(y, length(frequencies));
% Plot the angle spectrum
subplot(2,1,1);
plot(frequencies, angle(H));
title('Angle Spectrum');
xlabel('Normalized frequency (\times\pi rad/sample)');
ylabel('Angle (rad)');
% Plot the magnitude spectrum
subplot(2,1,2);
plot(frequencies, abs(H));
title('Magnitude Spectrum');
xlabel('Normalized frequency (\times\pi rad/sample)');
ylabel('Magnitude');
```



C) Comment on the relation between x[n] and y[n].

x[n] is a cosine function, which has a real part that is symmetric about the y-axis and an imaginary part that is zero. y[n] is the result of multiplying x[n] with a complex exponential function exp(j**pi**n/4). This means that y[n] is a modified version of x[n] with a phase shift of pi/4 radians. This can be seen in the time domain, where the cosine wave in x[n] has been rotated by pi/4 radians to the left in y[n]. In the frequency domain, this results in the magnitude spectrum being the same for both x[n] and y[n] but the phase spectra will be different. The phase spectrum of x[n] will be zero as it is a real signal and y[n] will have a non-zero phase spectrum as it is a complex signal.