भारतीय प्रौद्योगिकी संस्थान तिरुपति
TIRUPATI

LAB SHEET 02

NAME : MADHU SAI NAIK P **ROLL NO: EE23B039**

A. Implementation of convolution

1. Generate a random impulse response sequence h[n] , 0<=n<=m where h[n]~u[a,b].
2. Generate a random input sequence x[n], 0<=n<=m where x[n] ~ u[c,d]
3. Write a function to compute the output sequence y[n] = x[n] * h[n] = sgm k =0 to infinity (x[k] h[n-k]) for n = 0,1,2 .. n+m. Function name as conv_EE23B039.
4. Plot y[n] vs n.
5. Verify your result using conv command.
6. Repeats the exercise for a few more randomly generated x[n] and h[n].

```
%Declaring the variables
m = 7; %Max value
a = -2; % lower bound
b = 2;   % upper bound
c = -1; %lower bound
d = 1; % uppper bound
```

Question 1 :

```
h_n = a + (b-a).*rand(1,m+1);
```

Question 2:

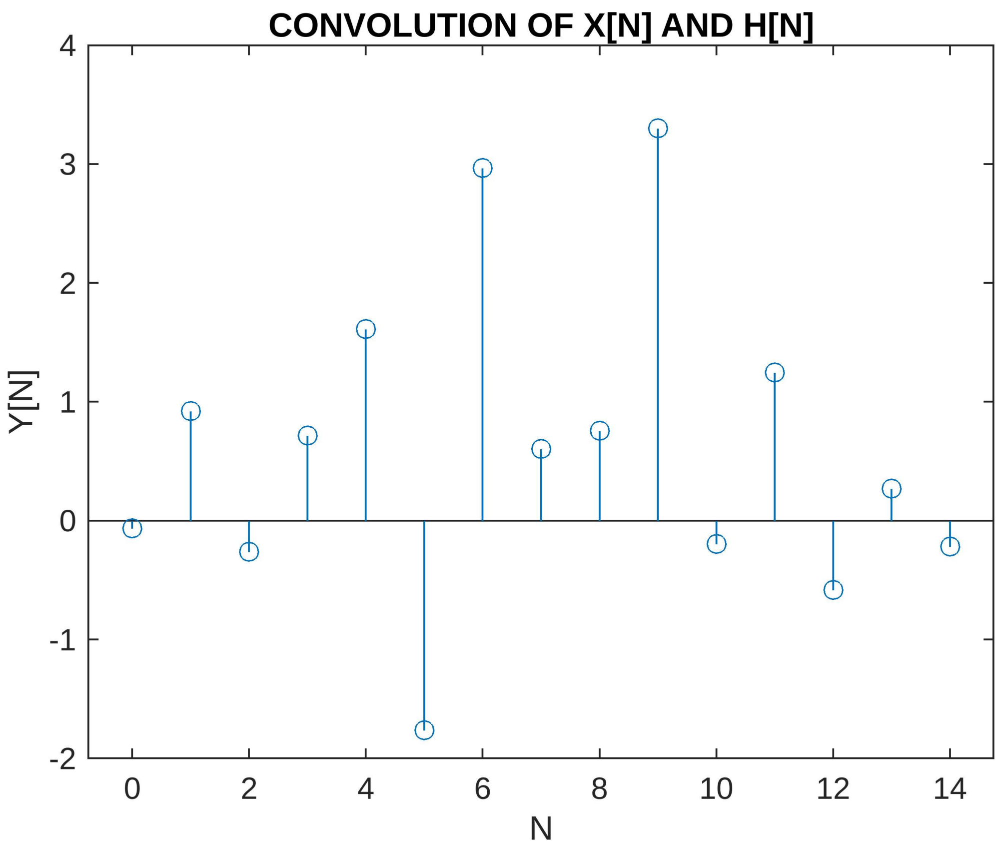
```
x_n = c + (d-c).* rand(1,m+1);
```

Question 3:

```
y_n1 = CONV_EE23B039(x_n,h_n);
```

Question 4:

```
n = 0:1:2*m;
stem(n,y_n1);
xlabel("N");
ylabel("Y[N]");
title("CONVOLUTION OF X[N] AND H[N]")
```



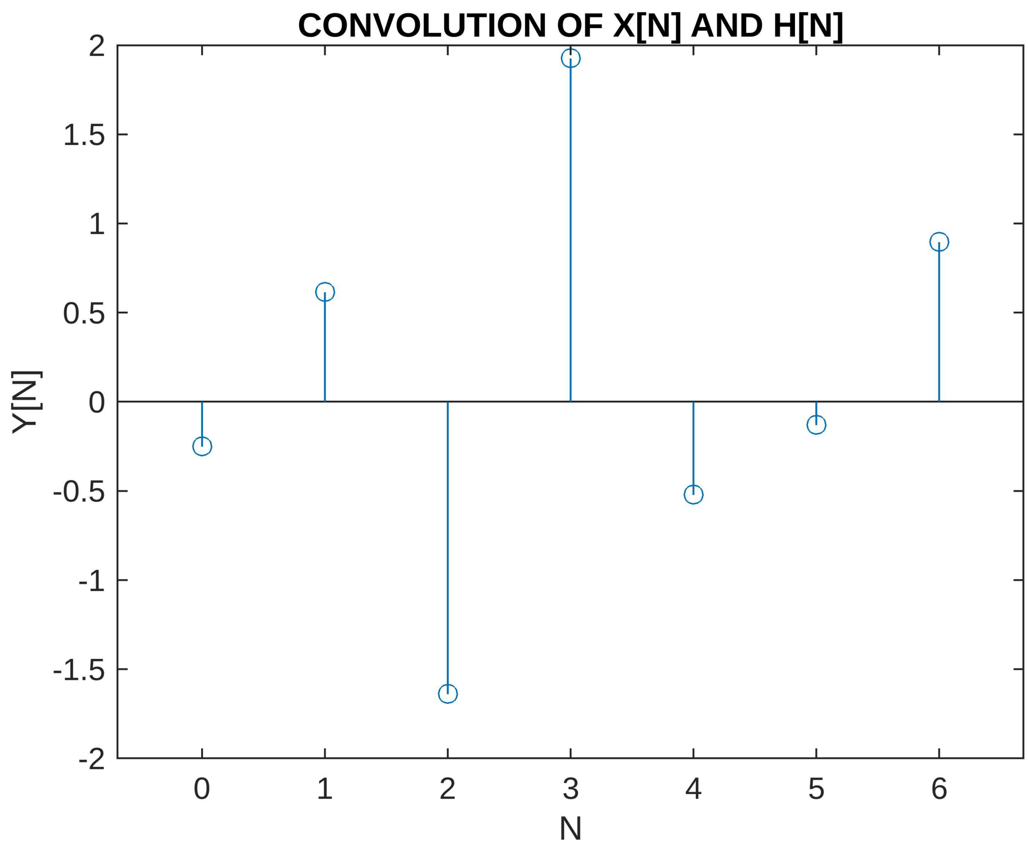
question 5:

```
y_n2 = conv(x_n,h_n);
if(y_n1 == y_n2)
    disp("True");
else
    disp("False");
end
```

```
True
```

Question 6:

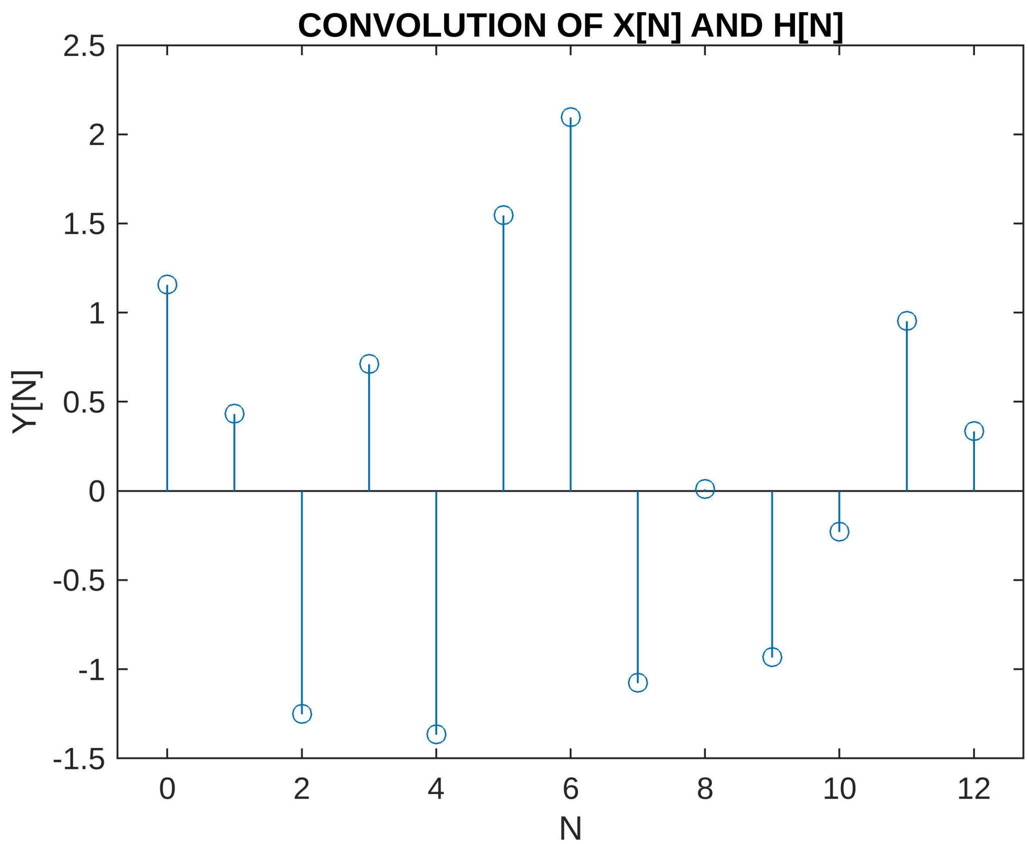
```
m = 3;
h_n = a + (b-a).*rand(1,m+1);
x_n = c + (d-c).* rand(1,m+1);
y_n1 = CONV_EE23B039(x_n,h_n);
n = 0:1:2*m;
stem(n,y_n1);
xlabel("N");
ylabel("Y[N]");
title("CONVOLUTION OF X[N] AND H[N]");
```




```
y_n2 = conv(x_n,h_n);
if(y_n1 == y_n2)
    disp("True");
else
    disp("False");
end
```

```
True
```

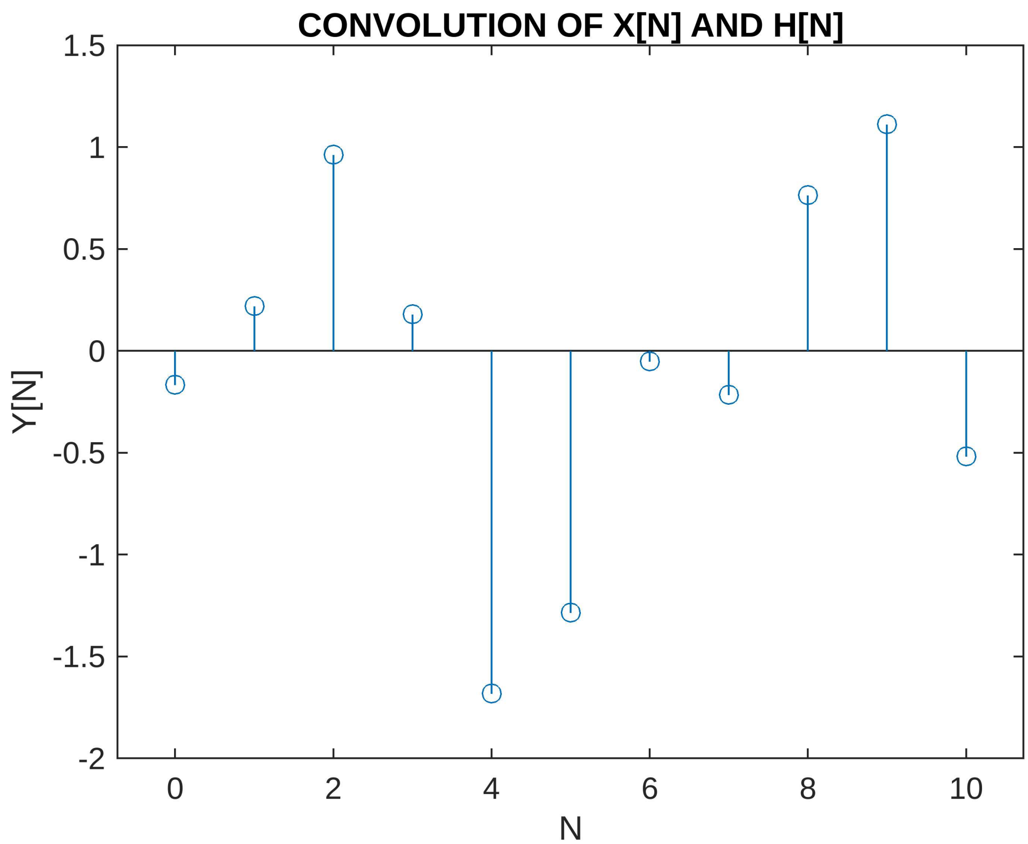
```
m = 6;
h_n = a + (b-a).*rand(1,m+1);
x_n = c + (d-c).* rand(1,m+1);
y_n1 = CONV_EE23B039(x_n,h_n);
n = 0:1:2*m;
stem(n,y_n1);
xlabel("N");
ylabel("Y[N]");
title("CONVOLUTION OF X[N] AND H[N]");
```




```
y_n2 = conv(x_n,h_n);
if(y_n1 == y_n2)
    disp("True");
else
    disp("False");
end
```

```
True
```

```
m = 5;
h_n = a + (b-a).*rand(1,m+1);
x_n = c + (d-c).* rand(1,m+1);
y_n1 = CONV_EE23B039(x_n,h_n);
n = 0:1:2*m;
stem(n,y_n1);
xlabel("N");
ylabel("Y[N]");
title("CONVOLUTION OF X[N] AND H[N]");
```




```
y_n2 = conv(x_n,h_n);
if(y_n1 == y_n2)
    disp("True");
else
    disp("False");
end
```

```
True
```

B. SYSTEMS IN PARAELL

1. Generate a random h1[n] 0<= n <=m1 and h2[n] 0<=n<m2.
2. Generate random x[n] 0<=n<=N.
3. Obatain y1[n] = x[n] * h1[n] and y2[n] = x2[n]*h2[n] using your command. compute Ya[n] = y1[n] + y2[n].
4. Simlarly obtain Yb[n] = x[n]*(h1[n]+h2[n]).
5. Compare Ya[n] with Yb[n]. Report your observations.
6. Repeat the experiements for a few values of random generated h1[n],h2[n] and x[n].

Question 1

```
m1=7;
m2=9;
h1_n = 0+m1.*rand(1,8);
h2_n = 0+m2.*rand(1,8);
```

Question 2

```
N = 15;
x_n = N.*rand(1,N+1);
```

Question 3

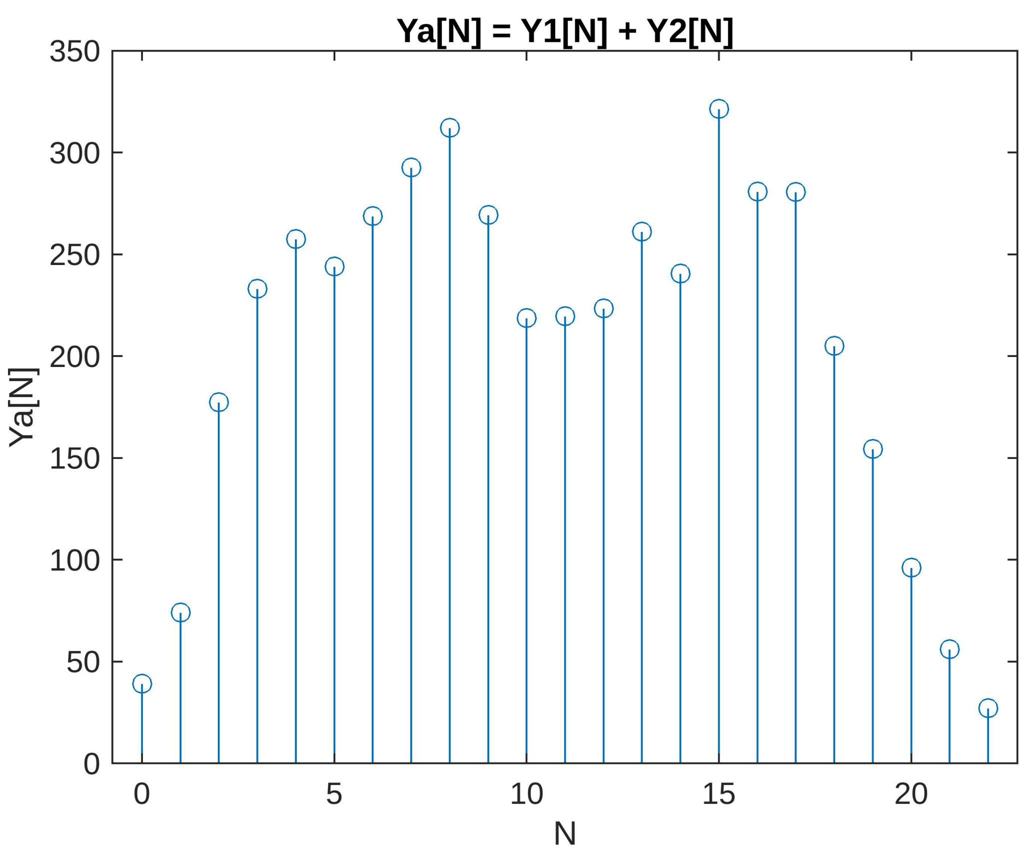
```
y1_n = conv(x_n,h1_n);
y2_n = conv(x_n,h2_n);
Ya_n = y1_n + y2_n;
```

Question 4

```
h_n = h1_n + h2_n;
Yb_n = conv(x_n,h_n);
```

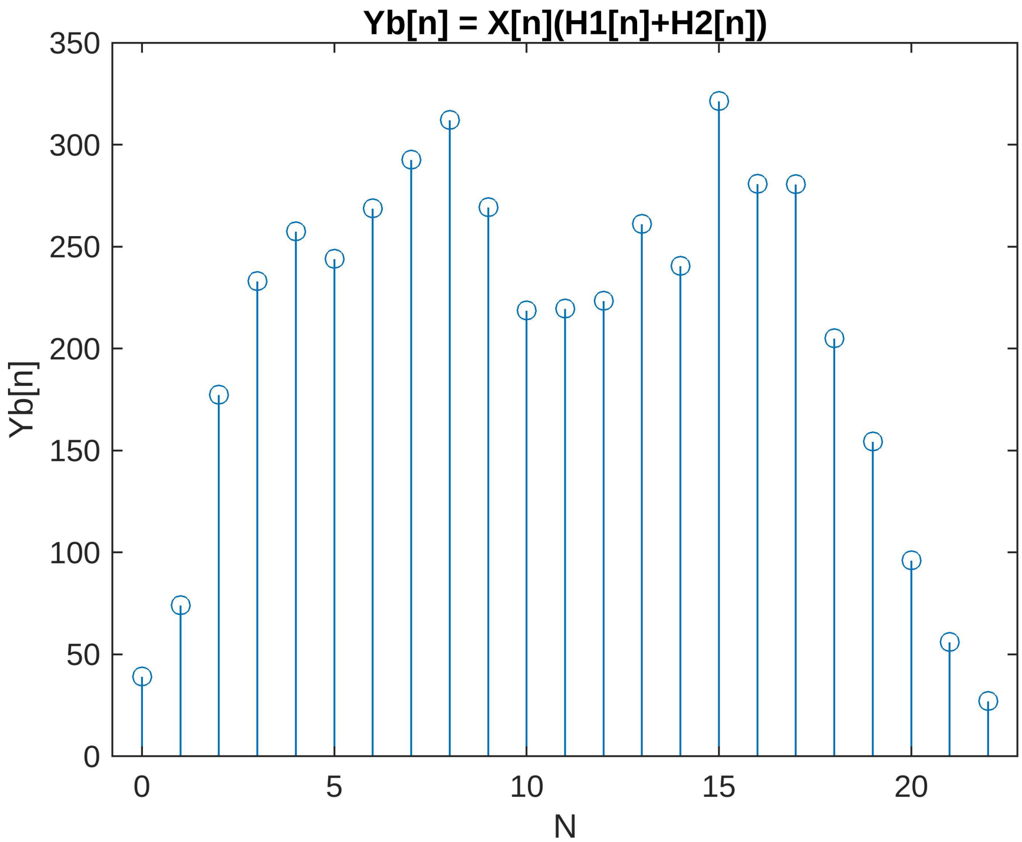
Question 5

```
n=0:1:22;
stem(n,Ya_n);
xlabel("N");
ylabel("Ya[N]");
title("Ya[N] = Y1[N] + Y2[N]");
```




```
stem(n,Yb_n);
xlabel("N");
```

```
ylabel("Yb[n]");
title("Yb[n] = X[n](H1[n]+H2[n])")
```



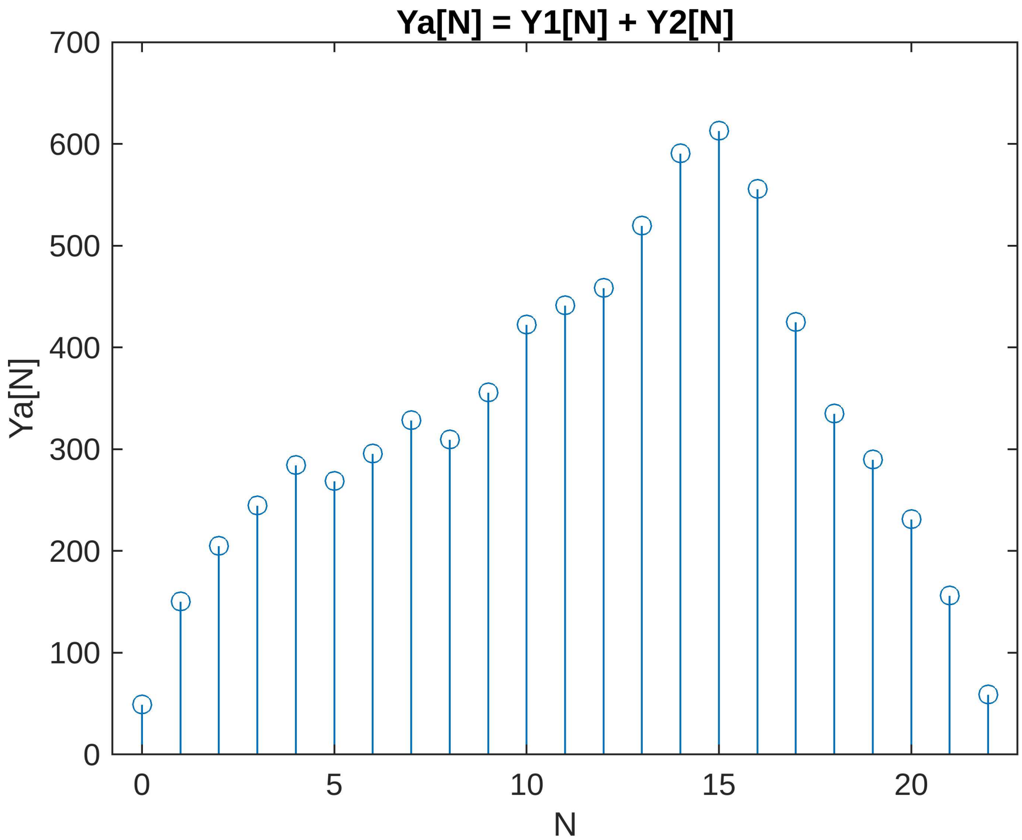

```
% As we can see both Ya[n] and Yb[n] both are same , this shows that the
% property of the system being in parallel.
```

Question 6

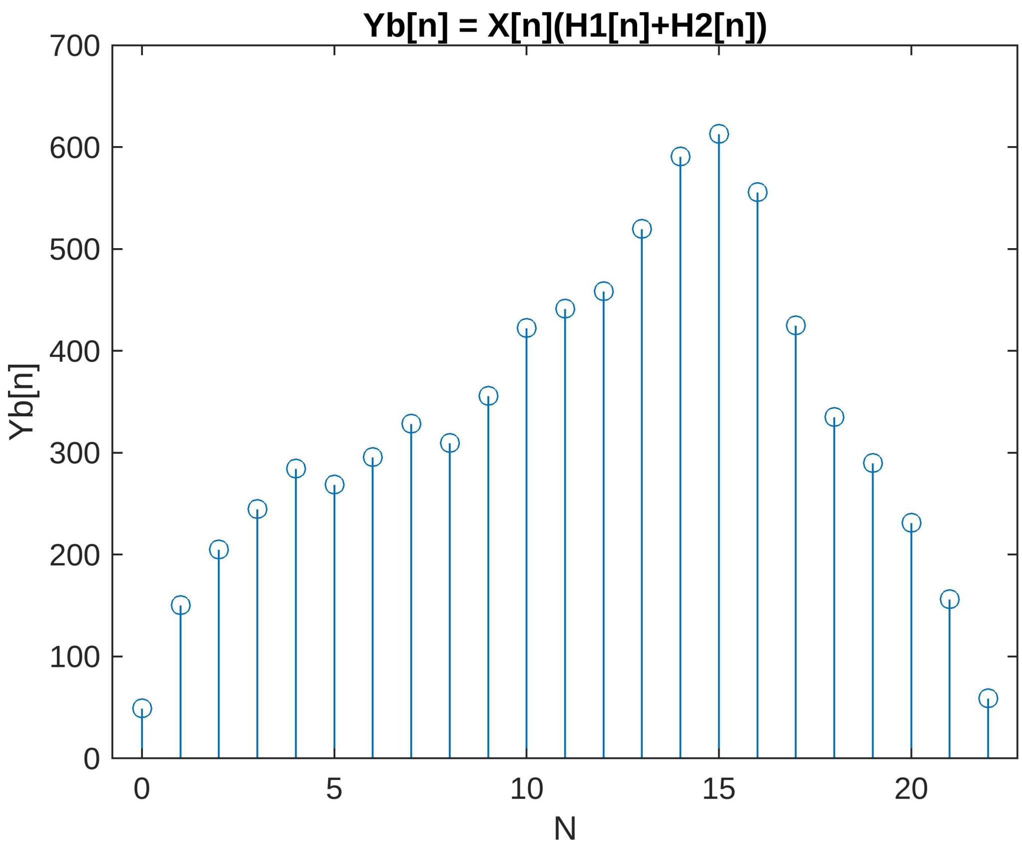
```
display("MORE EXAMPLES");
```

    "MORE EXAMPLES"

```
h1_n = 0+m1.*rand(1,8);
h2_n = 0+m2.*rand(1,8);
x_n = N.*rand(1,N+1);
y1_n = conv(x_n,h1_n);
y2_n = conv(x_n,h2_n);
Ya_n = y1_n + y2_n;
h_n = h1_n + h2_n;
Yb_n = conv(x_n,h_n);
n=0:1:22;
stem(n,Ya_n);
xlabel("N");
ylabel("Ya[N]");
title("Ya[N] = Y1[N] + Y2[N]");
```



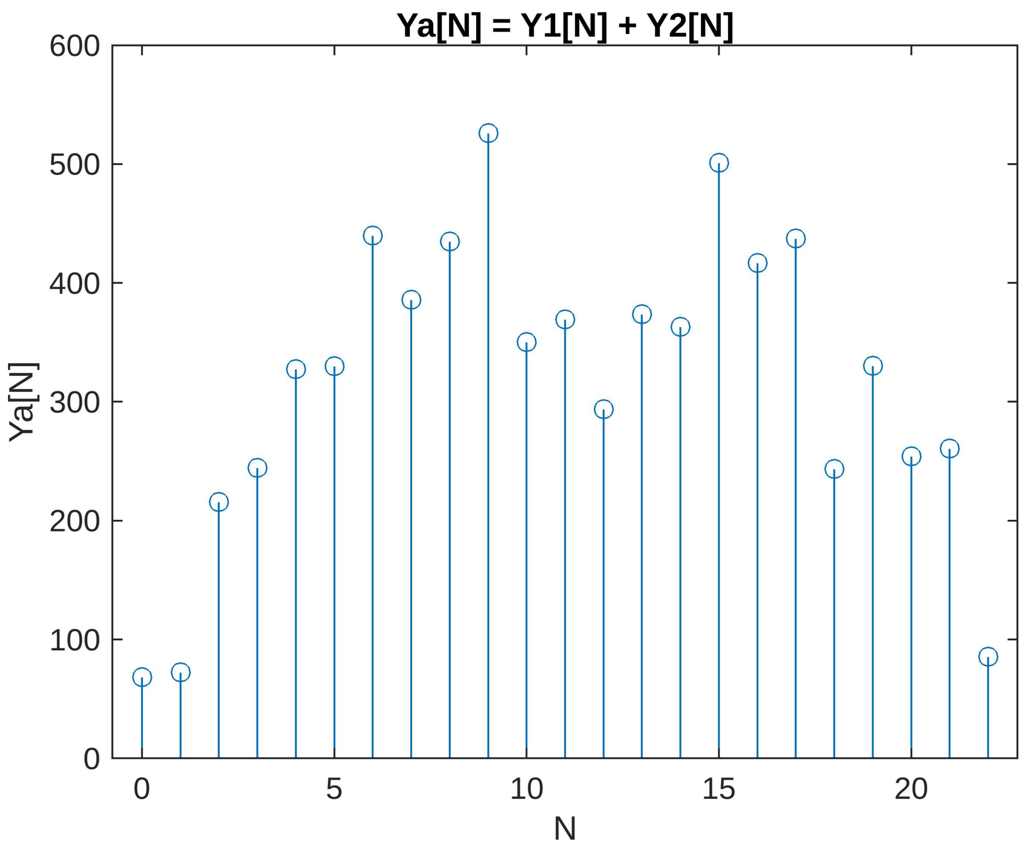

```
stem(n,Yb_n);
xlabel("N");
ylabel("Yb[n]");
title("Yb[n]  = X[n](H1[n]+H2[n])")
```



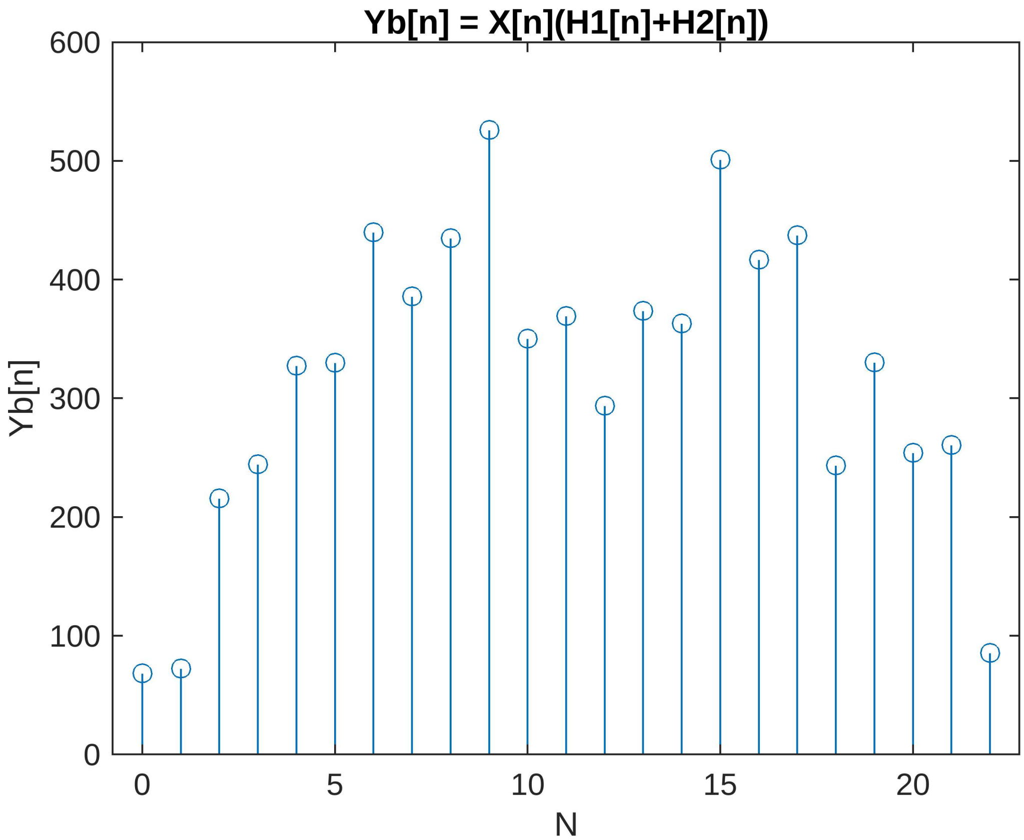

```
display("Example 2");
```

    "Example 2"

```
h1_n = 0+m1.*rand(1,8);
h2_n = 0+m2.*rand(1,8);
x_n = N.*rand(1,N+1);
y1_n = conv(x_n,h1_n);
y2_n = conv(x_n,h2_n);
Ya_n = y1_n + y2_n;
h_n = h1_n + h2_n;
Yb_n = conv(x_n,h_n);
n=0:1:22;
stem(n,Ya_n);
xlabel("N");
ylabel("Ya[N]");
title("Ya[N] = Y1[N] + Y2[N]");
```




```
stem(n,Yb_n);
xlabel("N");
ylabel("Yb[n]");
title("Yb[n] = X[n](H1[n]+H2[n])")
```



C. SYSTEMS IN CASCADE

1. Generate a random h1[n] 0<= n <=m1 and h2[n] 0<=n<m2.

2. Generate random x[n] 0<=n<=N.

3. Obatain i) x1[n] = x[n]*h1[n]

ii) Ya[n] = x1[n]*h2[n]

4. Similarly compute

i) x2[n] = x[n] * h2[n]

ii) Yb[n] = x2[n] *h1[n]

5. Finally obtain

i) h[n] = h1[n] * h2[n]

ii) Yc[n] = x[n] * h[n]

6. Compare Ya[n],Yb[n] and Yc[n]. Report your observations.

7. Repeat the experiements for a few values of random generated h1[n],h2[n] and x[n].

Question 1

```
m1=3;
m2=5;
h1_n = 0+m1.*rand(1,m1+1);
h2_n = 0+m2.*rand(1,m2+1);
```

Question 2

```
N = 10;
x_n = N.*rand(1,N+1);
```

Question 3

```
x1_n = conv(x_n,h1_n);
Ya_n = conv(x1_n,h2_n);
```

Question 4

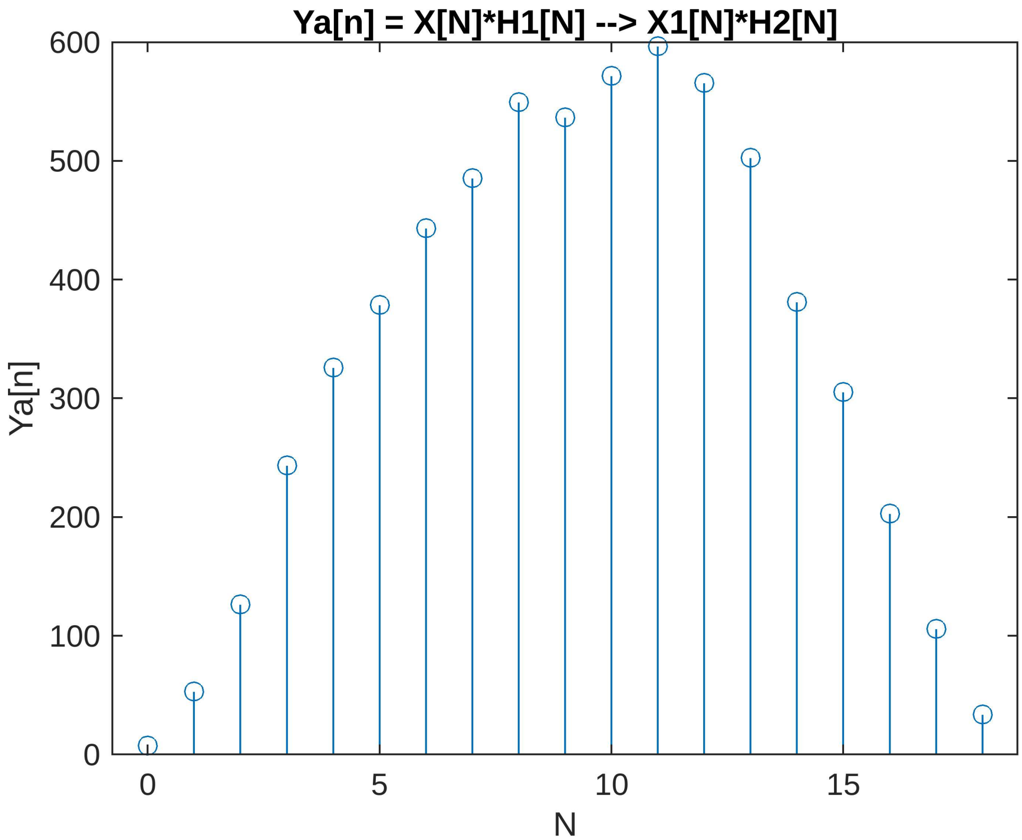
```
x2_n = conv(x_n,h2_n);
Yb_n = conv(x2_n,h1_n);
```

Question 5

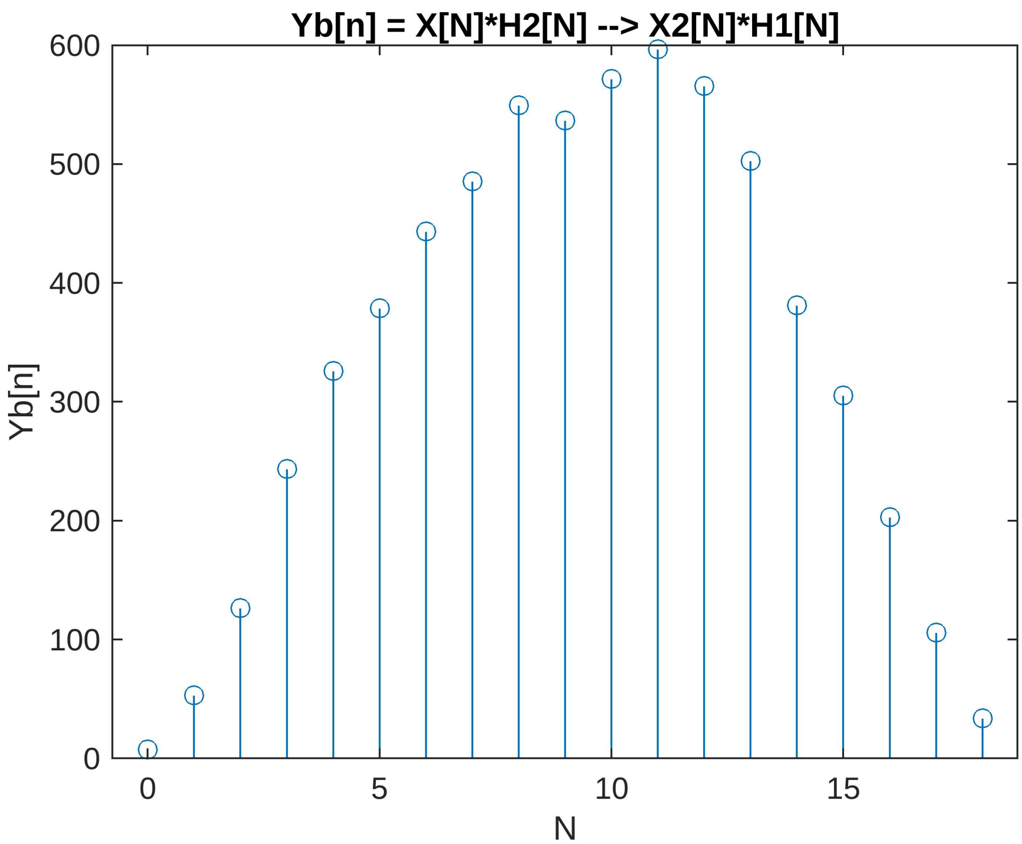
```
h_n = conv(h1_n,h2_n);
Yc_n = conv(x_n,h_n);
```

Question 6

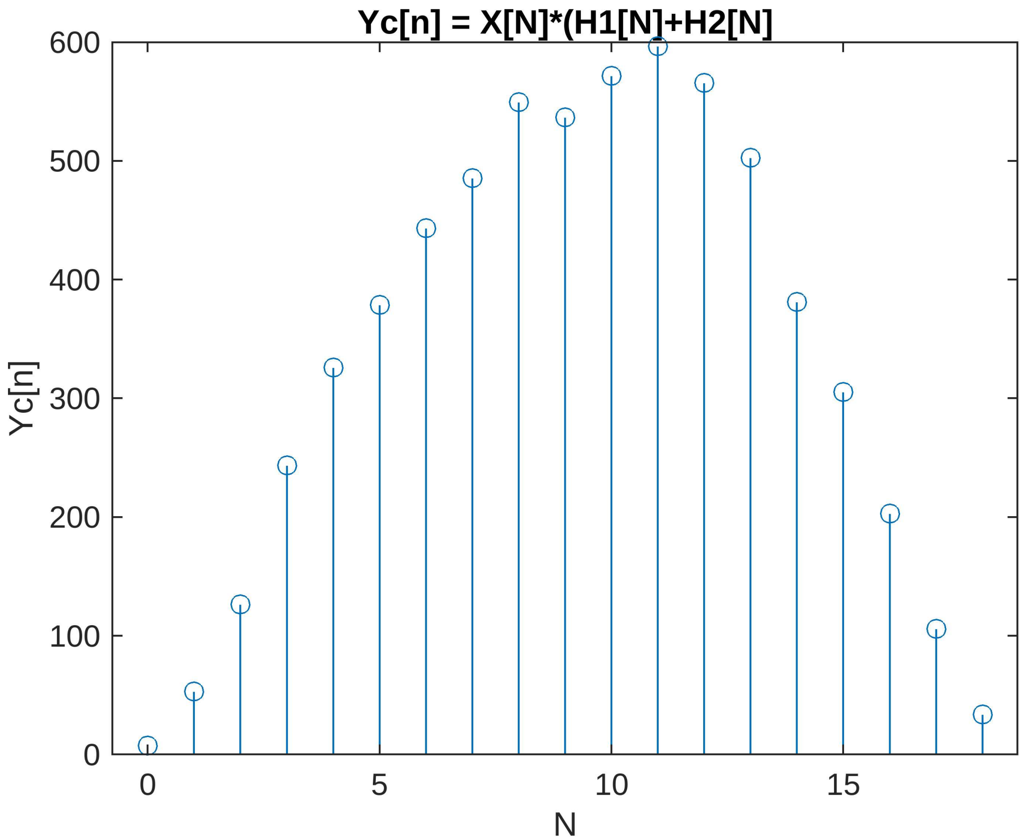
```
n = 0:1:18;
stem(n,Ya_n);
xlabel("N");
ylabel("Ya[n]");
title("Ya[n] = X[N]*H1[N] --> X1[N]*H2[N]");
```




```
stem(n,Yb_n);
xlabel("N");
ylabel("Yb[n]");
title("Yb[n] = X[N]*H2[N] --> X2[N]*H1[N]");
```




```
stem(n,Yc_n);
xlabel("N");
ylabel("Yc[n]");
title("Yc[n] = X[N]*(H1[N]+H2[N]");
```



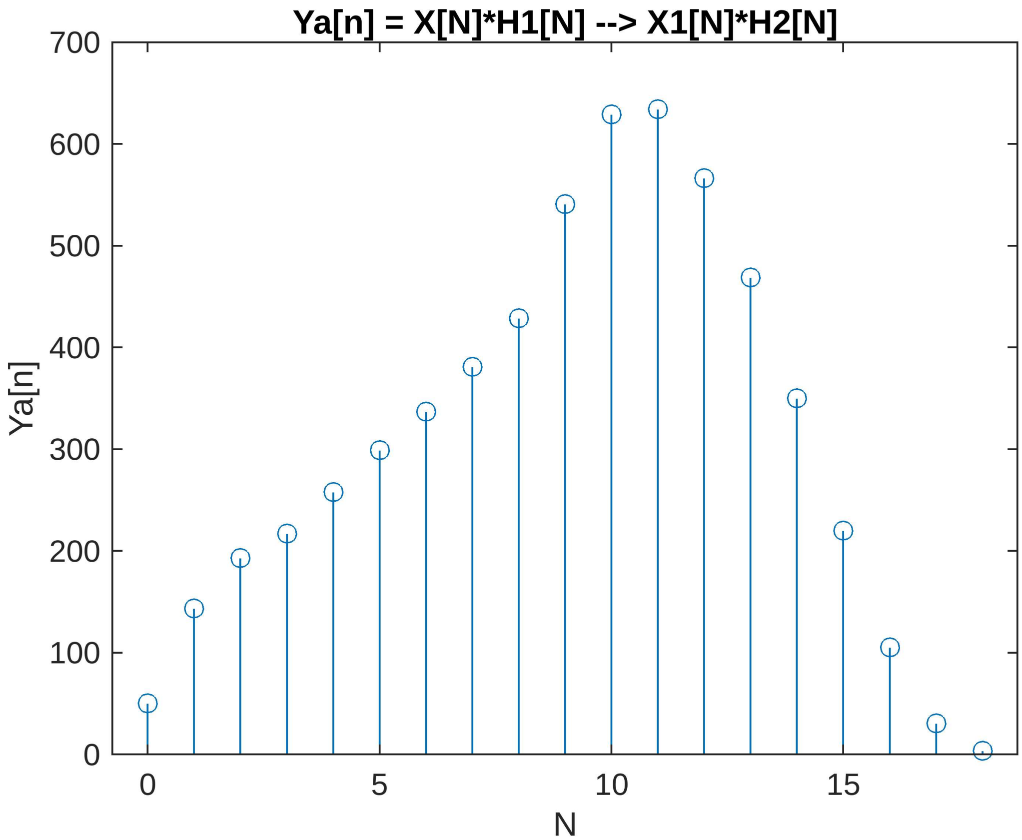

```
% As we can see all the plots of Ya[n] , Yb[n] and Yc[n] are same. It show
% the property of cascade systems.
```

Question 7

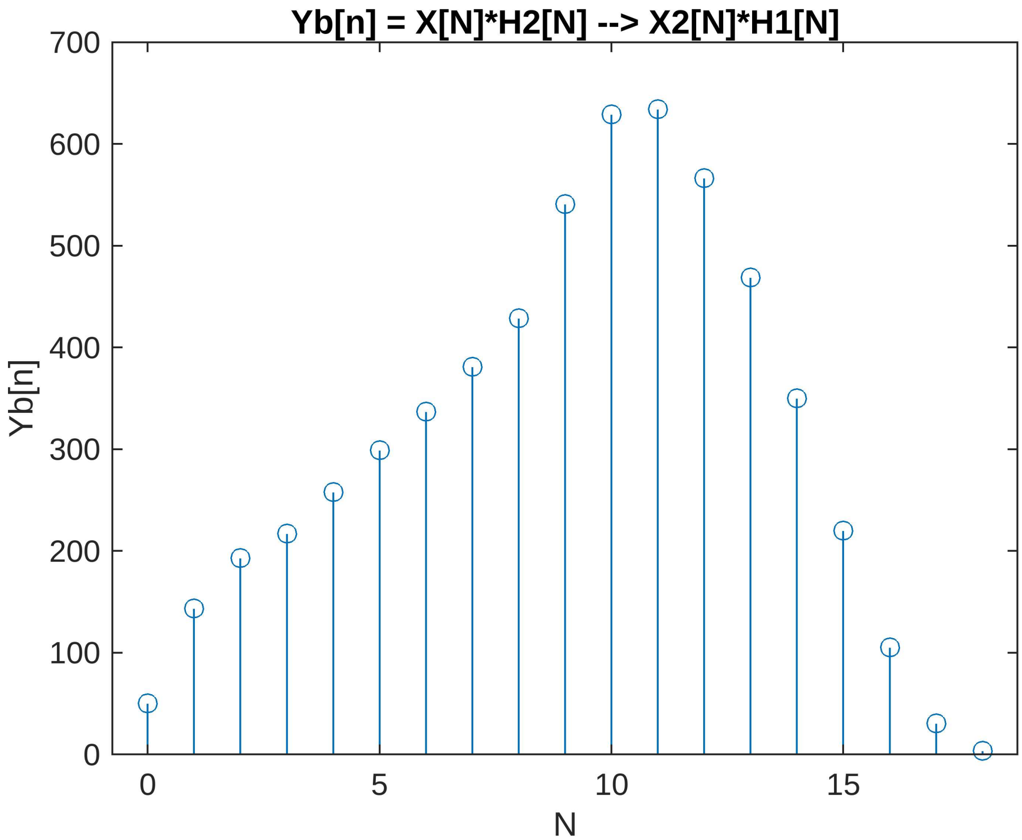
```
display("More random generated examples")
```

```
"More random generated examples"
```

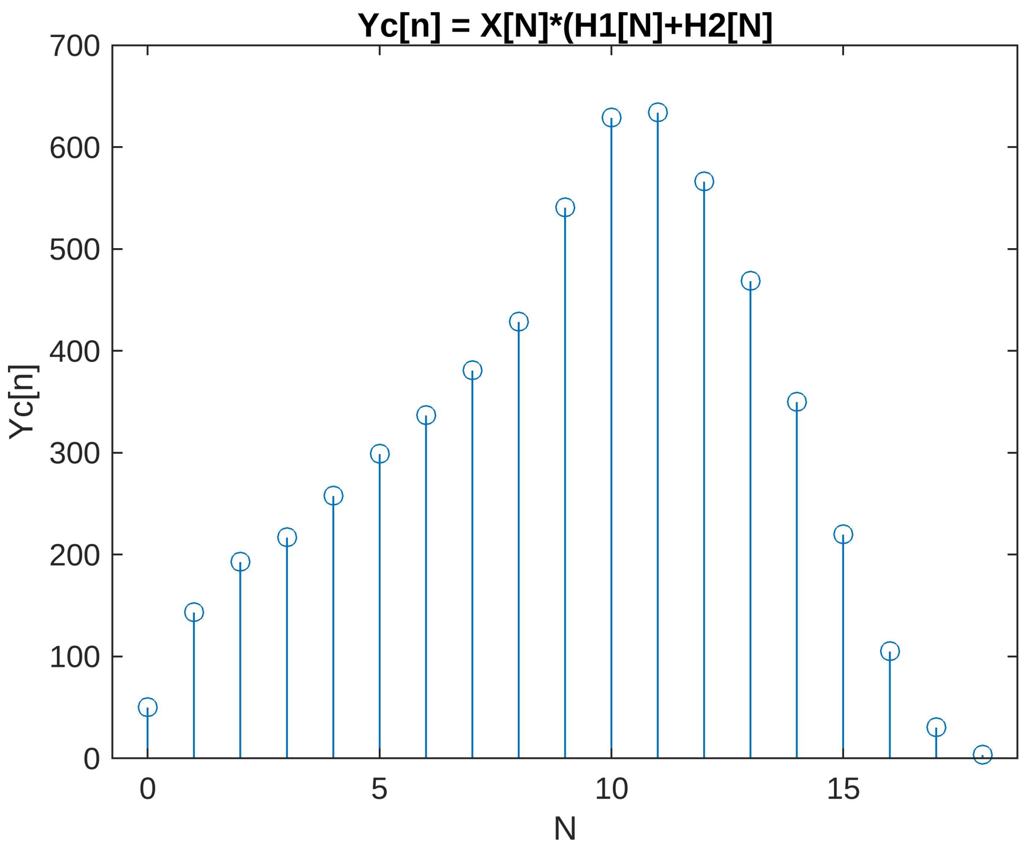
```
h1_n = 0+m1.*rand(1,m1+1);
h2_n = 0+m2.*rand(1,m2+1);
x_n = N.*rand(1,N+1);
x1_n = conv(x_n,h1_n);
Ya_n = conv(x1_n,h2_n);
x2_n = conv(x_n,h2_n);
Yb_n = conv(x2_n,h1_n);
h_n = conv(h1_n,h2_n);
Yc_n = conv(x_n,h_n);
n = 0:1:18;
stem(n,Ya_n);
xlabel("N");
ylabel("Ya[n]");
title("Ya[n] = X[N]*H1[N] --> X1[N]*H2[N]");
```




```
stem(n,Yb_n);
xlabel("N");
ylabel("Yb[n]");
title("Yb[n] = X[N]*H2[N] --> X2[N]*H1[N]");
```



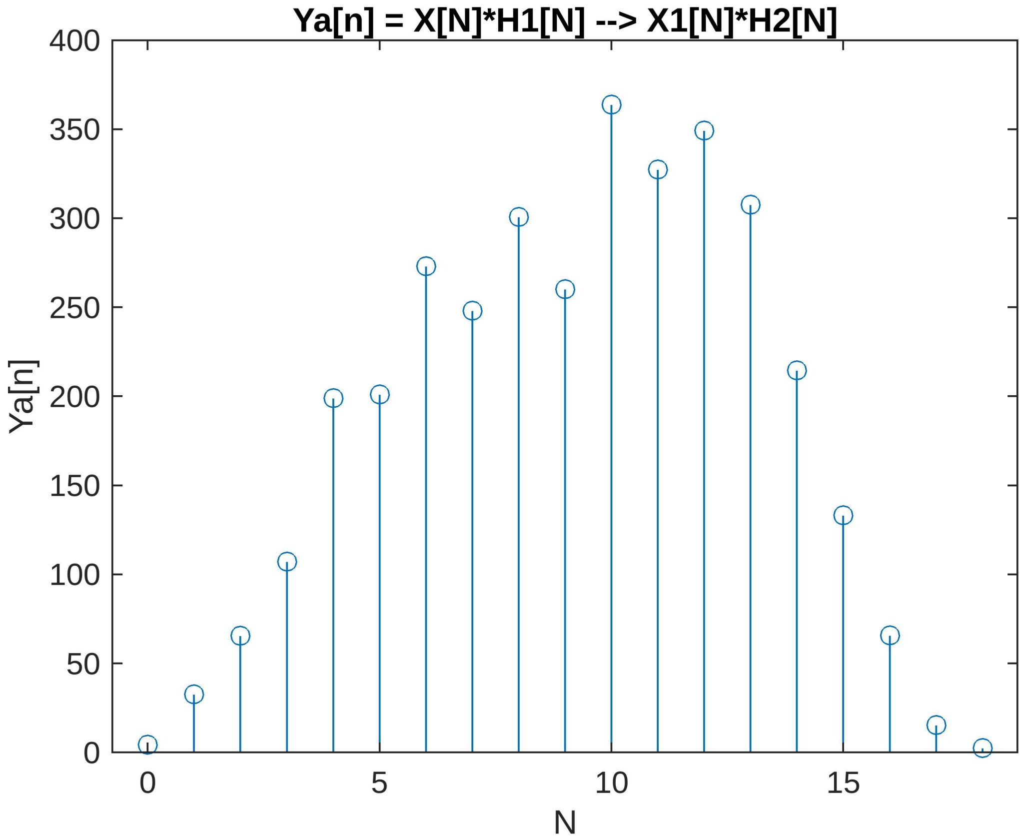

```
stem(n,Yc_n);
xlabel("N");
ylabel("Yc[n]");
title("Yc[n] = X[N]*(H1[N]+H2[N]");
```



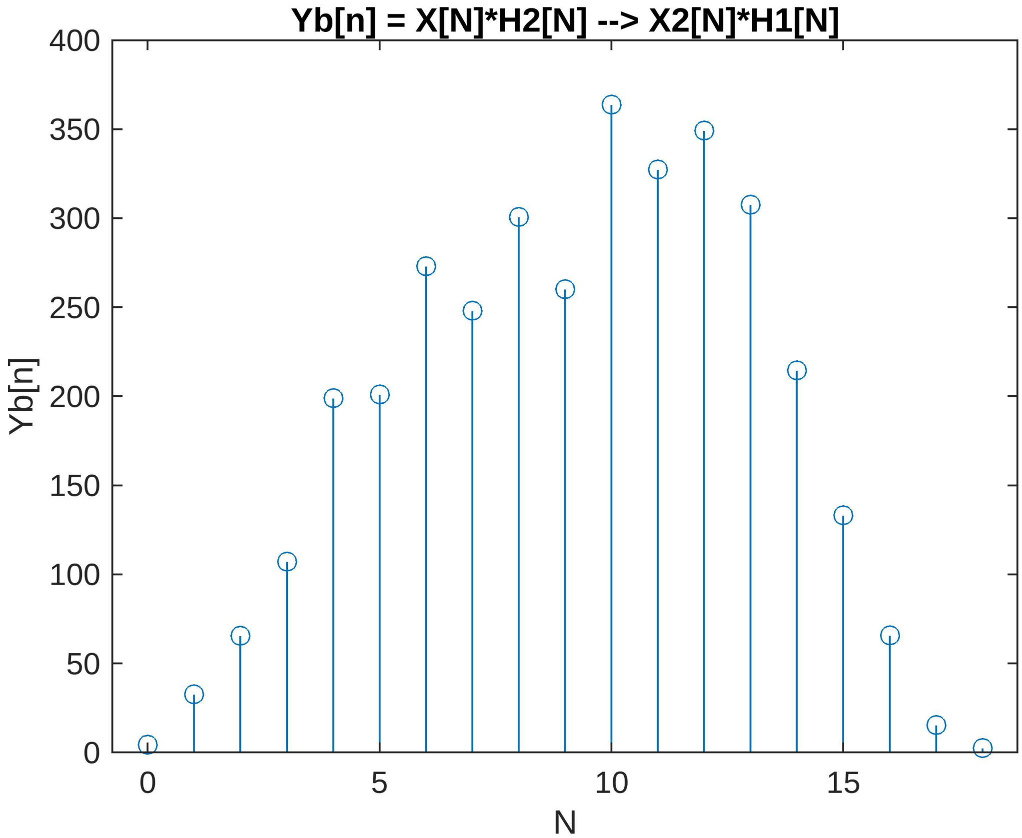

```
display("Example 2");
```

```
"Example 2"
```

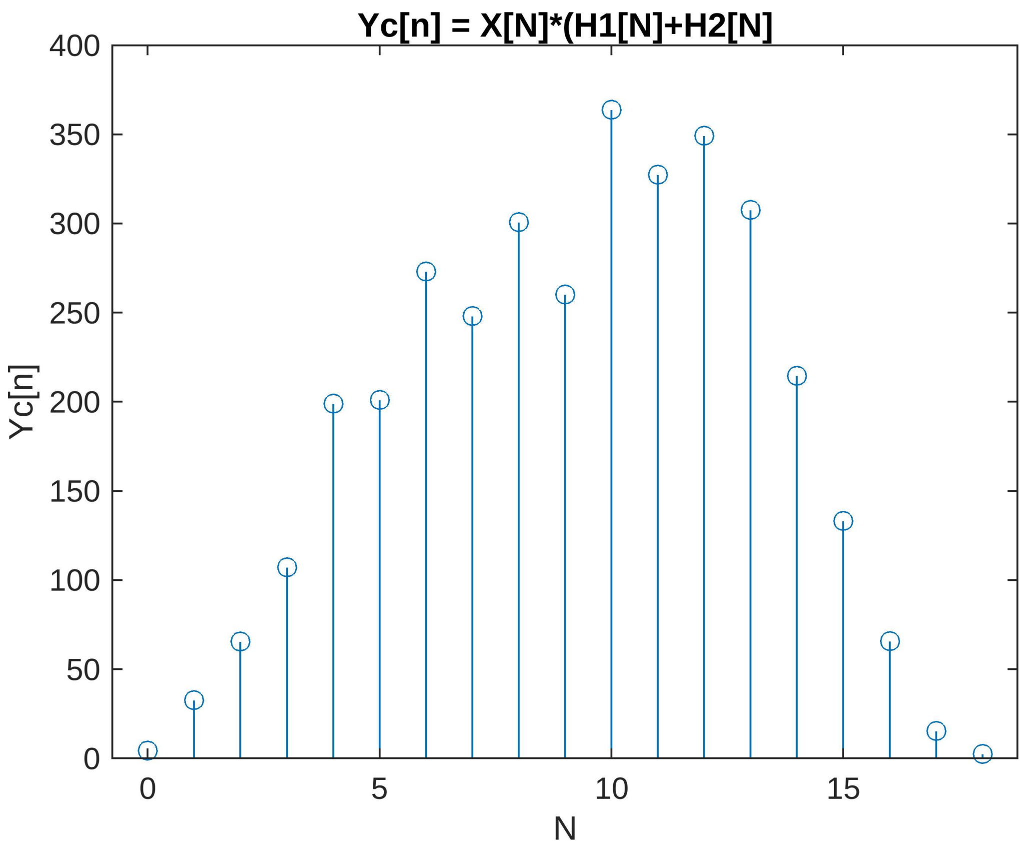
```
h1_n = 0+m1.*rand(1,m1+1);
h2_n = 0+m2.*rand(1,m2+1);
x_n = N.*rand(1,N+1);
x1_n = conv(x_n,h1_n);
Ya_n = conv(x1_n,h2_n);
x2_n = conv(x_n,h2_n);
Yb_n = conv(x2_n,h1_n);
h_n = conv(h1_n,h2_n);
Yc_n = conv(x_n,h_n);
n = 0:1:18;
stem(n,Ya_n);
xlabel("N");
ylabel("Ya[n]");
title("Ya[n] = X[N]*H1[N] --> X1[N]*H2[N]");
```




```
stem(n,Yb_n);
xlabel("N");
ylabel("Yb[n]");
title("Yb[n] = X[N]*H2[N] --> X2[N]*H1[N]");
```




```
stem(n,Yc_n);
xlabel("N");
ylabel("Yc[n]");
title("Yc[n]  = X[N]*(H1[N]+H2[N]");
```



D. CONJUGATE SYMMENTRY OF FOURIER TRANSFORM

1. Generate a random signal x[n] ,0<=n<=N.
2. Compute the fourier transform of x[n] : X[e^jwn] = sgm(x[n] * e^-jwn) n = N1 to N2 for w = -pi to pi
3. plot i) w vs Re{x(e^jw)} ii) w vs Im{x(e^jw)} report your observations.
4. plot i) w vs |`x(e^jw)`| ii) w vs ang(x(e^jw)) Report your observations
5. Repeat the experiements for a few values of random generated x[n].

Question 1

```
N = 20;
x_n = 20.*rand(1,N);
```

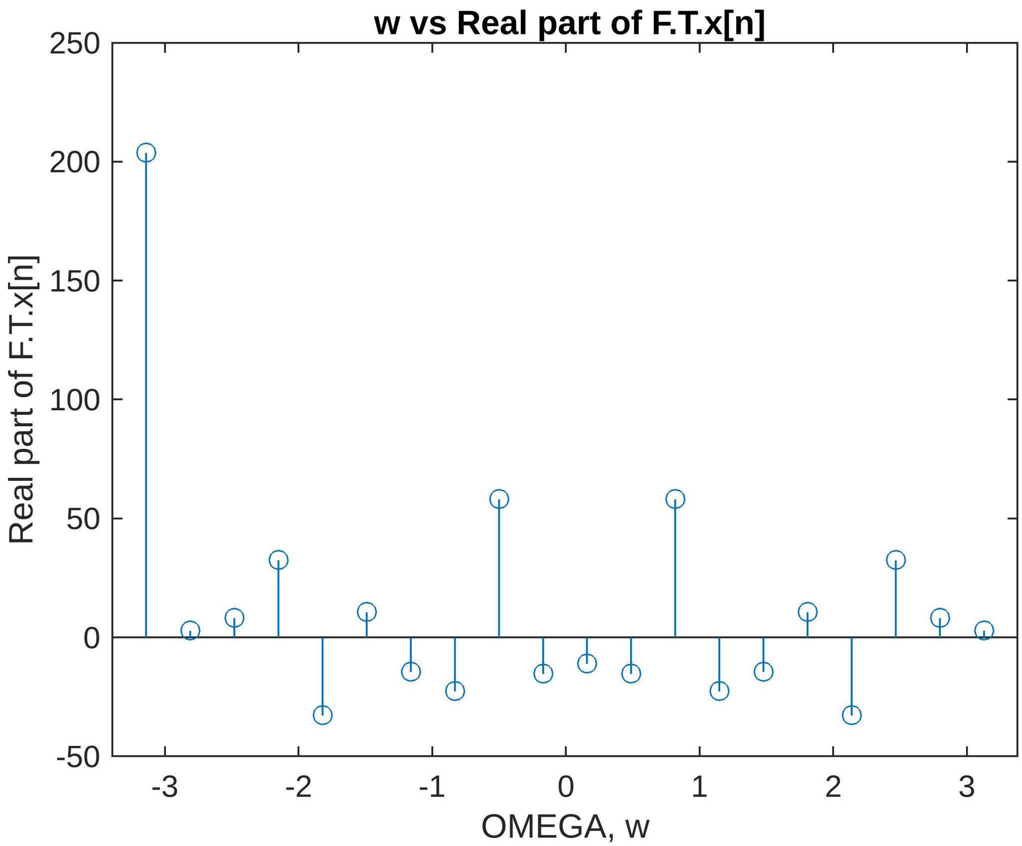
Question 2

```
xf_n = fft(x_n);
```

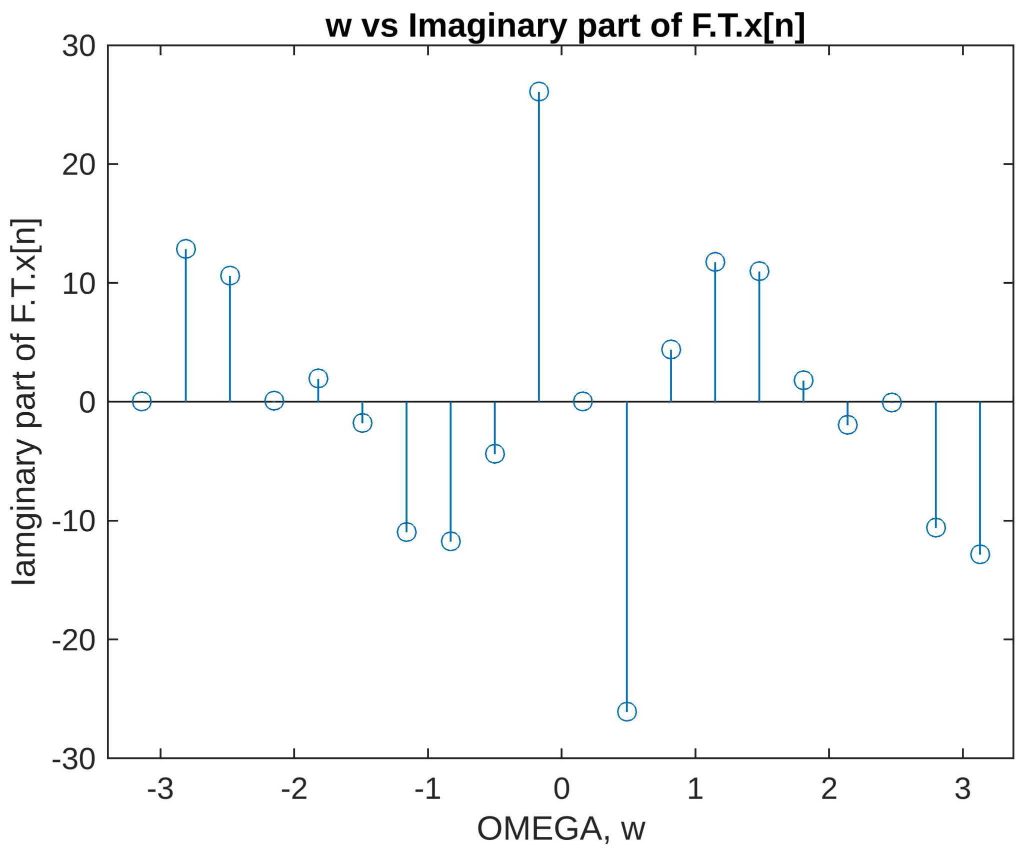
Question 3

```
w = -pi:0.33:pi;
stem(w,real(xf_n));
xlabel("OMEGA, w");
```

```
ylabel("Real part of F.T.{x[n]}");
title(" w vs Real part of F.T.{x[n]}")
```

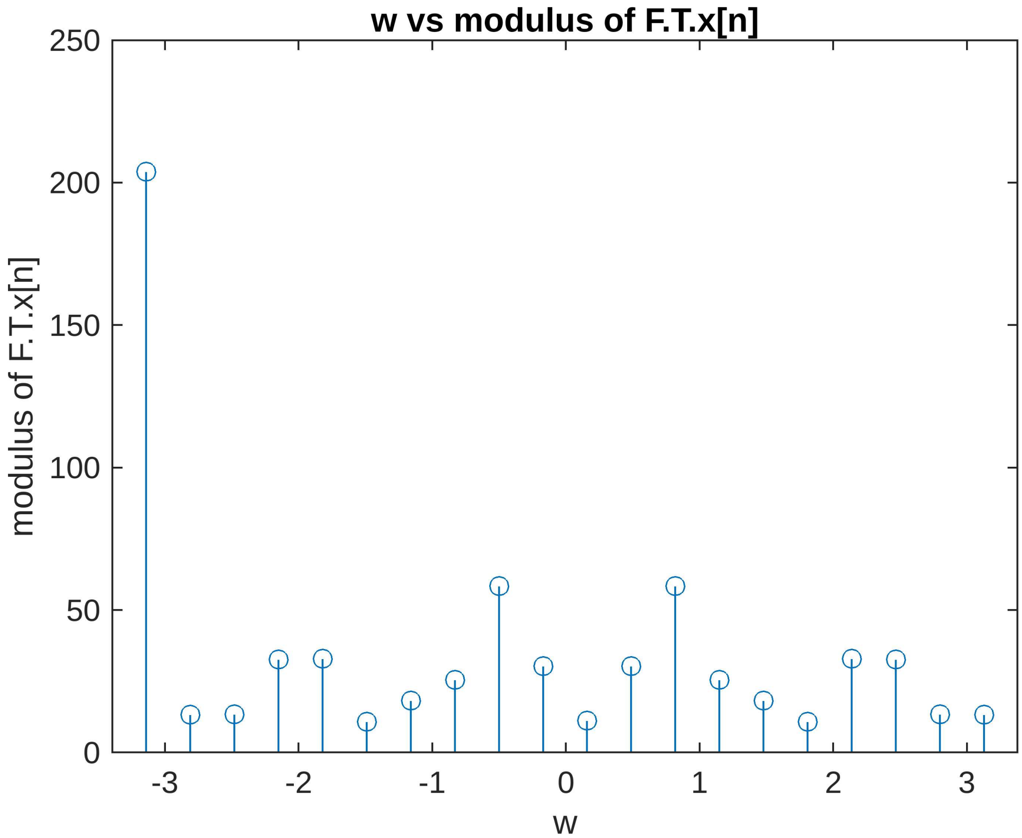



```
stem(w,imag(xf_n));
xlabel("OMEGA, w");
ylabel("Iamginary part of F.T.{x[n]}");
title(" w vs Imaginary part of F.T.{x[n]}");
```

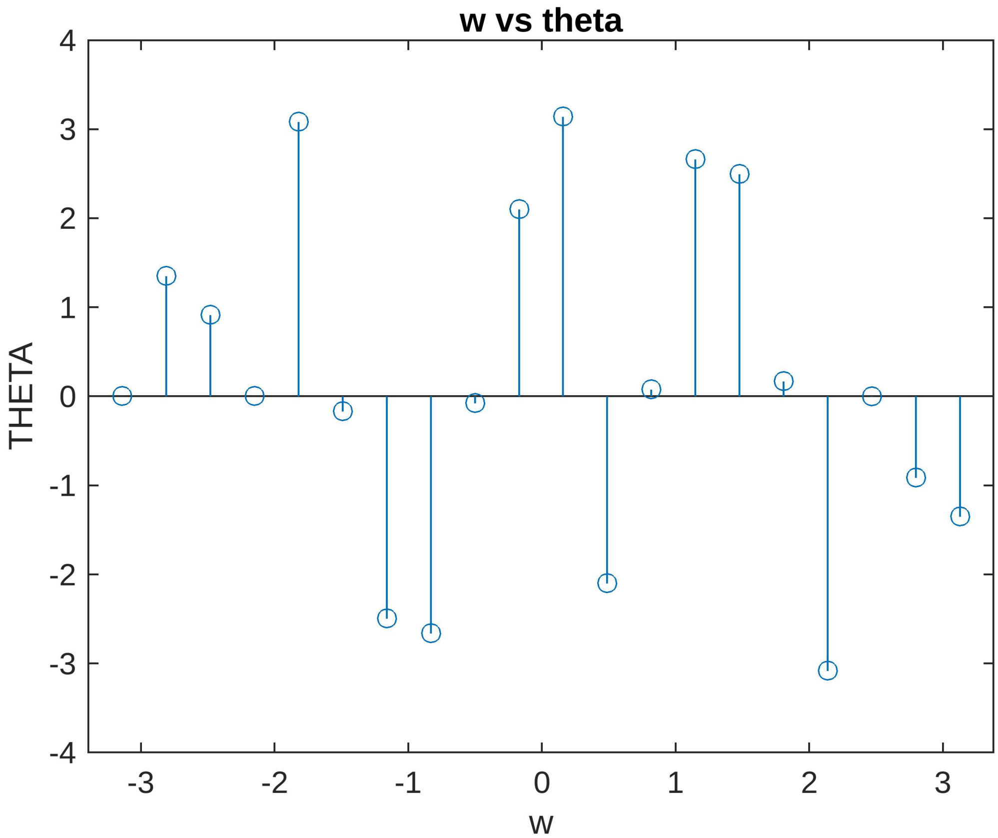


Question 4

```
mod_xf = abs(xf_n);
stem(w,mod_xf);
xlabel("w");
ylabel("modulus of F.T.{x[n]}");
title("w vs modulus of F.T.{x[n]}");
```




```
theta = angle(xf_n);
stem(w,theta);
xlabel("w");
ylabel("THETA");
title("w vs theta");
```

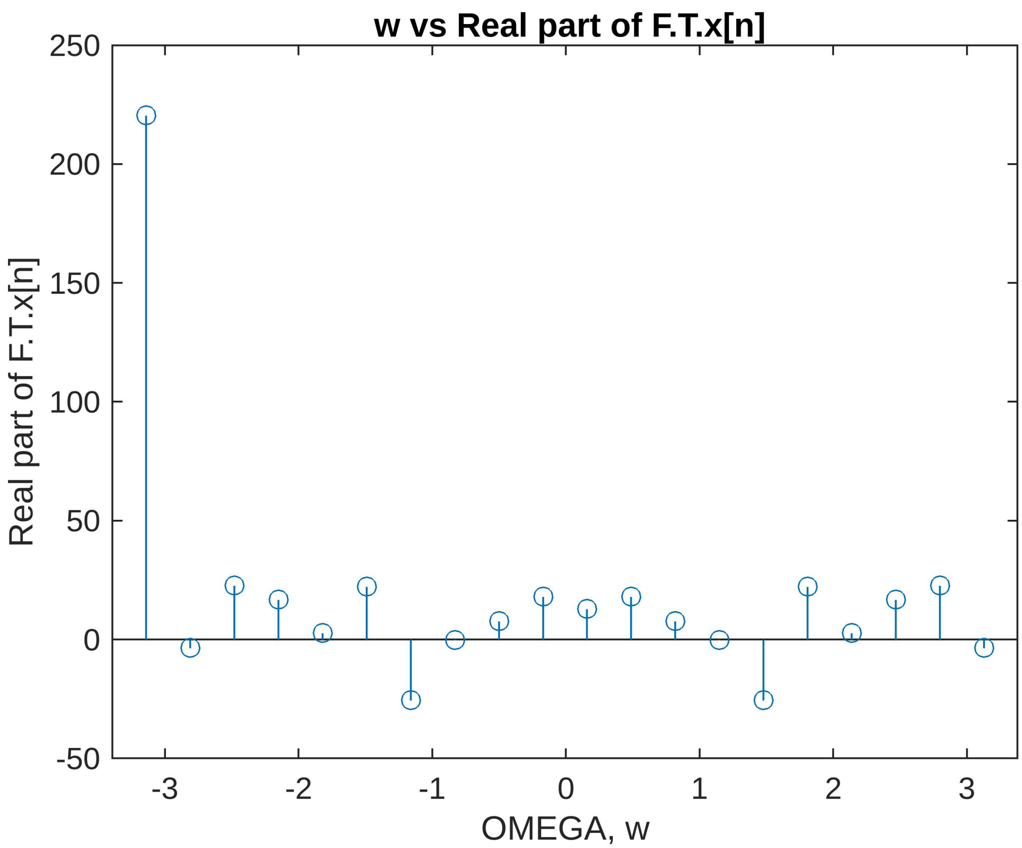


Question 5

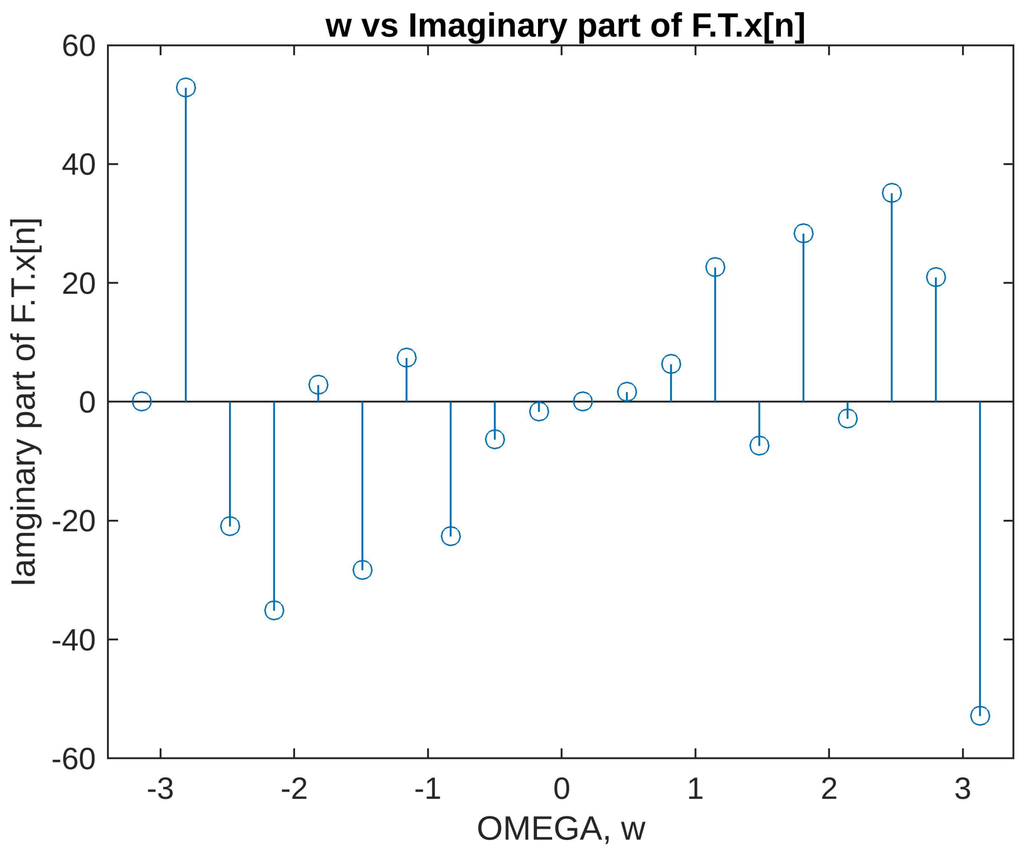
```
display("Different examples");
```

```
"Different examples"
```

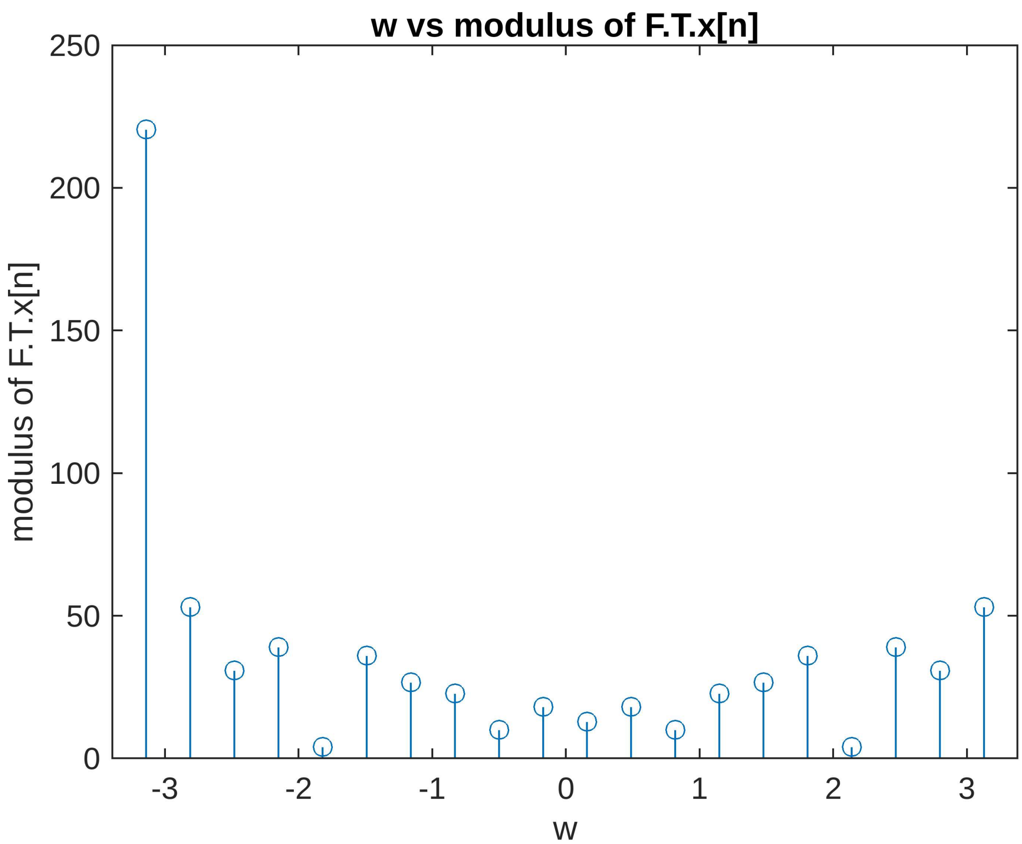
```
x_n = 20.*rand(1,N);
xf_n = fft(x_n);
w = -pi:0.33:pi;
stem(w,real(xf_n));
xlabel("OMEGA, w");
ylabel("Real part of F.T.{x[n]}");
title(" w vs Real part of F.T.{x[n]}")
```



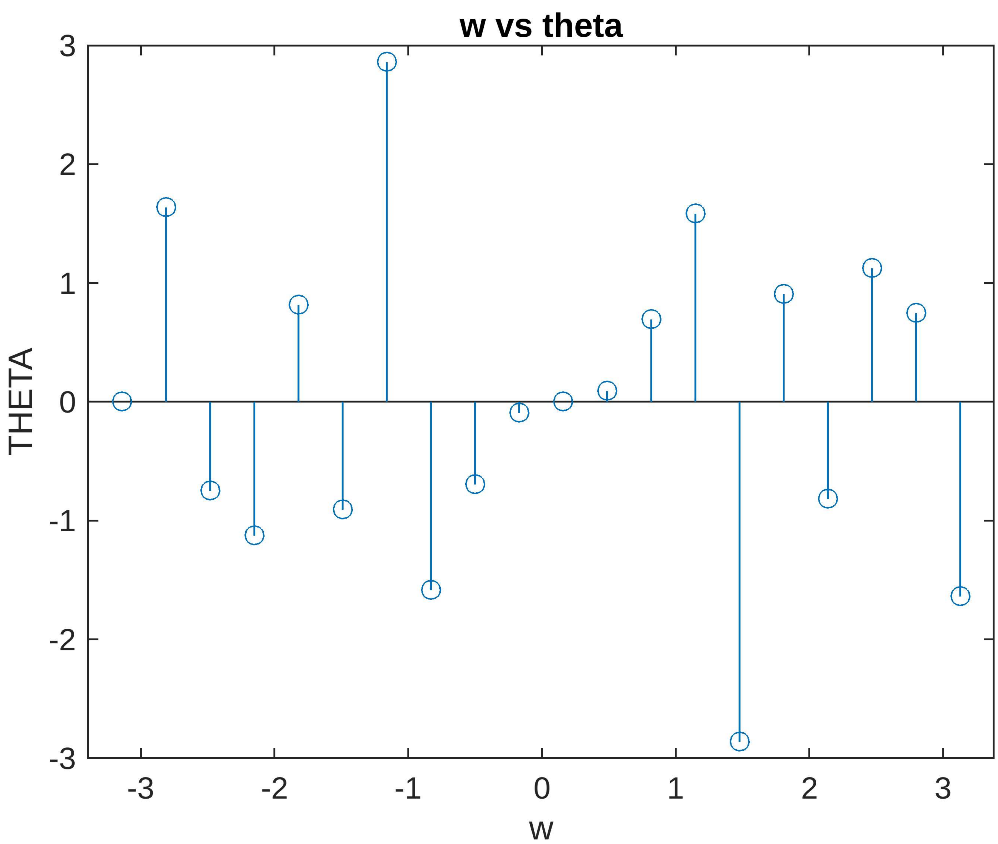

```
stem(w,imag(xf_n));
xlabel("OMEGA, w");
ylabel("Iamginary part of F.T.{x[n]}");
title(" w vs Imaginary part of F.T.{x[n]}");
```




```
mod_xf = abs(xf_n);
stem(w,mod_xf);
xlabel("w");
ylabel("modulus of F.T.{x[n]}");
title("w vs modulus of F.T.{x[n]}");
```



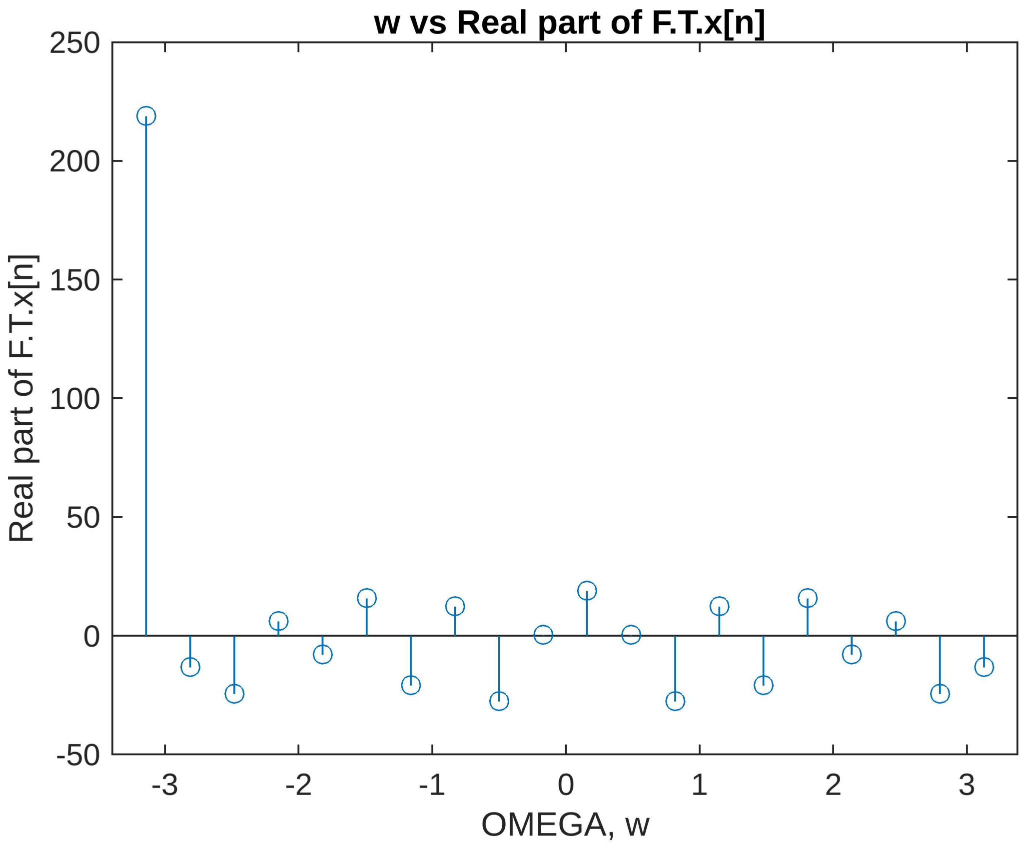

```
theta = angle(xf_n);
stem(w,theta);
xlabel("w");
ylabel("THETA");
title("w vs theta");
```



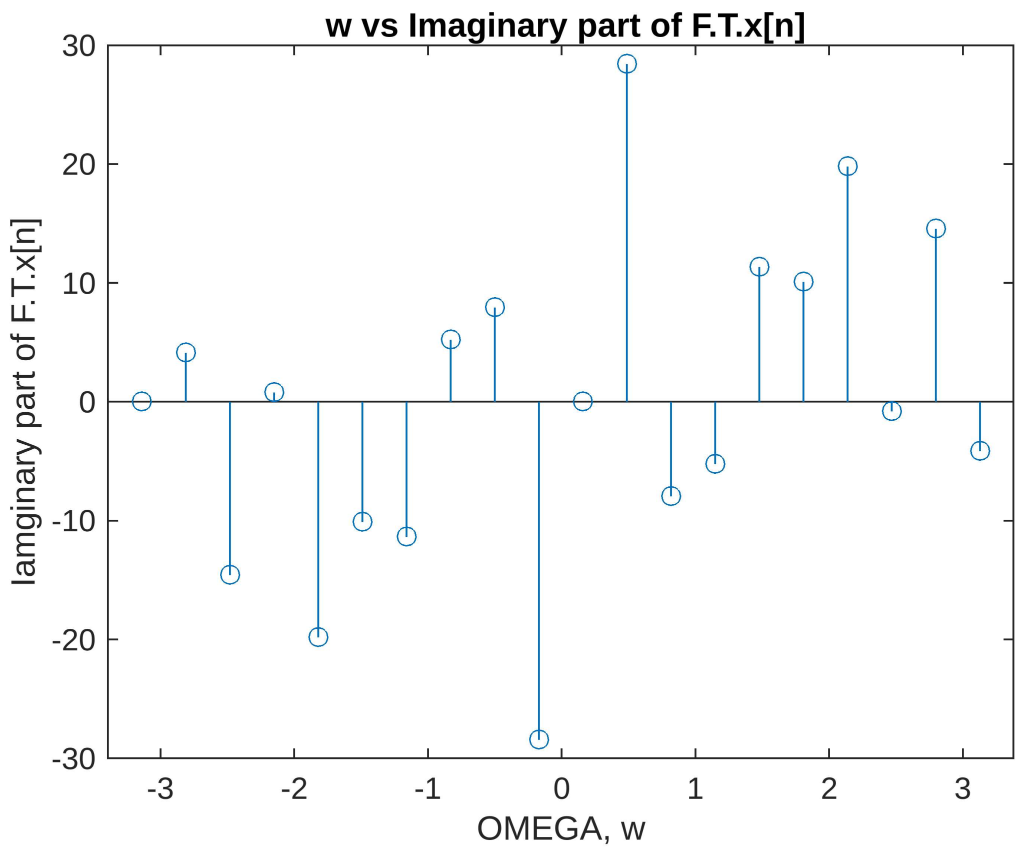

```
display("Example 2");
```

```
"Example 2"
```

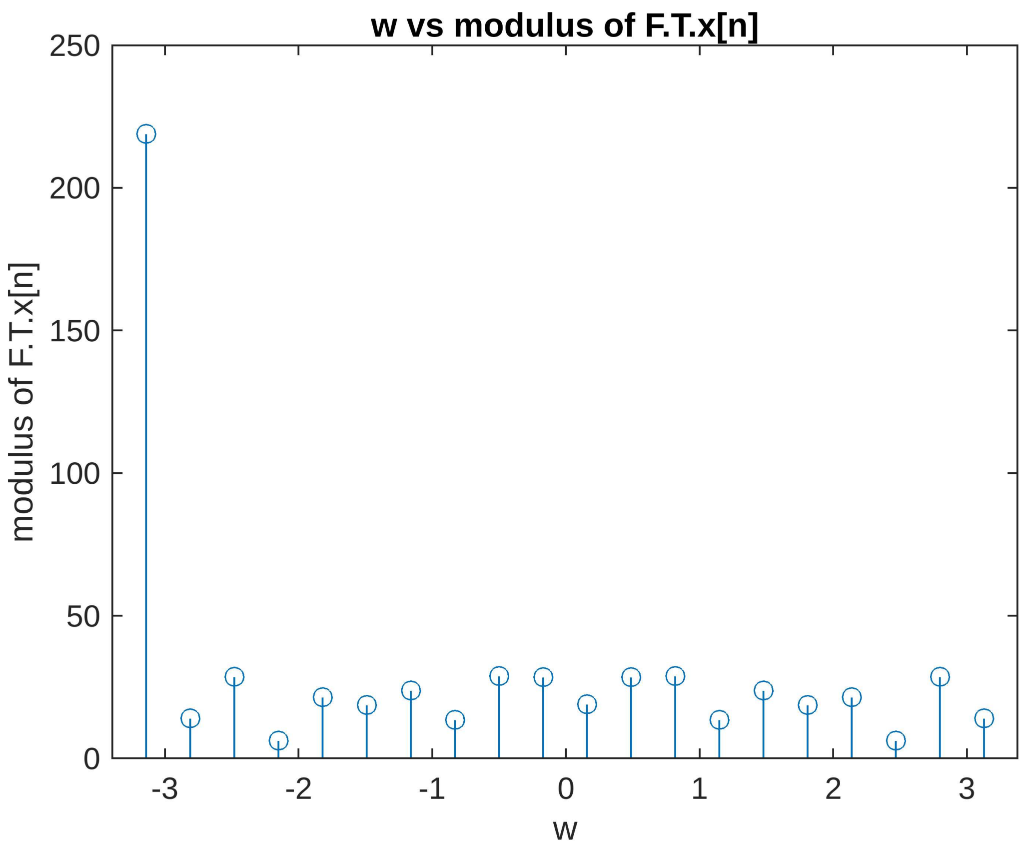
```
x_n = 20.*rand(1,N);
xf_n = fft(x_n);
w = -pi:0.33:pi;
stem(w,real(xf_n));
xlabel("OMEGA, w");
ylabel("Real part of F.T.{x[n]}");
title(" w vs Real part of F.T.{x[n]}")
```



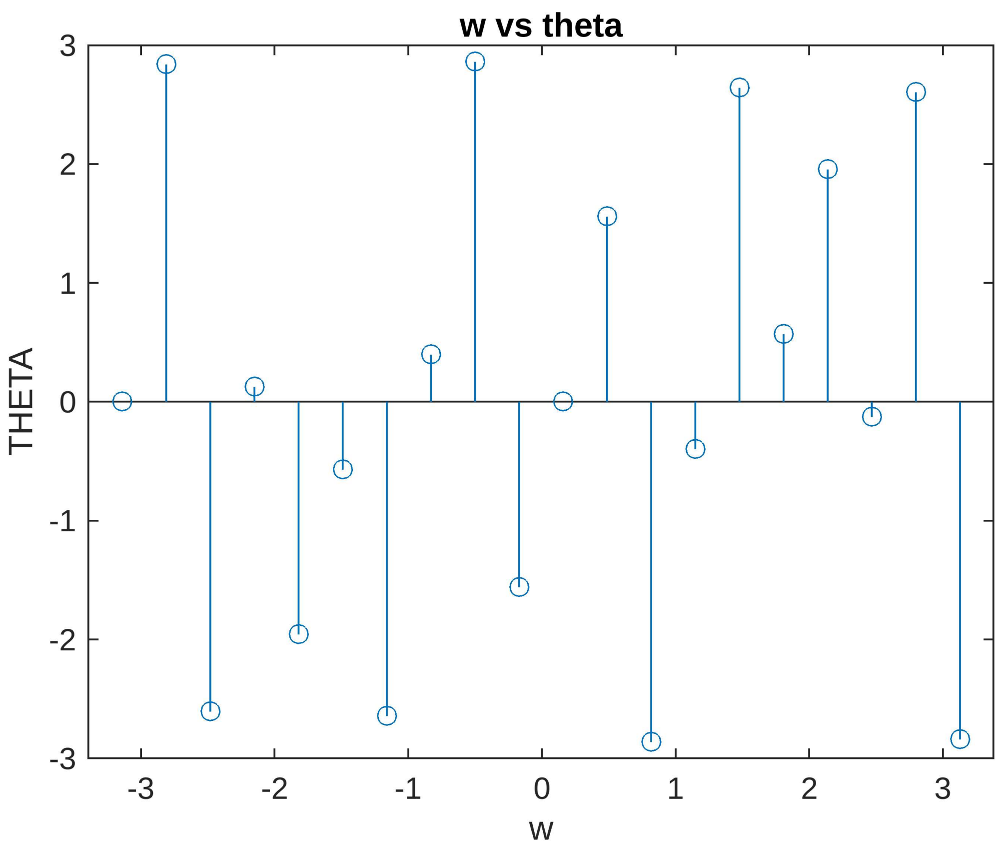

```
stem(w,imag(xf_n));
xlabel("OMEGA, w");
ylabel("Iamginary part of F.T.{x[n]}");
title(" w vs Imaginary part of F.T.{x[n]}");
```




```
mod_xf = abs(xf_n);
stem(w,mod_xf);
xlabel("w");
ylabel("modulus of F.T.{x[n]}");
title("w vs modulus of F.T.{x[n]}");
```




```
theta = angle(xf_n);
stem(w,theta);
xlabel("w");
ylabel("THETA");
title("w vs theta");
```



E. FOURIER TRANSFORM OF EVEN OR ODD SIGNALS.

1. Generate a random signal x[n] ,0<=n<=N.
2. Compute the fourier transform of x[n] : X[e^jwn] = sgm(x[n] * e^-jwn) n = N1 to N2 for w = -pi to pi.
3. Obtain even part of X[n] i.e. Xe[n].
4. Find F.T. of Xe[n].
5. Plot and compare F.T.(Xe[n]) and Re(X(e^jw).
6. Similarly obtain odd part i.e. Xo[n].
7. Find F.T.(Xo[n]).
8. Plot and compare F.T.(Xo[n]) and Im(X(e^jw).

Question 1

```
N = 7;
x_n = 7.*rand(1,N);
```

Question 2

```
xf_n = fft(x_n);
```

Question 3

```
x_nn = fliplr(x_n);
Xe_n = (x_n + x_nn)./2;
```

Question 4

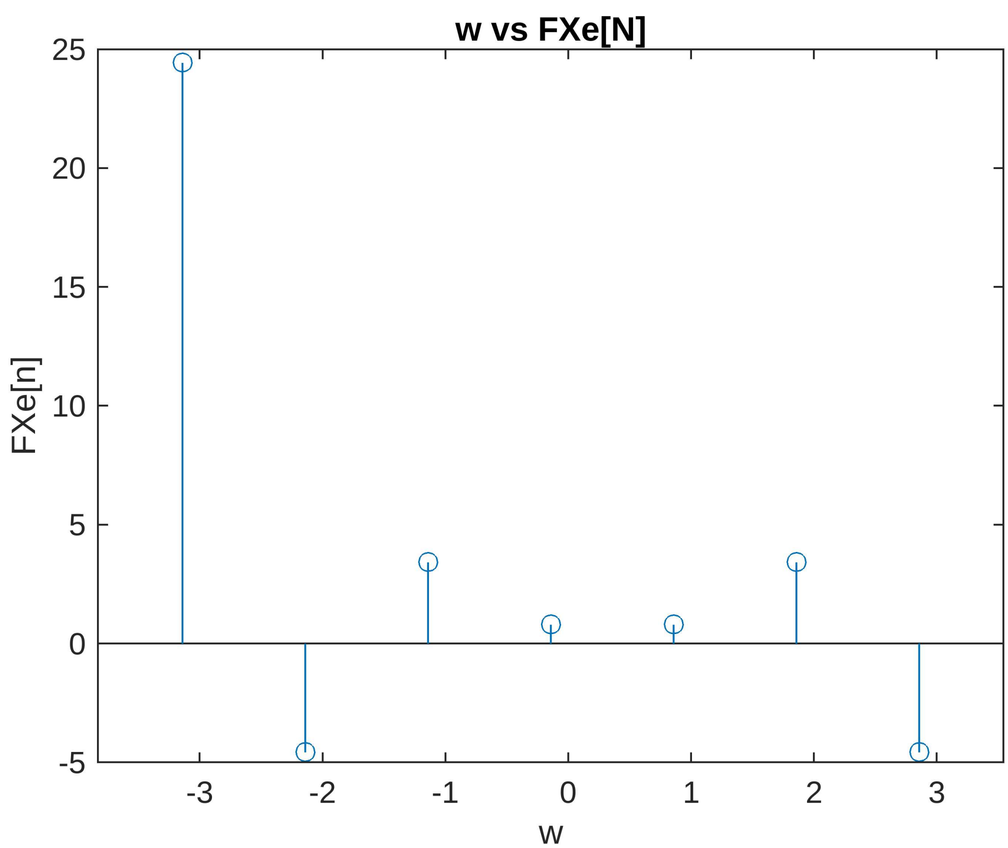
```
Xe_ft = fft(Xe_n);
```

Question 5

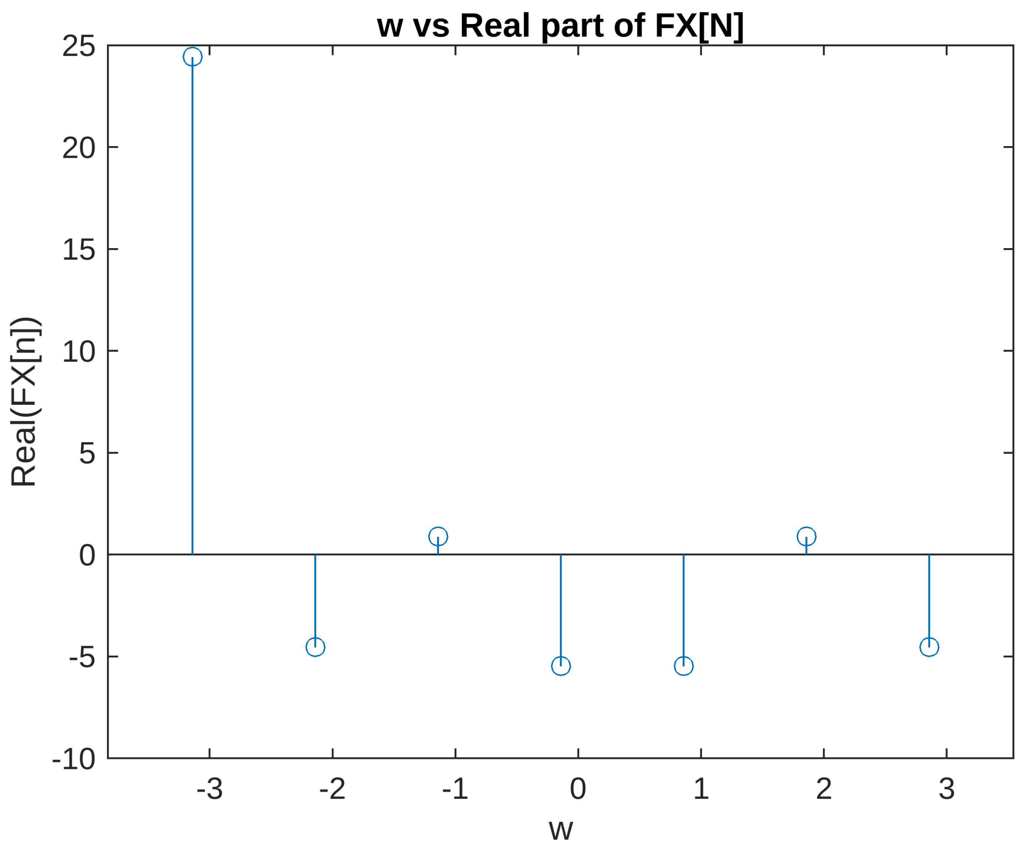
```
w = -pi:1:pi;
stem(w,Xe_ft);
```

```
Warning: Using only the real component of complex data.
```

```
xlabel("w");
ylabel("F{Xe[n]}");
title("w vs F{Xe[N]}");
```




```
stem(w,real(xf_n));
 xlabel("w");
ylabel("Real(F{X[n]})");
title("w vs Real part of F{X[N]}");
```



Question 6

```
Xo_n = (x_n - x_nn)./2;
```

Question 7

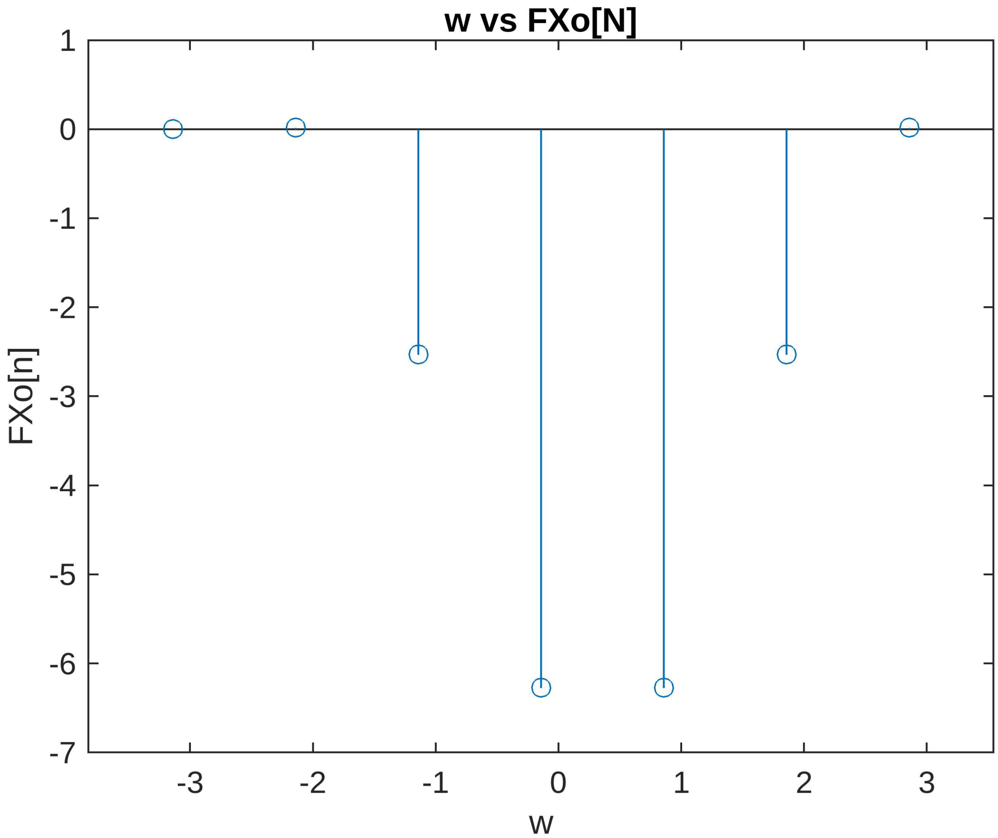
```
Xo_ft = fft(Xo_n);
```

Question 8

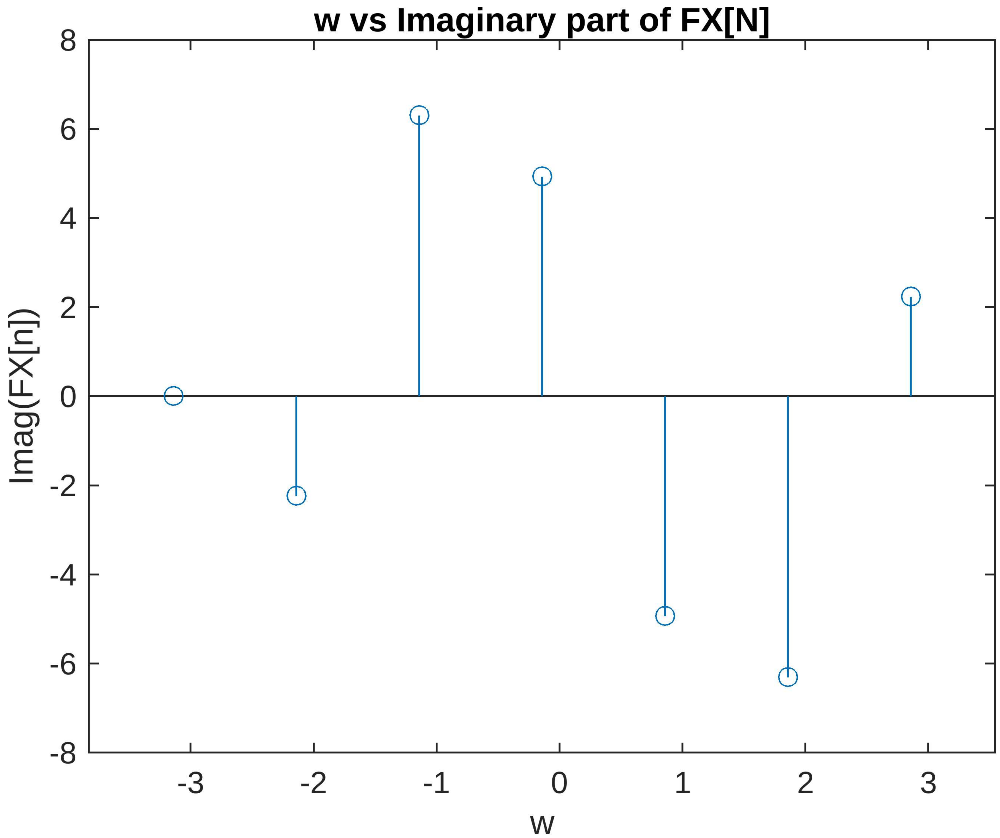
```
stem(w,Xo_ft);
```

```
Warning: Using only the real component of complex data.
```

```
xlabel("w");
 ylabel("F{Xo[n]}");
 title("w vs F{Xo[N]}");
```




```
stem(w,imag(xf_n));
xlabel("w");
 ylabel("Imag(F{X[n])}");
 title("w vs Imaginary part of F{X[N]}");
```


w vs Imaginary part of FX[N]
Imag(FX[n])
w
8
6
4
2
0
-2
-4
-6
-8
-3
-2
-1
0
1
2
3