DIGITAL SIGNAL PROCESSING

PRACTICAL FILE

SEMESTER V

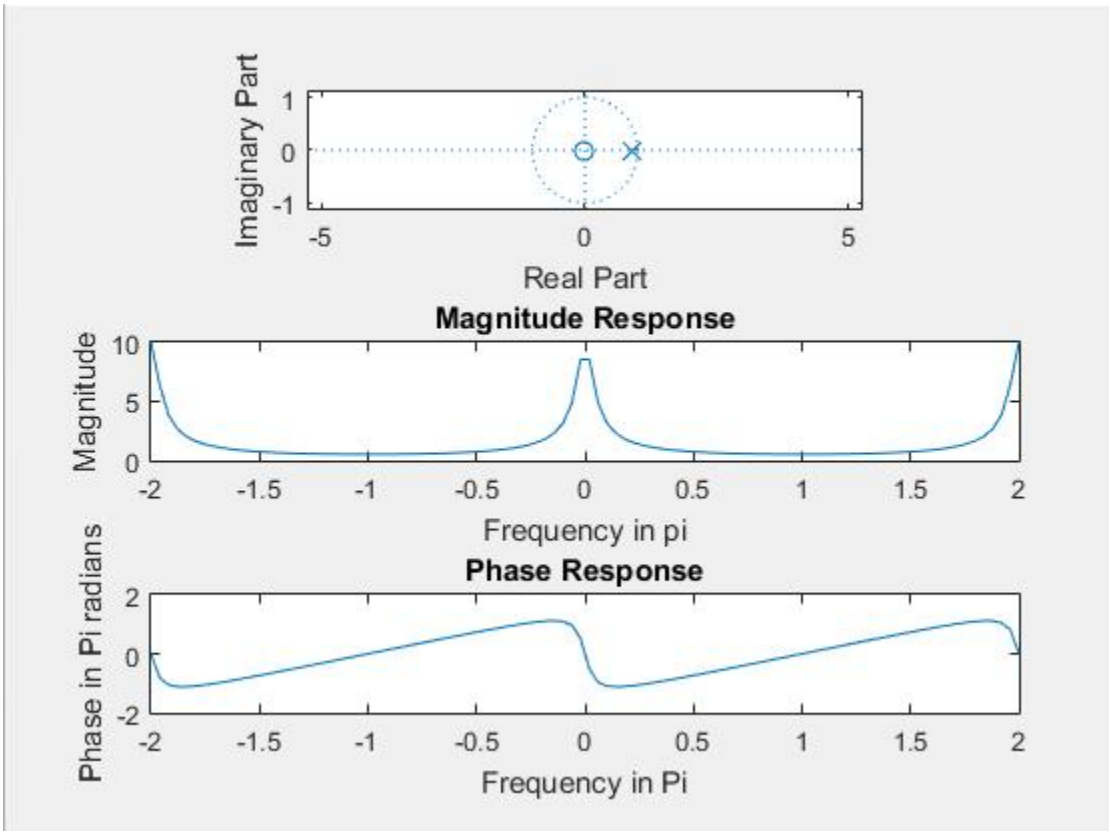
BY:
YASH BALYAN
2019UEC2619

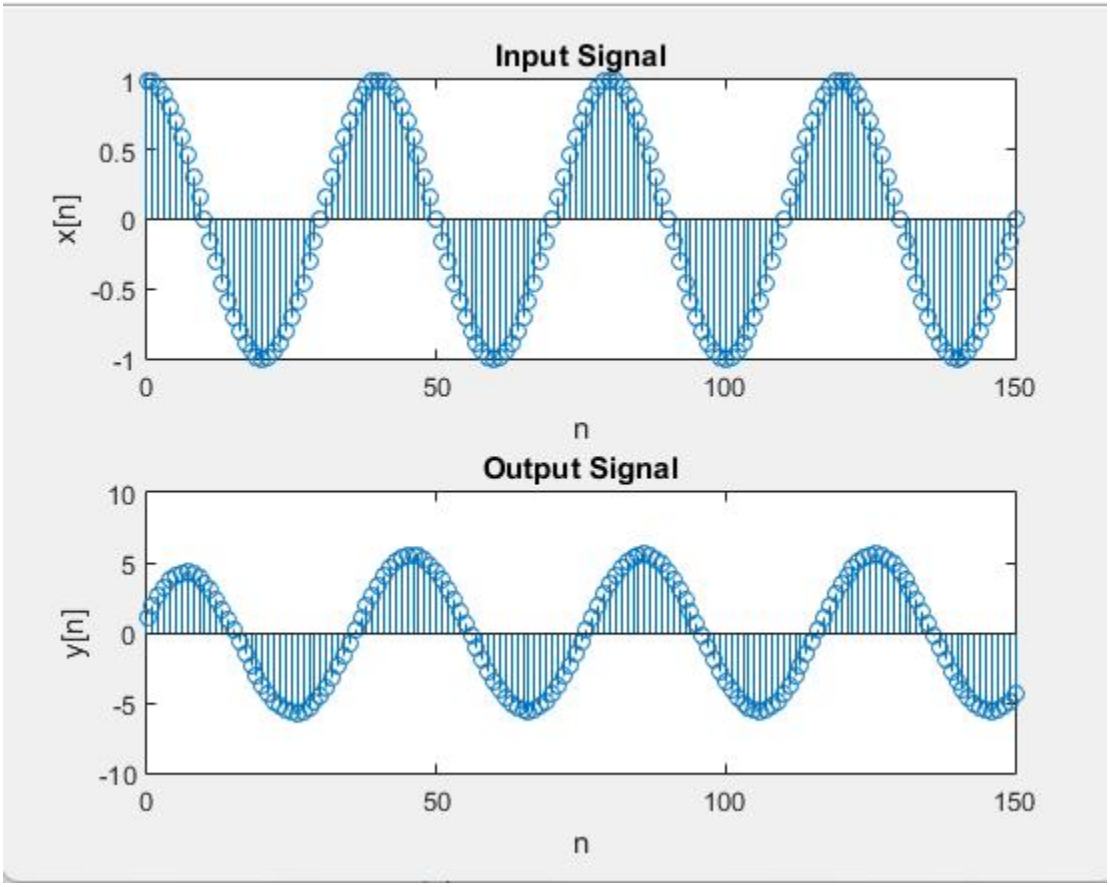
EXPERIMENT 1

MATLAB Code:

```
clc;
close all;

 % 1
 x=[1]; %Numerator of Transfer Function
 y=[1,-0.9]; %Denominator of Transfer Function
 subplot(3,1,1)
 zplane(x,y); %Pole-Zero Plot

% 2

x=[1]; %Numerator of Transfer Function
y=[1,-0.9]; %Denominator of Transfer Function
w=linspace(-2*pi,2*pi);
H=freqz(x,y,w);
magH=abs(H);
angH=angle(H);
subplot(3,1,2);
plot(w/pi,magH);
xlabel('Frequency in pi');
ylabel('Magnitude');
title('Magnitude Response');
subplot(3,1,3);
plot(w/pi,angH);
xlabel('Frequency in Pi');
ylabel('Phase in Pi radians');
title('Phase Response');


% 3
x=[1]; %Numerator of Transfer Function
y=[1,-0.9]; %Denominator of Transfer Function
n=[0:150];
z=cos(0.05*pi*n);
a=filter(x,y,z);
figure,subplot(2,1,1);
stem(n,z);
xlabel('n');
ylabel('x[n]');
title('Input Signal');
subplot(2,1,2);
stem(n,a);
xlabel('n');
ylabel('y[n]');
title('Output Signal');xlabel('n');
```

Output:


Imaginary Part
1
0
-1
-5
0
5
Real Part
Magnitude Response
Magnitude
10
5
0
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
Frequency in pi
Phase Response
Phase in Pi radians
2
0
-2
Frequency in Pi


Input Signal
x[n]
1
0.5
0
-0.5
-1
0
50
100
150
n
Output Signal
y[n]
10
5
0
-5
-10
n

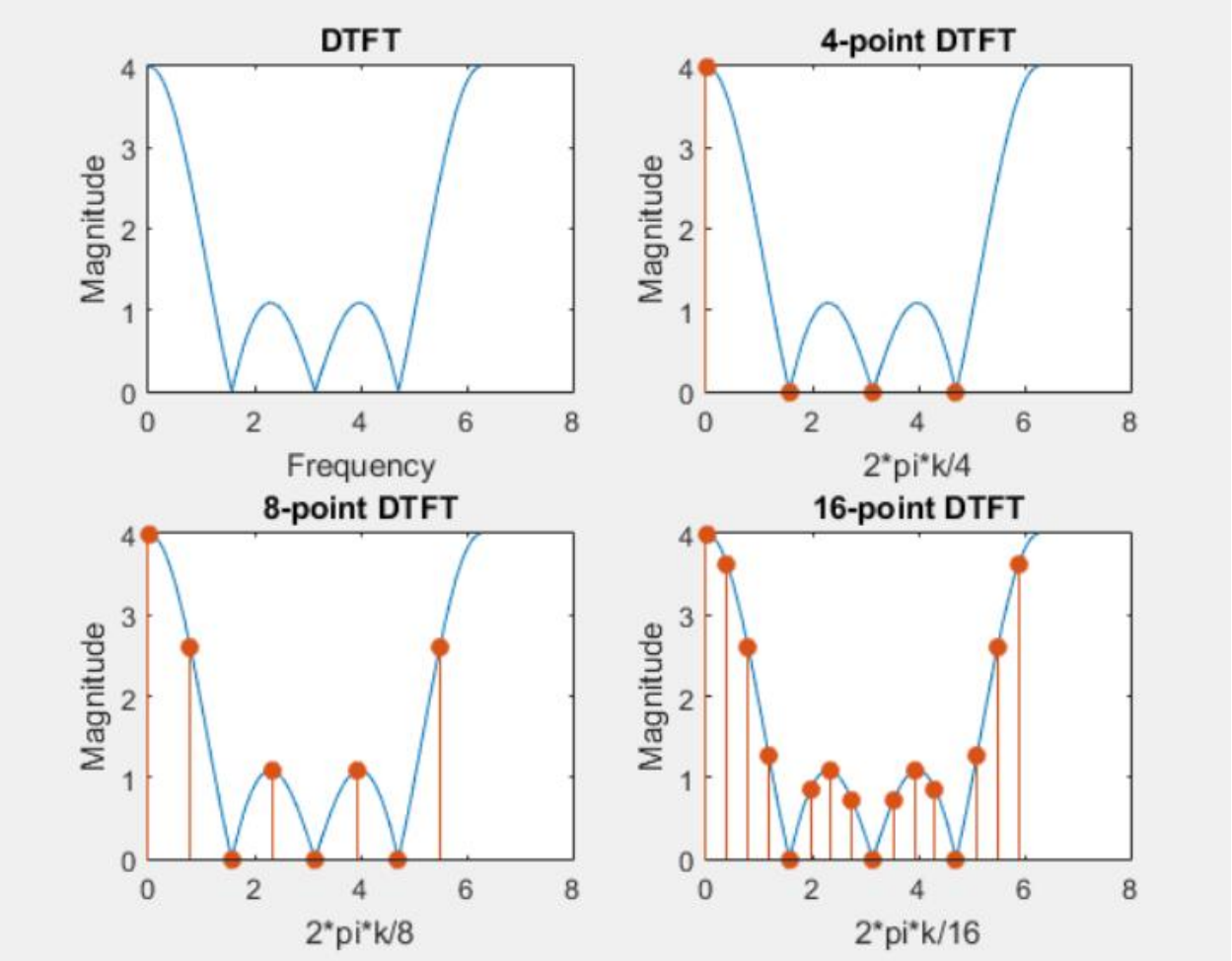
EXPERIMENT 2

MATLAB Code:

```
clc;
close all;

%DTFT
w=0:0.01:2*pi;
Xw=1+exp(-j*w)+exp(-j*2*w)+exp(-j*3*w);
%4-point DTFT
xn=ones(1,4);
N=4;
n=0:N-1;
k=0:N-1;
WN=exp(-j*2*pi/N*n'*k);
Xk=xn*WN;
subplot(221);
plot(w,abs(Xw));
xlabel('Frequency');
ylabel('Magnitude');
title('DTFT');
subplot(222);
plot(w,abs(Xw));hold on
stem(2*pi*k/N,abs(Xk),'filled');
xlabel('2*pi*k/4');
ylabel('Magnitude');
title('4-point DTFT');
%8-point DTFT
clear N;
xn=[ones(1,4) zeros(1,4)];
```

```
N=8;
n=0:N-1;
k=0:N-1;
WN=exp(-j*2*pi/N*n'*k);
Xk=xn*WN;
subplot(223);
plot(w,abs(Xw));hold on
stem(2*pi*k/N,abs(Xk),'filled');
xlabel('2*pi*k/8');
ylabel('Magnitude');
title('8-point DTFT');
%16-point DTFT
clear N;
xn=[ones(1,4) zeros(1,12)];
N=16;
n=0:N-1;
k=0:N-1;
WN=exp(-j*2*pi/N*n'*k);
Xk=xn*WN;
subplot(224);
plot(w,abs(Xw));hold on
stem(2*pi*k/N,abs(Xk),'filled');
xlabel('2*pi*k/16');
ylabel('Magnitude');
title('16-point DTFT');
```

Output:



EXPERIMENT 3

MATLAB Code:

(i)
```
clc;
close all;

N=4;
x1=[1 2 2 1];
x2=[1 -1 -1 1];
y=zeros(1,N);
%circular convolution
for n=0:N-1
 m=0:N-1;
 n1=mod(n-m,N);
 xs2=x2(n1+1);
 x12=x1.*xs2;
 y(n+1)=sum(x12);
end
n=0:N-1;
```

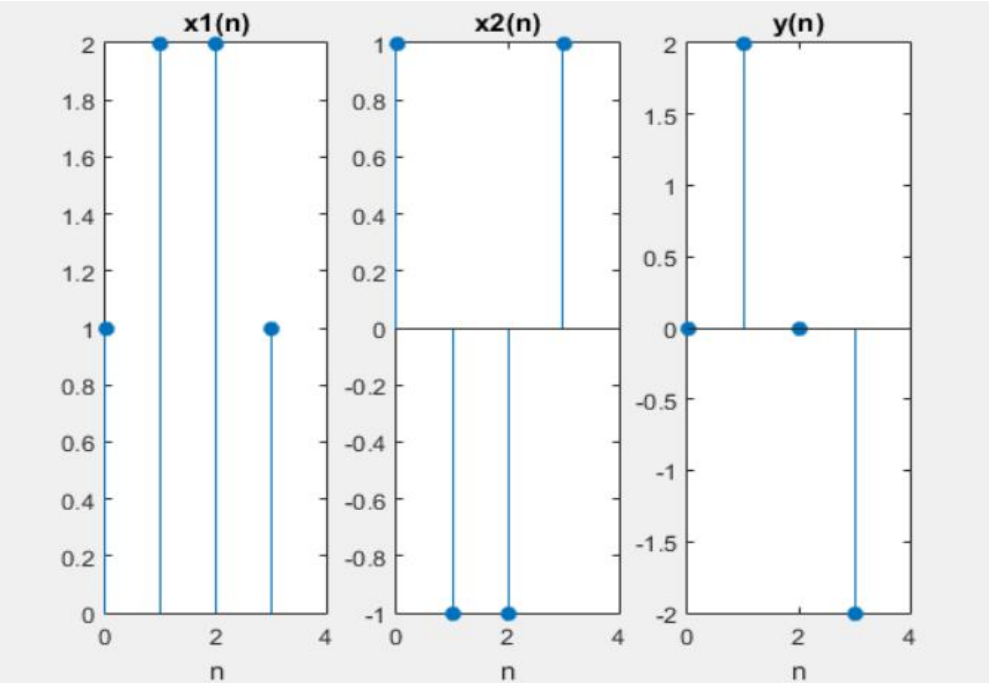
```
subplot(131);
stem(n,x1,'filled');
xlabel('n');
title('x1(n)');
subplot(132);
stem(n,x2,'filled');
xlabel('n');
title('x2(n)');
subplot(133);
stem(n,y,'filled');
xlabel('n');
title('y(n)');
```

(ii)

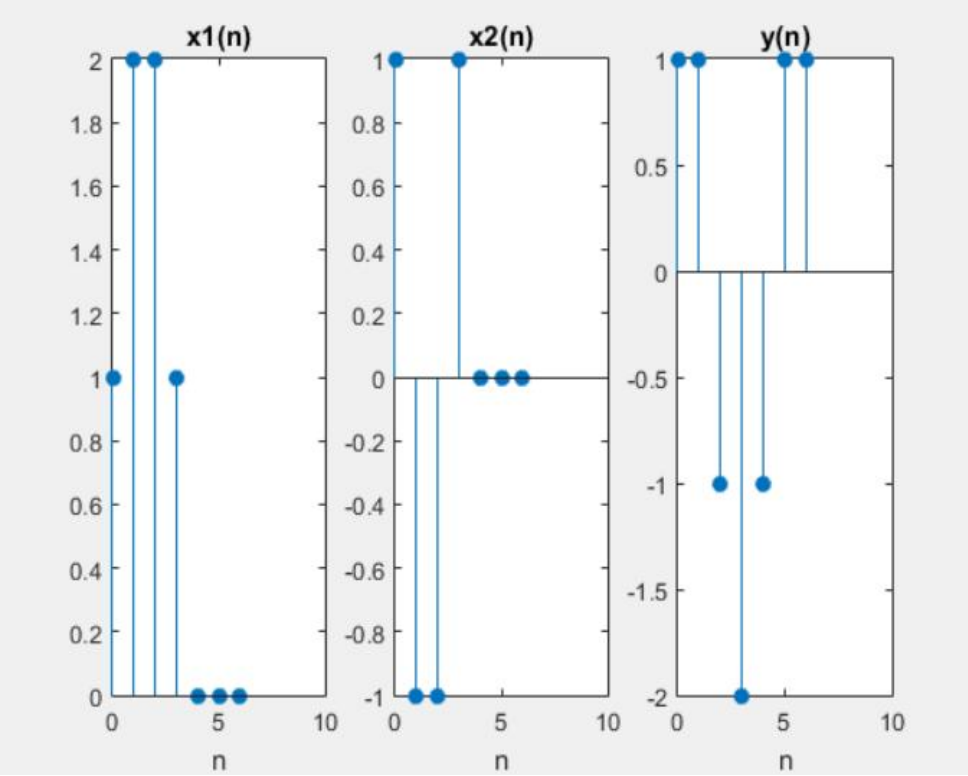
```
clc;
close all;

N=4;
x1=[1 2 2 1];
Nx1=length(x1);
x2=[1 -1 -1 1];
Nx2=length(x2);
Ny=Nx1+Nx2-1;
x1z=[x1 zeros(1,Ny-Nx1)];
x2z=[x2 zeros(1,Ny-Nx2)];
y=zeros(1,Ny);
%circular convolution
for n=0:Ny-1
 m=0:Ny-1;
 n1=mod(n-m,Ny);
 x2s=x2z(n1+1);
 x12=x1z.*x2s;
 y(n+1)=sum(x12);
end
n=0:Ny-1;
subplot(131);
stem(n,x1z,'filled');
xlabel('n');
title('x1(n)');
subplot(132);
stem(n,x2z,'filled');
xlabel('n');
title('x2(n)');
subplot(133);
stem(n,y,'filled');
xlabel('n');
title('y(n)');
```

Output:

(i)



(ii)



EXPERIMENT 4

MATLAB Code:

(i)

```
%Experiment 4(a)
clc;
close all;

n=0:9;
N=6;
x=n+1;
Lenx= length(x);
h=[1 0 -1];
M=length(h);
M1=M-1;
L=N-M1;
hz=[h zeros(1,N-M)]; %Appending N-M Zeros
nhz=0:length(hz)-1;
xz=[zeros(1,M1) x zeros(1,N-1)]; %Pre Appending M-1 Zeros
nxz=0:length(xz)-1;
K=ceil((Lenx+M1-1)/L);
y=zeros(K,N);
for i=0:K-1
 xi=xz(i*L+1:i*L+N)
 for j=0:N-1
 m=0:N-1;
 n1=mod(j-m,N);
 hs=hz(n1+1);
 xh=xi.*hs;
 y(i+1,j+1)=sum(xh);
 end
end
y=y(:,M:N)'; %DiscaRding First M-1 Samples
y=[y(:)]'; %Concatenating The Output
ny=0:length(y)-1;
subplot(1,3,1);
stem(nxz,xz,'filled');
xlabel('n');
ylabel('x(n)');
title('x(n)');
subplot(1,3,2);
stem(nhz,hz,'filled');
xlabel('n');
ylabel('h(n)');
title('h(n)');
subplot(1,3,3);
stem(ny,y,'filled');
xlabel('n');
```

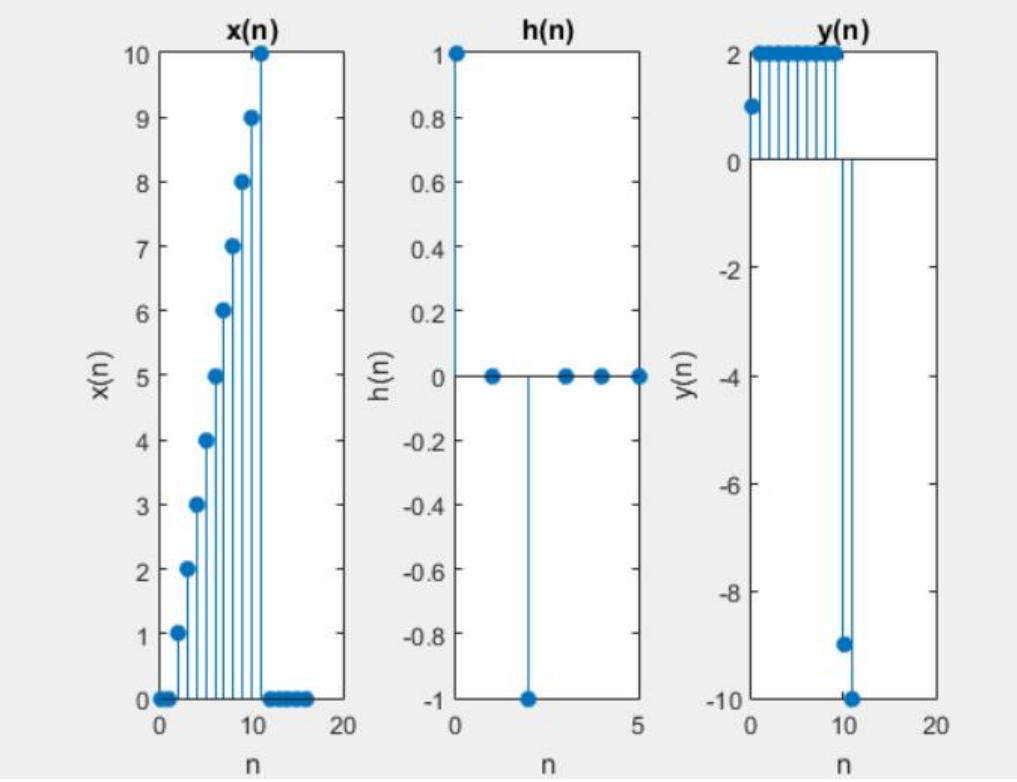
```
ylabel('y(n)');
title('y(n)');


(ii)
close all;
clc;

n=0:9;
N=4;
x=n+1;
Lenx=length(x);
h=[1 0 -1];
M=length(h);
M1=M-1;
L=N+M-1;
hz=[h zeros(1,L-M)]; %Appending N-M Zeros
nhz=0:length(hz)-1;
K=ceil(Lenx/N);
xx=[x zeros(1,N*K-Lenx)]; %Preappending M-1 Zeros
nxx=0:length(xx)-1;
y=zeros(K,N);
for i=0:K-1
 xi=xx(i*N+1:N*(i+1));
 xr=[xi zeros(1,M1)];
 for j=0:L-1
 m=0:L-1;
 n1=mod(j-m,L);
 hs=hz(n1+1);
 xh=xr.*hs;
 y(i+1,j+1)=sum(xh);
 end
end
yy=[]; %Adding Last M-1 Samples
for i=1:K-1;
 y(i,:)=[y(i,1:N) y(i,N+1:L)+y(i+1,1:M1)];
end
yy=[y(1,1:L) y(2,M:L) y(3,M:L-M1)];
ny=0:length(yy)-1;
subplot(1,3,1);
stem(nxx,xx,'filled');
xlabel('n');
ylabel('x(n)');
title('x(n)');
subplot(1,3,2);
stem(nhz,hz,'filled');
xlabel('n');
ylabel('h(n)');
title('h(n)');
```

```
subplot(1,3,3);
stem(ny,yy,'filled');
xlabel('n');
ylabel('y(n)');
title('y(n)');
```

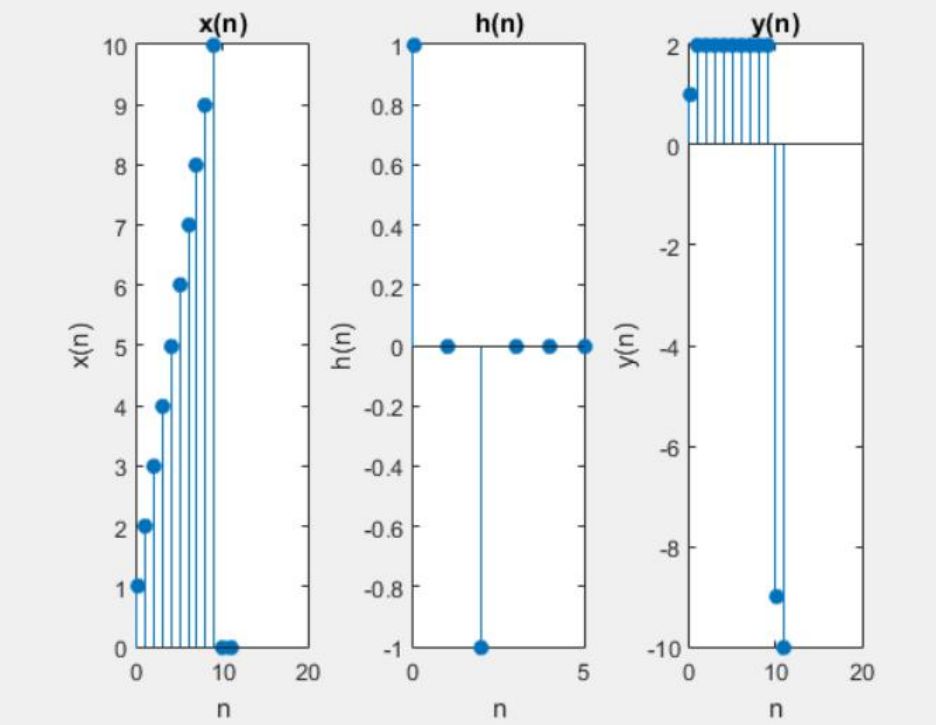
Output:

(i)

```
xi =

     0     0     1     2     3     4


xi =

     3     4     5     6     7     8


xi =

     7     8     9    10     0     0
```



(ii)

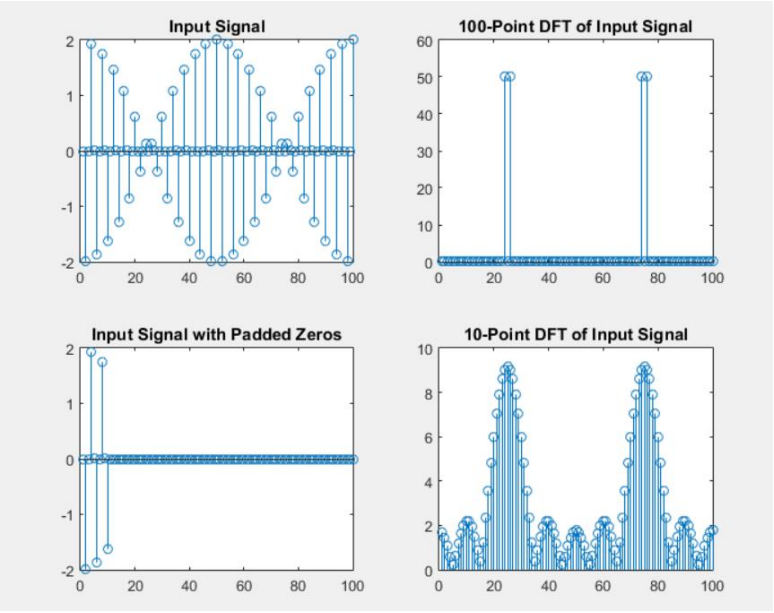


EXPERIMENT 5

MATLAB Code:

```
clc;
close all;

n=1:100;
k=1:100;
x=cos(0.48*pi*n)+cos(0.52*pi*n); %Input Signal
xpad=[x(1:10),zeros(1,90)]; %Input Signal With Padded Zeros
N=100; %N-point DFT
w=2*pi/N;
subplot(2,2,1);
stem(x);
title('Input Signal');
subplot(2,2,2);
stem(abs(x*exp(-1i*(n'*w*k))));
title('100-Point DFT of Input Signal');
subplot(2,2,3);
stem(xpad);
title('Input Signal with Padded Zeros');
subplot(2,2,4);
stem(abs(xpad*exp(-1i*(n'*w*k))));
title('10-Point DFT of Input Signal');
```

Output:

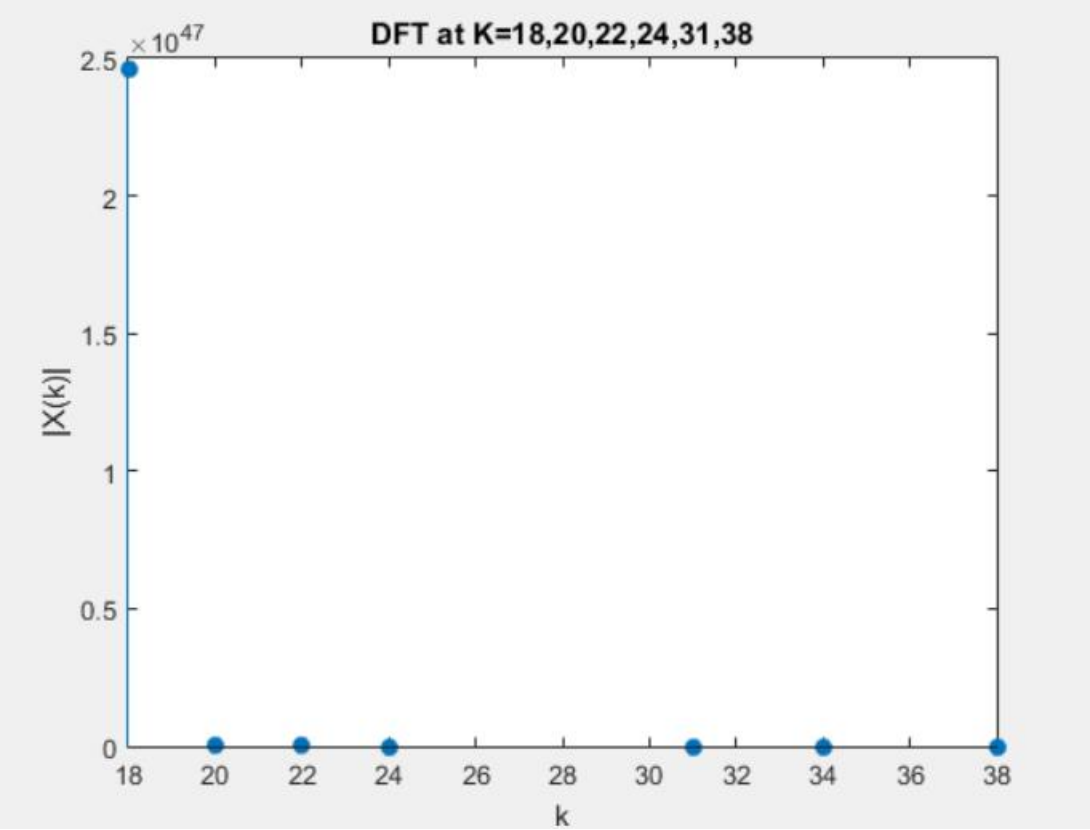

Input Signal
100-Point DFT of Input Signal
Input Signal with Padded Zeros
10-Point DFT of Input Signal

EXPERIMENT 6

MATLAB Code:

```
clc;
close all;

N=205; % DFT length
n=0:N-1;
x=sin(2*pi*n*697/8000)+sin(2*pi*n*1209/8000);
if length(x)<N
    xz=[x zeros(1,N-length(x))];
else
    xz=x;
end
x1=[xz 0];
k=[18 20 22 24 31 34 38];
for i=1:7
    W(i)=exp(-j*2*pi*k(i)/N);
    den(i,: ) = [1 -2*cos(2*pi*k(i)/N)];
    vk(i,:)=filter(1, den(i,:), x1);
    Xk(i)=vk(i,N+1)-W(i)*vk(i,N);
end
stem(k,abs(Xk), 'filled');
xlabel('k');
ylabel('|X(k)|');
title('DFT at K=18,20,22,24,31,38');
```

Output:


DFT at K=18,20,22,24,31,38
×10^47
2.5
2
1.5
1
0.5
0
|X(k)|
18
20
22
24
26
28
30
32
34
36
38
k

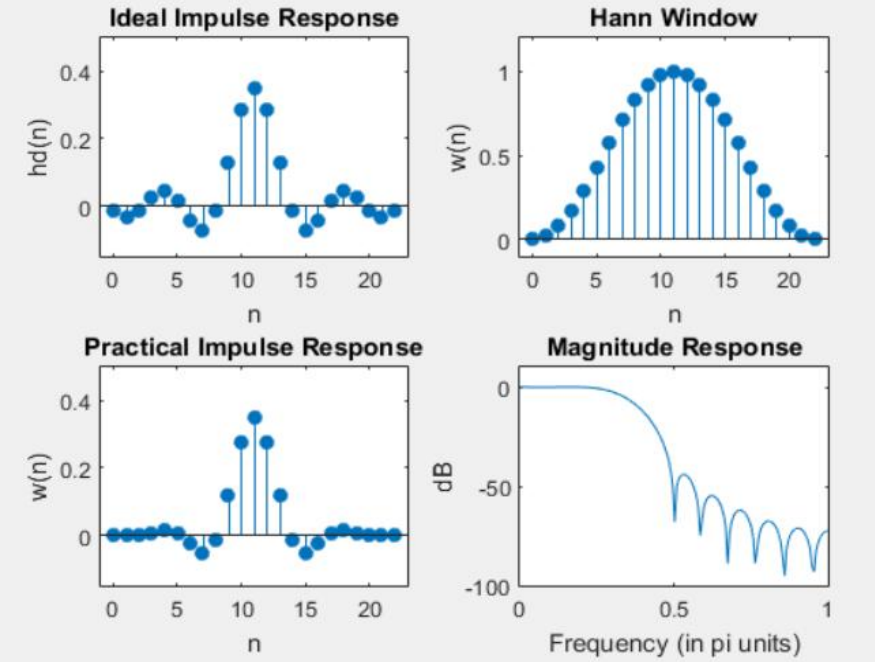
MATLAB Code:

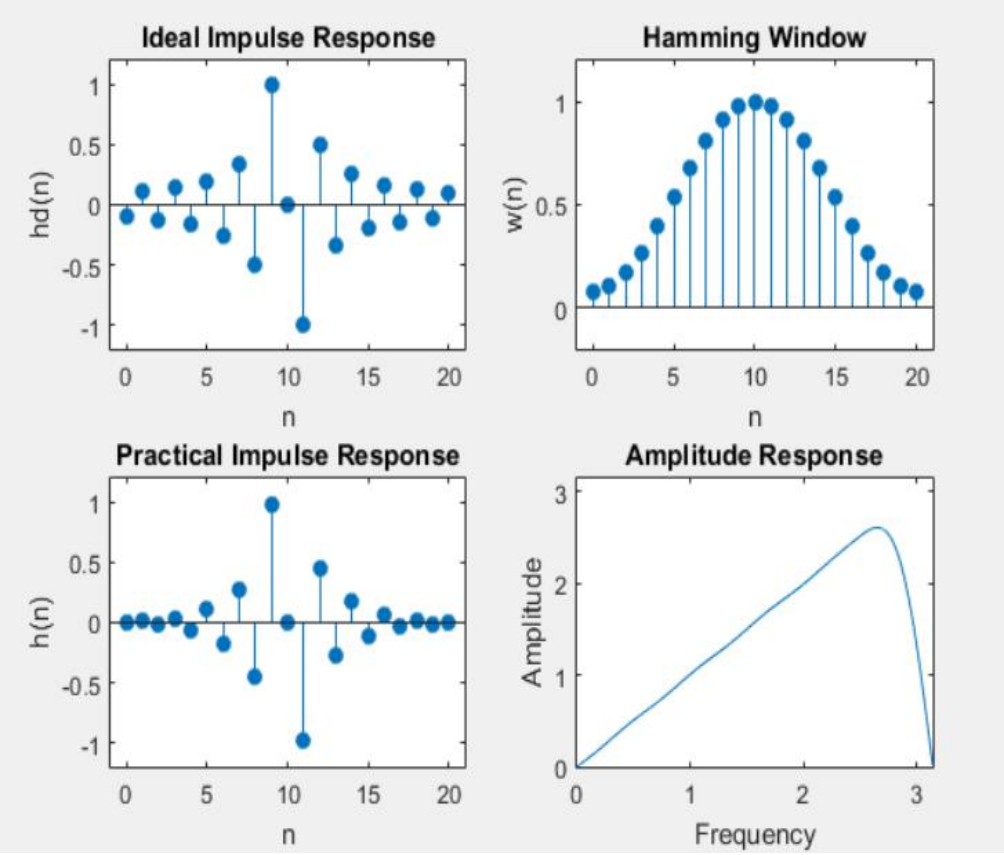
```
clc;
close all;

Fp=2000; %Fp=2KHz
Fs=5000; %Fs=5KHz
Ft=20000; %Ft=20KHz
wp=(2*pi*Fp)/Ft;
ws=(2*pi*Fs)/Ft;
trwidth=ws-wp; %Transition Width
M=ceil(6.2*pi/trwidth)+2; %Filter Length
tau=(M-1)/2;
wc=(wp+ws)/2;
n=0:M-1;
hd=(sin(wc*(n-tau)))./(pi*(n-tau));
hd(tau+1)=0.35;
whan=hann(M)';
h=hd.*whan;
w=0:0.02:pi;
Hw=freqz(h,1,w);
MagHw=abs(Hw); %Magnitude Response
HwdB=20*log10(MagHw/max(MagHw)); %In Decibels
subplot(2,2,1);
stem(n,hd,'filled');
axis([-1 M -0.15 0.5]);
xlabel('n');
ylabel('hd(n)');
title('Ideal Impulse Response');
subplot(2,2,2);
stem(n,whan,'filled');
axis([-1 M -0.1 1.2]);
xlabel('n');
ylabel('w(n)');
title('Hann Window');
subplot(2,2,3);
stem(n,h,'filled');
axis([-1 M -0.15 0.5]);
xlabel('n');
ylabel('w(n)');
title('Practical Impulse Response');
subplot(2,2,4);
plot(w/pi,HwdB);
axis([0 1 -100 10]);
xlabel('Frequency (in pi units)');
ylabel('dB');
title('Magnitude Response');
```

MATLAB Code:

```
clc;
close all;

M=21; %Hamming Window Length=21
tau=(M-1)/2;
n=0:M-1;
hd=((cos(pi*(n-tau)))./(n-tau))-((sin(pi*(n-tau)))./(pi*(n-tau).^2));
hd(tau+1)=0;
whamm=hamming(M)';
h=hd.*whamm;
w=0:0.01:pi;
Hw=freqz(h,1,w);
Hrw=exp(-j*(pi/2-10*w)).*Hw;
subplot(2,2,1);
stem(n,hd,'filled');
axis([-1 M -1.2 1.2]);
xlabel('n');
ylabel('hd(n)');
title('Ideal Impulse Response');
subplot(2,2,2);
stem(n,whamm,'filled');
axis([-1 M -0.2 1.2]);
xlabel('n');
ylabel('w(n)');
title('Hamming Window');
subplot(2,2,3);
stem(n,h,'filled');
axis([-1 M -1.2 1.2]);
xlabel('n');
ylabel('h(n)');
title('Practical Impulse Response');
subplot(2,2,4);
plot(w,Hrw);
axis([0 pi 0 pi]);
xlabel('Frequency');
ylabel('Amplitude');
title('Amplitude Response');
```

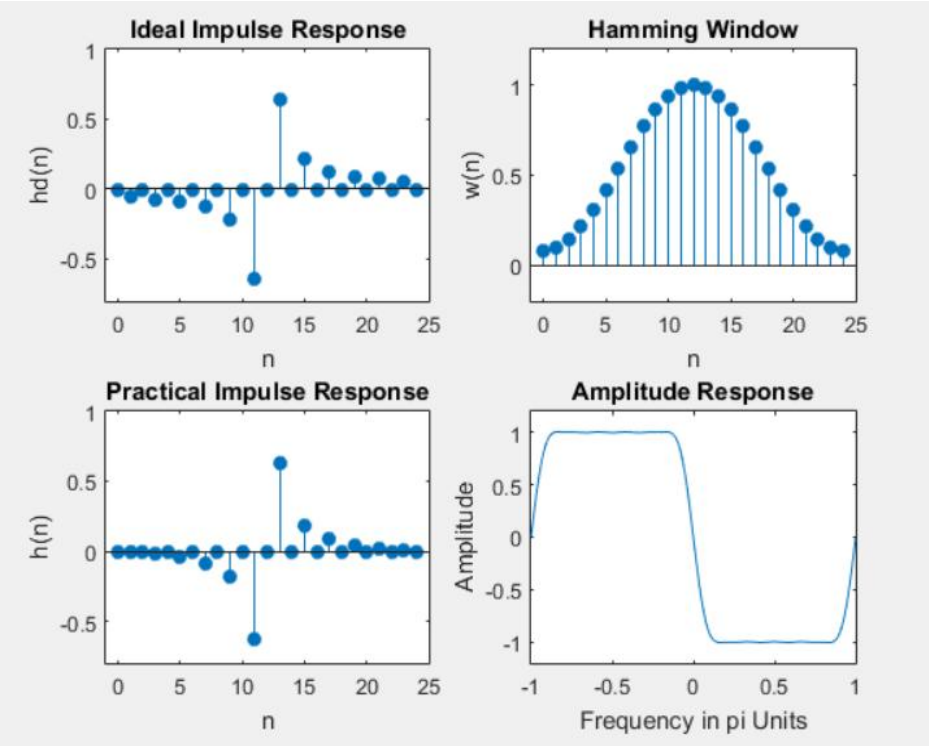
Output:


Ideal Impulse Response
hd(n)
n
Hann Window
w(n)
n
Practical Impulse Response
w(n)
n
Magnitude Response
dB
Frequency (in pi units)


Ideal Impulse Response
hd(n)
n
Hamming Window
w(n)
n
Practical Impulse Response
h(n)
n
Amplitude Response
Amplitude
Frequency

MATLAB Code:

```
clc;
close all;

M=25; %Hamming Window Length=25
tau=(M-1)/2;
n=0:M-1;
hd=(2./(pi*(n-tau))).*(sin(pi*(n-tau)/2).^2);
hd(tau+1)=0;
whamm=hamming(M)';
h=hd.*whamm;
w=-pi:0.01:pi;
Hw=freqz(h,1,w);
Hrw=exp(-j*(pi/2-12*w)).*Hw;
subplot(2,2,1);
stem(n,hd,'filled');
axis([-1 M -0.8 1]);
xlabel('n');
ylabel('hd(n)');
title('Ideal Impulse Response');
subplot(2,2,2);
stem(n,whamm,'filled');
axis([-1 M -0.2 1.2]);
xlabel('n');
ylabel('w(n)');
title('Hamming Window');
subplot(2,2,3);
stem(n,h,'filled');
axis([-1 M -0.8 1]);
xlabel('n');
ylabel('h(n)');
title('Practical Impulse Response');
subplot(2,2,4);
plot(w/pi,Hrw);
axis([-1 1 -1.2 1.2]);
xlabel('Frequency in pi Units');
ylabel('Amplitude');
title('Amplitude Response');
```



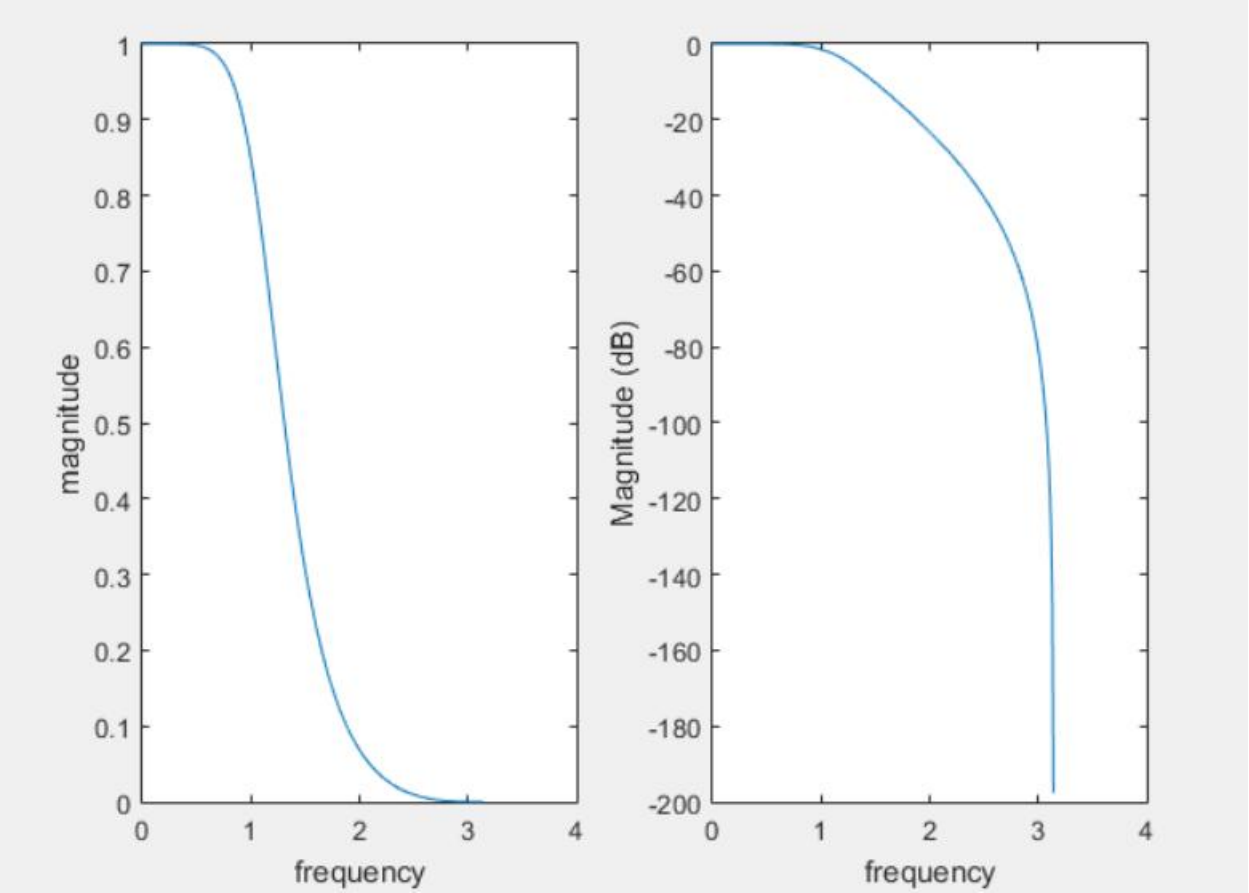
EXPERIMENT 10

MATLAB Code:

```
clc;
close all;

T=1;
wp=0.3*pi;
ws=0.8*pi;
Ap=1;
As=40;

Wp= (2 /T) *tan (wp/ 2) ; %analog pass band edge freq
Ws= (2/T) *tan (ws / 2 ) ; %analog stop band edge freq
R=(10^(0.1*Ap) -1) / (10^(0.1*As)-1) ;

N=ceil ( (1/2)* (log10 (R) / (log10 (Wp/Ws))))
Wc=Wp/((10^(0.1*Ap) -1)^(1/ (2*N)));
[b,a] = butter (N, Wc , 'low' , 's' ) ;

Hs=tf (b,a)
[numd, dend]= bilinear(b, a, 1/T) ;
Hz=tf (numd , dend, T)
w=0:0.01:pi;

Hw=freqz (numd, dend, w) ;
subplot (121) ;
plot (w, abs (Hw)) ;
xlabel ('frequency') ;
ylabel ('magnitude');
subplot (122) ;
plot (w, 20*log10 (abs (Hw))) ;
xlabel('frequency');
ylabel('Magnitude (dB)');
```

Output:

The results are N=3 and Wc=1.2764


magnitude
frequency
Magnitude (dB)
frequency

EXPERIMENT 11

MATLAB Code:

1.

```
clc;
clear all;

x=3/8;
B=2;
xl=abs(x);
Qx=0;
for k=1:13
    Qxbeq(k)=fix(xl*2);
    Qx=fix(xl*2)/(2^k)+Qx;
    x1=(xl*2)-fix(xl*2);
end
sg=sign(x);
if sg>0
    Qxbeq =[0 Qxbeq];
else
    Qxbeq =[1 Qxbeq];
end

disp('Truncated Number:')
disp(Qx);
disp('Binary equivalent of truncated number:')
disp (Qxbeq)
```

2.

```
clc;
clear all;

x=3/8;
B=2;
x1= abs(x);
Qx=0;
x1= x1 +(1/2)*2^(-B);

for k=1:B
    Qxbeq(k)=fix(x1*2);
    Qx= fix(x1*2)/(2^k) +Qx;
    x1=(x1*2)-fix(x1*2);
end
sg=sign(x);
if sg>0
    Qxbeq =[0 Qxbeq];
else
    Qxbeq =[1 Qxbeq];
end
disp('Rounded Humber:')
disp(Qx)
```

```
disp('Binary equivalent of rounded number:')
disp(Qxbeq)
```

Output:

Truncated number: 0.2500
Binary equivalent of quantized number:
0 0 1

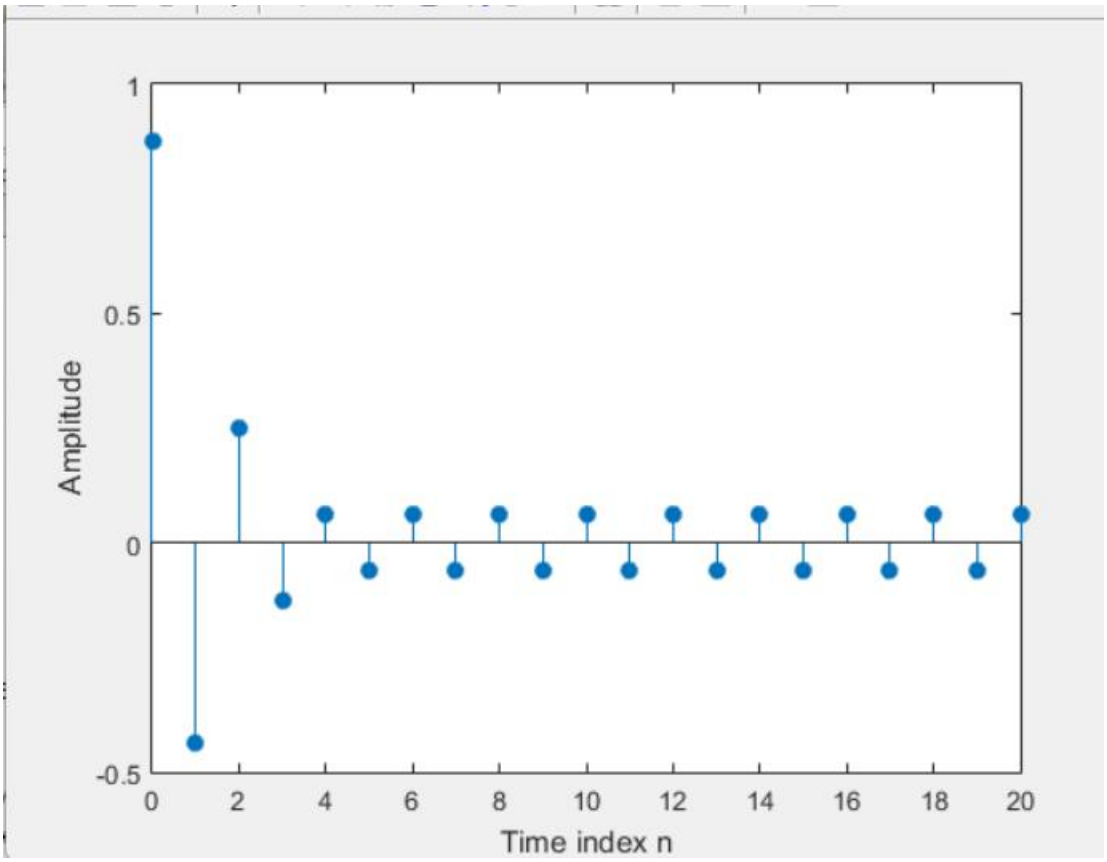
Rounded number: 0.5000
Binary equivalent of rounded number:
0 1 0

EXPERIMENT 12

MATLAB Code:

```
clc;
close all;

a=-1/2;
B=4;
x=[7/8 zeros(1,20)];

ycap=0;

for n=1:21
    ay=abs(a*ycap);
    ay=ay+(1/2)*2^(-B);
    Qy=0;
    for k=1:B
        Qy=fix(ay*2)/(2^k)+Qy;
        ay=(ay*2)-fix(ay*2);
    end
    Qy = sign(a*ycap)* Qy;
    y(n)=Qy+x(n);
    ycap=y(n);
end
k=0:20;
stem(k,y,'filled')
ylabel('Amplitude');
xlabel('Time index n')
```

Output:


1
0.5
0
-0.5
Amplitude
0
2
4
6
8
10
12
14
16
18
20
Time index n