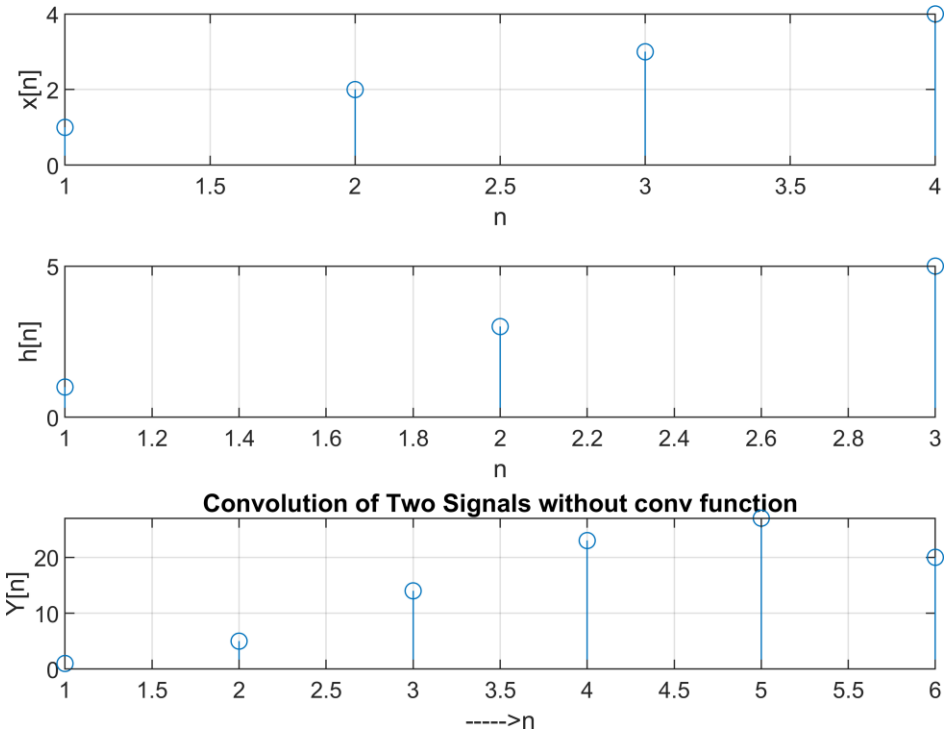
LINEAR CONVOLUTION (4B9)

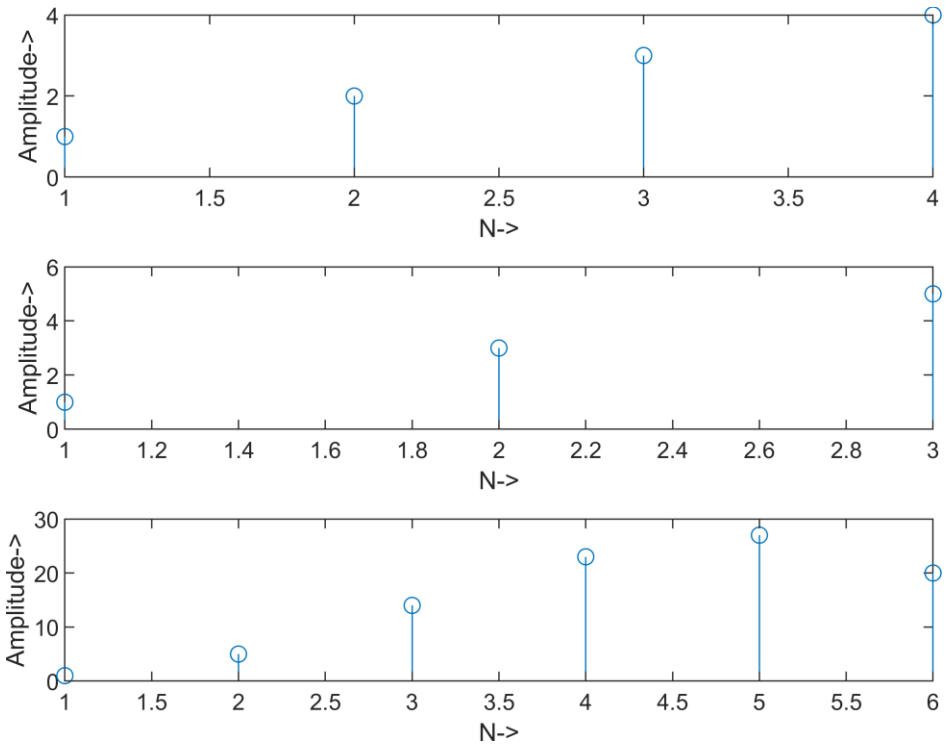
```
% LINEAR CONVOLUTION WITHOUT USING CONV FUNCTION
clear all;
close all;
x=input('Enter the sequence 1:');
h=input('Enter the sequence 2:');
% convolution
m=length(x);
n=length(h);
X=[x,zeros(1,n)];
H=[h,zeros(1,m)];
for i=1:n+m-1
    Y(i)=0;
    for j=1:m
        if(i-j+1>0)
            Y(i)=Y(i)+X(j)*H(i-j+1);
        else
        end
    end
end
% plot results
figure;
subplot(3,1,1); stem(x); xlabel('n');
ylabel('x[n]'); grid on;
subplot(3,1,2);  stem(h);
xlabel('n'); ylabel('h[n]'); grid on;
subplot(3,1,3); stem(Y);
ylabel('Y[n]'); xlabel('----->n'); grid on;
title('Convolution of Two Signals without conv function');
```




```
disp(Y)
```

```
     1     5    14    23    27    20
```

```
% LINEAR CONVOLUTION USING CONV FUNCTION
clear all;
close all;
x=input('Enter the sequence 1:');
h=input('Enter the sequence 2:');
y=conv(x,h);
figure;
subplot(3,1,1);
stem(x);
ylabel('Amplitude->');
xlabel('N->');
subplot(3,1,2);
stem(h);
ylabel('Amplitude->');
xlabel('N->');
subplot(3,1,3);
stem(y);
ylabel('Amplitude->');
xlabel('N->');
```




```
disp('The resultant signals:');
```

```
The resultant signals:
```

```
disp(y)
```

```
     1     5    14    23    27    20
```

EXPERIMENT-1 : CIRCULAR CONVOLUTION (4B9)

```
clear  all;
close  all;
x=[1,2,3,4]
```

```
x = 1×4
     1     2     3     4
```

```
h=[4,3,2,1]
```

```
h = 1×4
     4     3     2     1
```

```
for a=1:4
 X(:,a)=x
 x=[x(end) x(1:1:end-1)]
end
```

```
X = 4×1
     1
     2
     3
     4
x = 1×4
     4     1     2     3
X = 4×2
     1     4
     2     1
     3     2
     4     3
x = 1×4
     3     4     1     2
X = 4×3
     1     4     3
     2     1     4
     3     2     1
     4     3     2
x = 1×4
     2     3     4     1
X = 4×4
     1     4     3     2
     2     1     4     3
     3     2     1     4
     4     3     2     1
x = 1×4
     1     2     3     4
```

```
H=h'
```

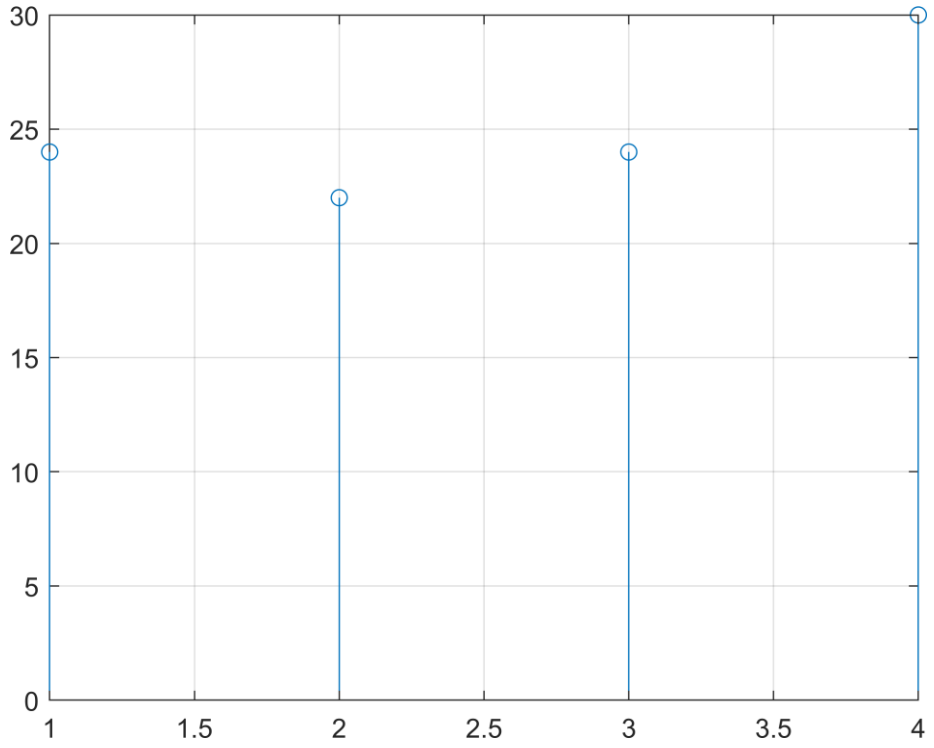
```
H = 4×1
     4
     3
     2
     1
```

```
r=X*H
```

```
r = 4×1
    24
```

```
    22
    24
    30
```

```
stem(r);
grid on;
```



EXPERIMENT-2 DISCRETE FOURIER TRANSFORM/ INVERSE DISCRETE FOURIER TRANSFORM (4B9)

```
% DFT USING MATHEMATICAL EQUATION
clear all;
close all;
x=input("enter the sequence ")
```

```
x = 1×4
     1     2     3     4
```

```
N=length(x)
```

```
N = 4
```

```
X=zeros(1,N)
```

```
X = 1×4
     0     0     0     0
```

```
for k=0:N-1
    for n=0:N-1
        X(k+1)=X(k+1)+x(n+1)*(exp(-1j*(2*pi/N)*k*n))
    end
end
```

```
X = 1×4
     1     0     0     0
X = 1×4
     3     0     0     0
X = 1×4
     6     0     0     0
X = 1×4
    10     0     0     0
X = 1×4
    10     1     0     0
X = 1×4 complex
  10.0000 + 0.0000i   1.0000 - 2.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
X = 1×4 complex
  10.0000 + 0.0000i  -2.0000 - 2.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
X = 1×4 complex
  10.0000 + 0.0000i  -2.0000 + 2.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
X = 1×4 complex
  10.0000 + 0.0000i  -2.0000 + 2.0000i   1.0000 + 0.0000i   0.0000 + 0.0000i
X = 1×4 complex
  10.0000 + 0.0000i  -2.0000 + 2.0000i  -1.0000 - 0.0000i   0.0000 + 0.0000i
X = 1×4 complex
  10.0000 + 0.0000i  -2.0000 + 2.0000i   2.0000 + 0.0000i   0.0000 + 0.0000i
X = 1×4 complex
  10.0000 + 0.0000i  -2.0000 + 2.0000i  -2.0000 - 0.0000i   0.0000 + 0.0000i
X = 1×4 complex
  10.0000 + 0.0000i  -2.0000 + 2.0000i  -2.0000 - 0.0000i   1.0000 + 0.0000i
X = 1×4 complex
  10.0000 + 0.0000i  -2.0000 + 2.0000i  -2.0000 - 0.0000i   1.0000 + 2.0000i
X = 1×4 complex
  10.0000 + 0.0000i  -2.0000 + 2.0000i  -2.0000 - 0.0000i  -2.0000 + 2.0000i
X = 1×4 complex
  10.0000 + 0.0000i  -2.0000 + 2.0000i  -2.0000 - 0.0000i  -2.0000 - 2.0000i
```

```
% DFT USING MATRIX METHOD
clear all;
close all;
x=input("enter the sequence")
```

```
x = 1×4
     1     2     3     4
```

```
N=length(x)
```

```
N = 4
```

```
for k=0:N-1
    for n=0:N-1
       D(k+1,n+1)=exp(-1j*(2*pi/4)*k*n)
    end
end
```

```
D = 1
D = 1×2
     1     1
D = 1×3
     1     1     1
D = 1×4
     1     1     1     1
D = 2×4
     1     1     1     1
     1     0     0     0
D = 2×4 complex
   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i
   1.0000 + 0.0000i   0.0000 - 1.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
D = 2×4 complex
   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i
   1.0000 + 0.0000i   0.0000 - 1.0000i  -1.0000 - 0.0000i   0.0000 + 0.0000i
D = 2×4 complex
   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i
   1.0000 + 0.0000i   0.0000 - 1.0000i  -1.0000 - 0.0000i  -0.0000 + 1.0000i
D = 3×4 complex
   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i
   1.0000 + 0.0000i   0.0000 - 1.0000i  -1.0000 - 0.0000i  -0.0000 + 1.0000i
   1.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
D = 3×4 complex
   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i
   1.0000 + 0.0000i   0.0000 - 1.0000i  -1.0000 - 0.0000i  -0.0000 + 1.0000i
   1.0000 + 0.0000i  -1.0000 - 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
D = 3×4 complex
   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i
   1.0000 + 0.0000i   0.0000 - 1.0000i  -1.0000 - 0.0000i  -0.0000 + 1.0000i
   1.0000 + 0.0000i  -1.0000 - 0.0000i   1.0000 + 0.0000i   0.0000 + 0.0000i
D = 3×4 complex
   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i
   1.0000 + 0.0000i   0.0000 - 1.0000i  -1.0000 - 0.0000i  -0.0000 + 1.0000i
   1.0000 + 0.0000i  -1.0000 - 0.0000i   1.0000 + 0.0000i  -1.0000 - 0.0000i
D = 4×4 complex
   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i
   1.0000 + 0.0000i   0.0000 - 1.0000i  -1.0000 - 0.0000i  -0.0000 + 1.0000i
   1.0000 + 0.0000i  -1.0000 - 0.0000i   1.0000 + 0.0000i  -1.0000 - 0.0000i
   1.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
D = 4×4 complex
   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i
   1.0000 + 0.0000i   0.0000 - 1.0000i  -1.0000 - 0.0000i  -0.0000 + 1.0000i
```

```
   1.0000 + 0.0000i  -1.0000 - 0.0000i   1.0000 + 0.0000i  -1.0000 - 0.0000i
   1.0000 + 0.0000i  -0.0000 + 1.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
D = 4×4 complex
   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i
   1.0000 + 0.0000i   0.0000 - 1.0000i  -1.0000 - 0.0000i  -0.0000 + 1.0000i
   1.0000 + 0.0000i  -1.0000 - 0.0000i   1.0000 + 0.0000i  -1.0000 - 0.0000i
   1.0000 + 0.0000i  -0.0000 + 1.0000i  -1.0000 - 0.0000i   0.0000 + 0.0000i
D = 4×4 complex
   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i
   1.0000 + 0.0000i   0.0000 - 1.0000i  -1.0000 - 0.0000i  -0.0000 + 1.0000i
   1.0000 + 0.0000i  -1.0000 - 0.0000i   1.0000 + 0.0000i  -1.0000 - 0.0000i
   1.0000 + 0.0000i  -0.0000 + 1.0000i  -1.0000 - 0.0000i   0.0000 - 1.0000i
```

```
disp(D*x')      %DFT
```

```
  10.0000 + 0.0000i
  -2.0000 + 2.0000i
  -2.0000 - 0.0000i
  -2.0000 - 2.0000i
```

```
disp((D^-1)*x') %IDFT
```

```
   2.5000 - 0.0000i
  -0.5000 - 0.5000i
  -0.5000 + 0.0000i
  -0.5000 + 0.5000i
```

```
%DFT USING FUNCTION
clear all;
close all;
x=input("enter the sequence")
```

```
x = 1×4
     1     2     3     4
```

```
N=length(x)
```

```
N = 4
```

```
a=anirudh(N)
```

```
Unrecognized function or variable 'anirudh'.
```

```
disp(a*x')
```

```
% IDFT USING MATRIX METHOD
close all;
x=input("enter the sequence")
N=length(x)
X=zeros(1,N)
for k=0:N-1
    for n=0:N-1
        X(k+1)=X(k+1)+(1/N)*x(n+1)*(exp(1j*(2*pi/N)*k*n))
    end
end
```

EXPERIMENT-3 POWER DENSITY SPECTRUM (4B9)

```
close all,
clear all,
clc,
x=[1,1,1,1,0,0,0,0]
```

```
x = 1×8
     1     1     1     1     0     0     0     0
```

```
ns=length(x)
```

```
ns = 8
```

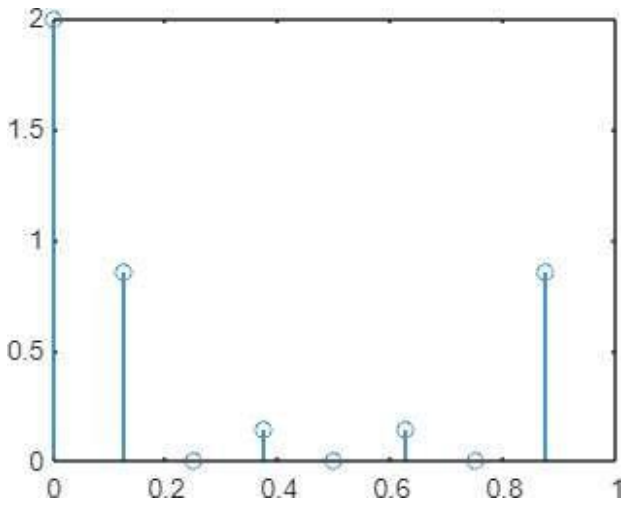
```
N=8
```

```
N = 8
```

```
powerspectraldensity=abs(fft(x,N)).^2/ns
```

```
powerspectraldensity = 1×8
    2.0000    0.8536         0    0.1464         0    0.1464         0    0.8536
```

```
stem((0:N-1)/N, powerspectraldensity)
```




```
close all,
clear all,
clc,
x=[1,1,1,1,0,0,0,0]
```

```
x = 1×8
     1     1     1     1     0     0     0     0
```

```
ns=length(x)
```

```
ns = 8
```

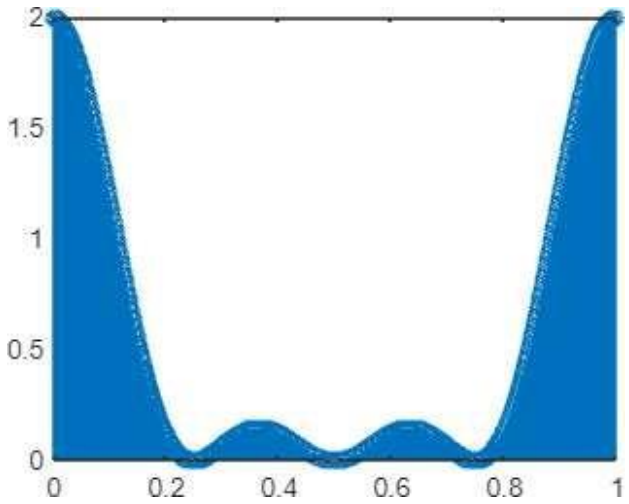
```
N=1024
```

```
N = 1024
```

```
powerspectraldensity=abs(fft(x,N)).^2/ns
```

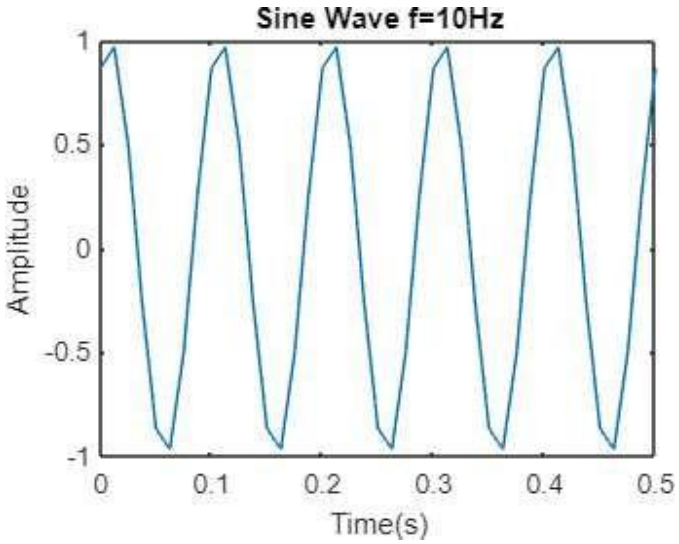
```
powerspectraldensity = 1×1024
    2.0000    1.9999    1.9996    1.9992    1.9985    1.9976    1.9966    1.9954 ···
```

```
stem((0:N-1)/N, powerspectraldensity)
```

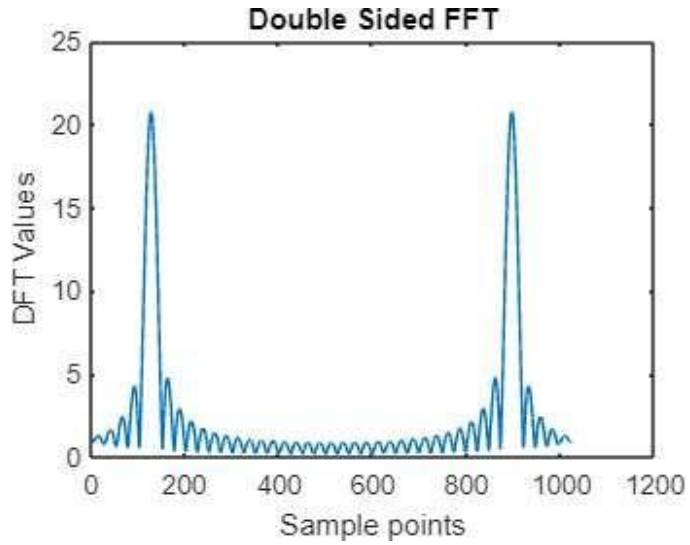


Frequency

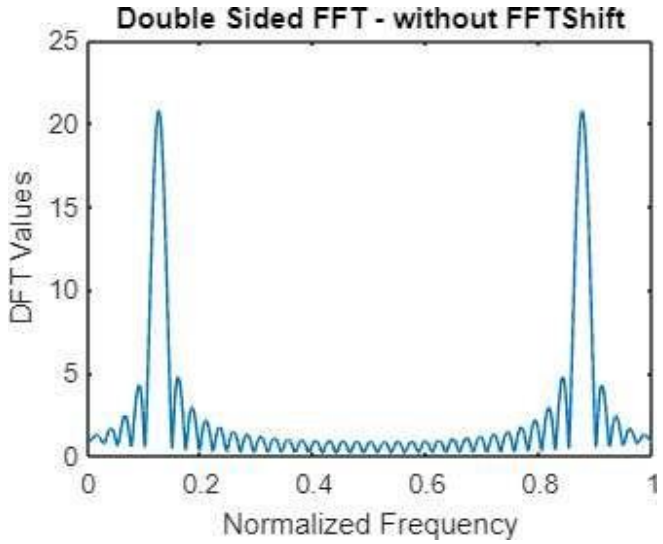
```
f=10;
overSampRate=20;
% fs=overSampRate*f;
fs=80;
phase = 1/3*pi;
nCyl = 5;
t=0:1/fs:nCyl*1/f;
x=sin(2*pi*f*t+phase);
plot(t,x);
title(['Sine Wave f=', num2str(f), 'Hz']);
xlabel('Time(s)');
ylabel('Amplitude');
```



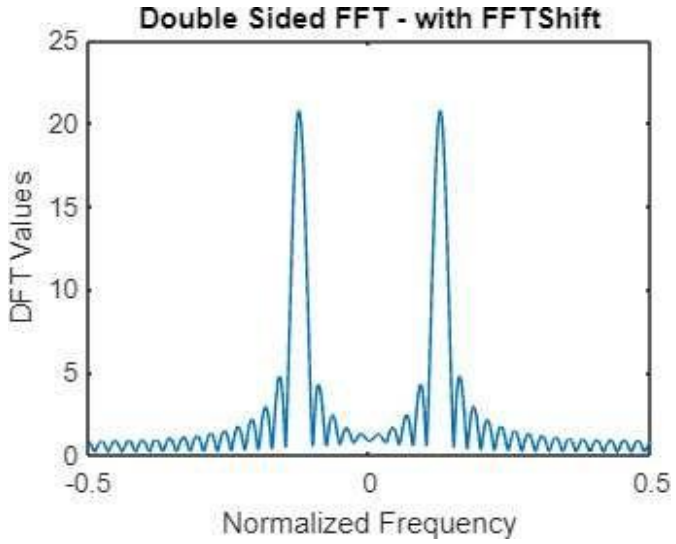

```
NFFT=1024;
X=fft(x,NFFT);
nVals=0:NFFT-1;
plot(nVals,abs(X));
title('Double Sided FFT');
xlabel('Sample points')
ylabel('DFT Values');
```




```
NFFT=1024;
X=fft(x,NFFT);
nVals=(0:NFFT-1)/NFFT;
plot(nVals,abs(X));
title('Double Sided FFT - without FFTShift');
xlabel('Normalized Frequency')
ylabel('DFT Values');
```

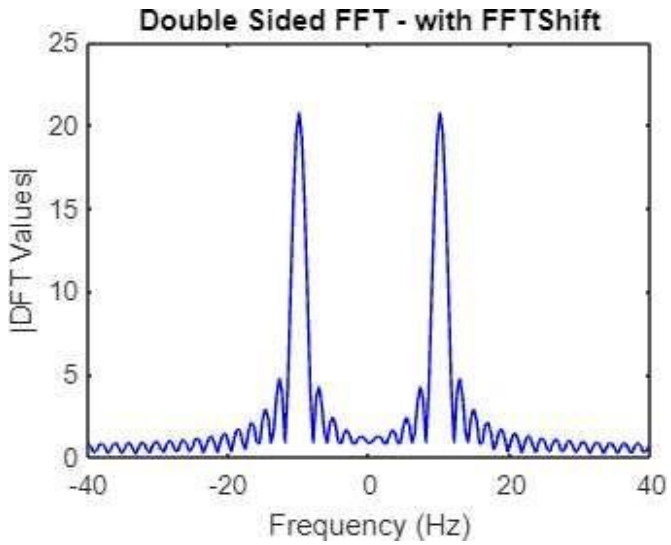



```
NFFT=1024;
X=fftshift(fft(x,NFFT));
% X=fft(x, NFFT);
fVals=(-NFFT/2:NFFT/2-1)/NFFT;
plot(fVals,abs(X));
title('Double Sided FFT - with FFTShift');
xlabel('Normalized Frequency')
ylabel('DFT Values');
```




```
NFFT=200;
X=fftshift(abs(fft(x,NFFT)));
fVals=fs*(-NFFT/2:NFFT/2-1)/NFFT;
plot(fVals,X,'b');
title('Double Sided FFT - with FFTShift');
xlabel('Frequency (Hz)')
```

```
ylabel('|DFT Values|');
```



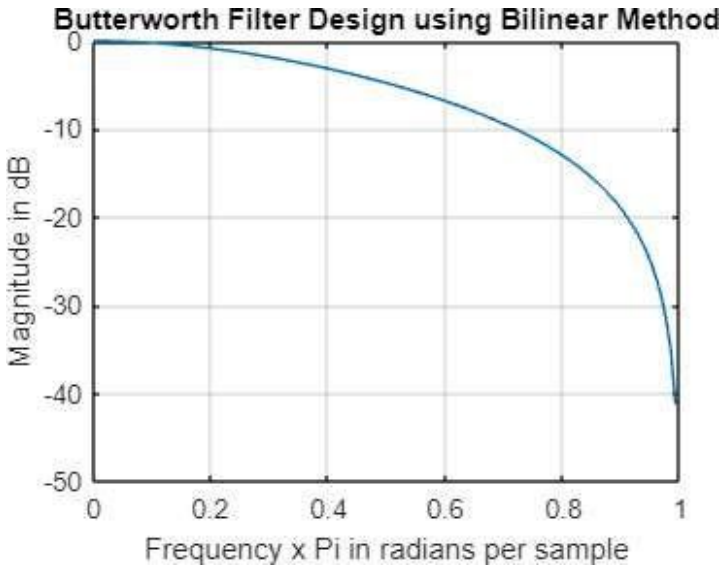
EXPERIMENT-4 IMPLEMENTATION OF FILTERS USING IIR (4B9)

```
clear all;
close all;
clc;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
delta1 = input('Enter the Pass Band Ripple');
delta2 = input('Enter the Stop Band Ripple');
wp = input('Enter the Digital Pass Band Edge Frequency');
ws = input('Enter the Digital Stop Band Edge Frequency');
T = input('Enter the Sampling Time Period in Seconds');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Analog Frequencies Calculated using Bilinear Transformation
omegap = (2/T)*tan(wp/2);
omegas = (2/T)*tan(ws/2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%To Calculate the filter order
den = 2*log10(omegas/omegap);
delta = ((1/(delta2^2))-1);
epsi  = ((1/(delta1^2))-1);
num = log10(delta/epsi);
N = ceil(num/den);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%To Calculate the Analog Cut off frequency
disp('If LPF enter 1, If HPF enter 2, If BPF enter 3, If BSF enter 4');
```

```
If LPF enter 1, If HPF enter 2, If BPF enter 3, If BSF enter 4
```

```
type = input('enter the type of the filter you want to design');
switch type
    case 1
        omegac = omegap/(epsi^(1/(2*N)));
        wc = 2*atan(omegac/2);
        [b,a] = butter(N,wc/pi);
    case 2
        omegac = omegap/(epsi^(1/(2*N)));
        wc = 2*atan(omegac/2);
        [b,a] = butter(N,wc/pi,'high');
    case 3
        omegac1 = omegap/(epsi^(1/(2*N)));
        omegac2 = omegas/(delta^(1/(2*N)));
        wc1 = 2*atan(omegac1/2);
        wc2 = 2*atan(omegac2/2);
        wc = [wc1,wc2]
        [b,a] = butter(N,wc/pi);

        case 4
        omegac1 = omegap/(epsi^(1/(2*N)));
        omegac2 = omegas/(delta^(1/(2*N)));
```

```
        wc1 = 2*atan(omegac1/2);
        wc2 = 2*atan(omegac2/2);
        wc = [wc1,wc2]
        [b,a] = butter(N,wc/pi,'stop');
    otherwise
        disp('The Type Entered is not a valid filter');
end
[H,W] = freqz(b,a,128);
plot(W/pi,20*log10(abs(H)));
grid on
xlabel('Frequency x Pi in radians per sample')
ylabel('Magnitude in dB')
title('Butterworth Filter Design using Bilinear Method')
```

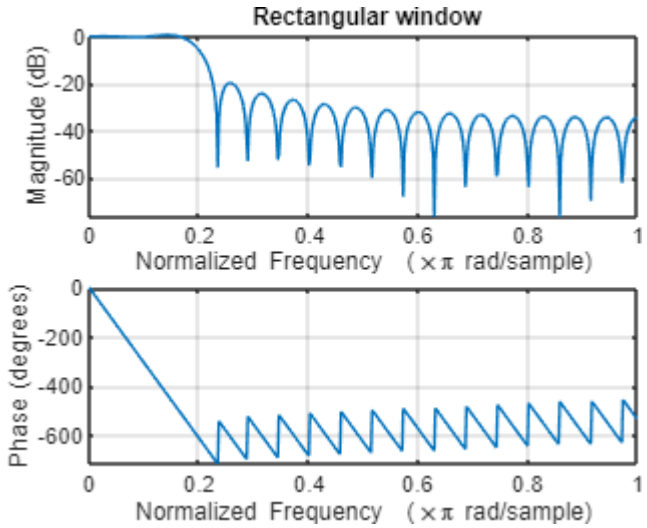


EXPERIMENT-5 IMPLEMENTATION OF FILTERS USING FIR (4B9)

```
%FIR filter using Rectangular window
clear all;
close all;
b=fir1(34,0.2,"low",rectwin(35))
```

```
b = 1×35
   -0.0177   -0.0116    0.0000    0.0133    0.0232    0.0251    0.0169   -0.0000 ···
```

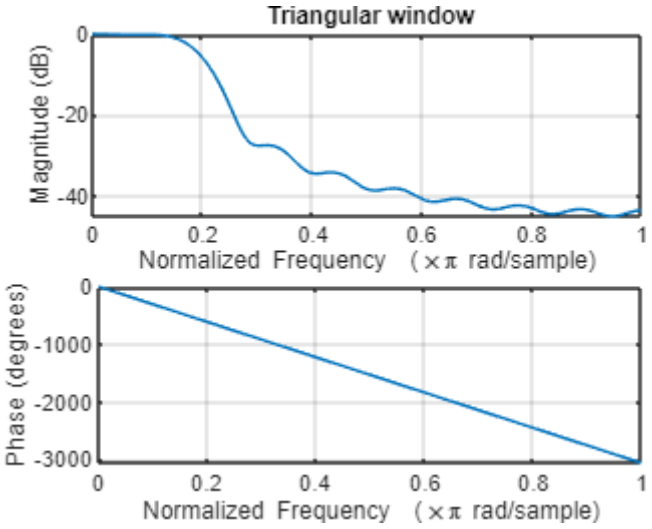
```
freqz(b,1)
title(' Rectangular window');
```




```
%FIR filter using Triangular window
clear all;
close all;
b=fir1(34,0.2,"low",triang(35))
```

```
b = 1×35
   -0.0010   -0.0014    0.0000    0.0031    0.0068    0.0088    0.0070   -0.0000 ···
```

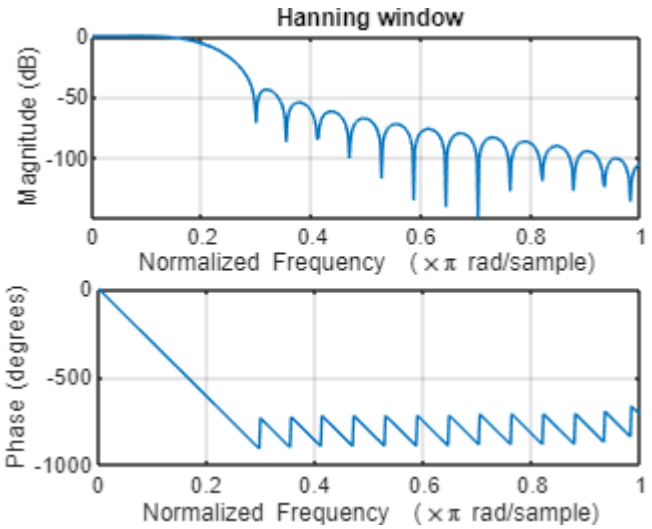
```
freqz(b,1)
title(' Triangular window');
```




```
%FIR filter using Hanning window
clear all;
close all;
b=fir1(34,0.2,"low",hann(35))
```

```
b = 1×35
         0   -0.0001    0.0000    0.0010    0.0030    0.0050    0.0047   -0.0000 ···
```

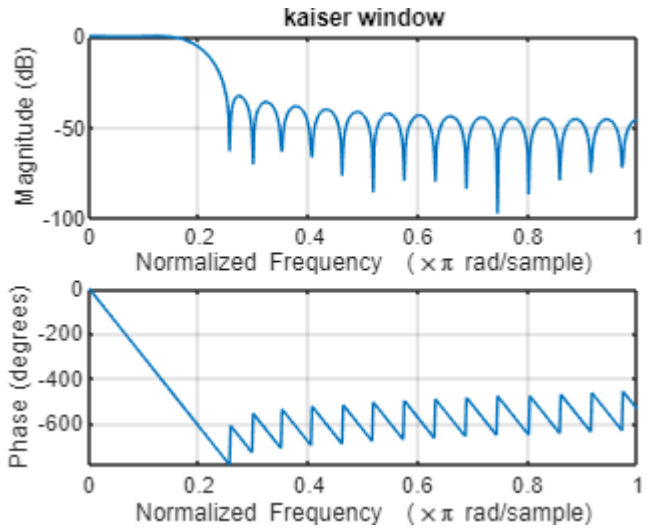
```
freqz(b,1)
title(' Hanning window');
```




```
%FIR filter using Kaiser window
clear all;
close all;
b=fir1(34,0.2,"low",kaiser(35,2.5))
```

```
b = 1×35
   -0.0054   -0.0042    0.0000    0.0063    0.0124    0.0149    0.0110   -0.0000 ···
```

```
freqz(b,1)
title('kaiser window');
```



EXPERIMENT-6 GENERATION OF SINUSOIDAL SIGNAL THROUGH FILTERING
(4B9)

```
% SINE WAVE WITHOUT USING FILTER
clear all;
close all;
f=1000;
fs=8000;
w=2*pi*f/fs;
b0=sin(w);
a1=2*cos(w);
a2=-1;
y1=0;
y2=0;
x1=1;
% increase no.of samples to get better result(i.e. 50 to 200)
y=zeros(1,50);
for n=1:50
    y(n)=b0*x1+a1*y1+a2*y2
    x1=0
    y2=y1
    y1=y(n)
end
```

```
y = 1×50
    0.7071         0         0         0         0         0         0         0 ···
x1 = 0
y2 = 0
y1 = 0.7071
y = 1×50
    0.7071    1.0000         0         0         0         0         0         0 ···
x1 = 0
y2 = 0.7071
y1 = 1
y = 1×50
    0.7071    1.0000    0.7071         0         0         0         0         0 ···
x1 = 0
y2 = 1
y1 = 0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000         0         0         0         0 ···
x1 = 0
y2 = 0.7071
y1 = 2.2204e-16
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071         0         0         0 ···
x1 = 0
y2 = 2.2204e-16
y1 = -0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000         0         0 ···
x1 = 0
y2 = -0.7071
y1 = -1
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071         0 ...
x1 = 0
y2 = -1
y1 = -0.7071
```

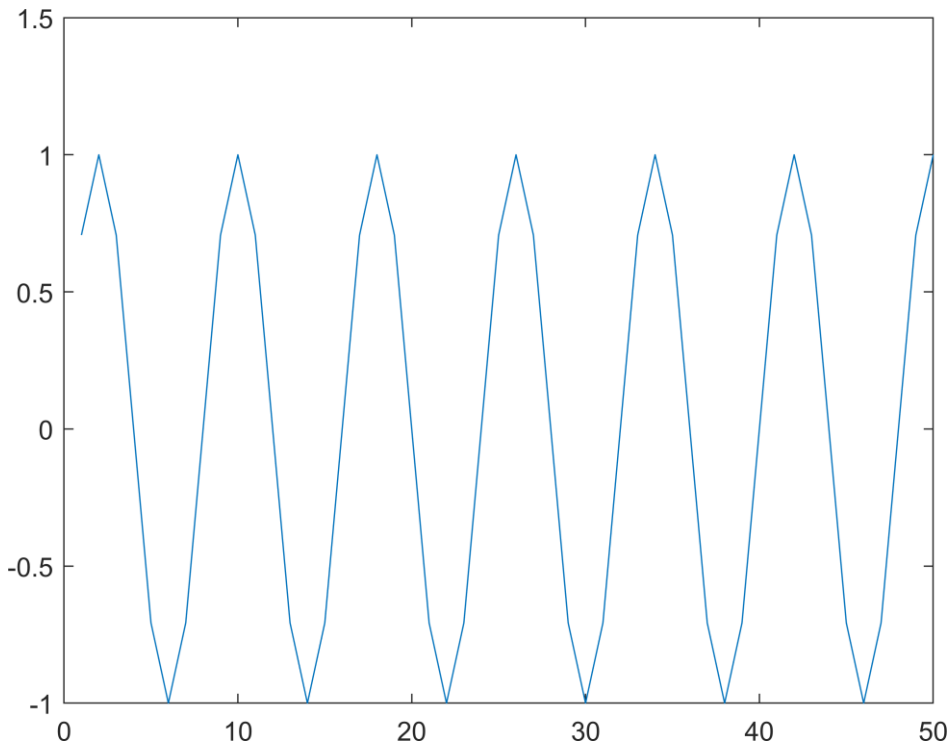
```
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = -0.7071
y1 = -4.4409e-16
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = -4.4409e-16
y1 = 0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = 0.7071
y1 = 1
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = 1
y1 = 0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = 0.7071
y1 = 6.6613e-16
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = 6.6613e-16
y1 = -0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = -0.7071
y1 = -1.0000
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = -1.0000
y1 = -0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = -0.7071
y1 = -8.8818e-16
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = -8.8818e-16
y1 = 0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = 0.7071
y1 = 1.0000
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = 1.0000
y1 = 0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = 0.7071
```

```
y1 = 1.1102e-15
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = 1.1102e-15
y1 = -0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = -0.7071
y1 = -1
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = -1
y1 = -0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = -0.7071
y1 = -1.3323e-15
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = -1.3323e-15
y1 = 0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = 0.7071
y1 = 1.0000
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = 1.0000
y1 = 0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = 0.7071
y1 = 1.4433e-15
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = 1.4433e-15
y1 = -0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = -0.7071
y1 = -1
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = -1
y1 = -0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = -0.7071
y1 = -1.7764e-15
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
```

```
y2 = -1.7764e-15
y1 = 0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = 0.7071
y1 = 1.0000
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = 1.0000
y1 = 0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = 0.7071
y1 = 1.8874e-15
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = 1.8874e-15
y1 = -0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = -0.7071
y1 = -1
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = -1
y1 = -0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = -0.7071
y1 = -2.2204e-15
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = -2.2204e-15
y1 = 0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = 0.7071
y1 = 1
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = 1
y1 = 0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = 0.7071
y1 = 2.4425e-15
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
x1 = 0
y2 = 2.4425e-15
y1 = -0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000···
```

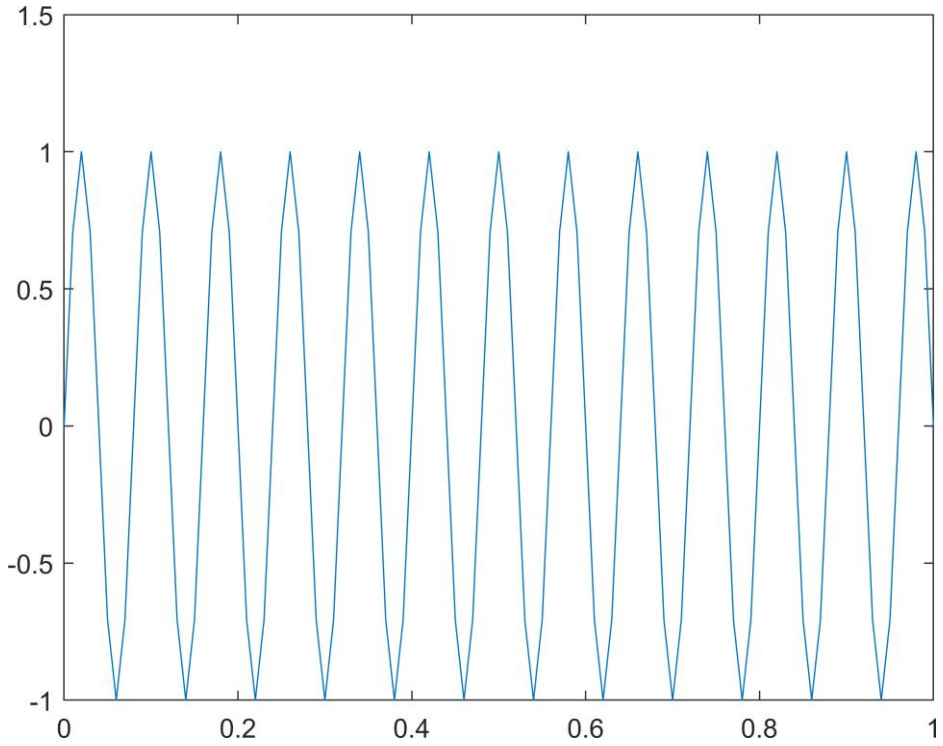
```
x1 = 0
y2 = -0.7071
y1 = -1
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000 ···
x1 = 0
y2 = -1
y1 = -0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000 ···
x1 = 0
y2 = -0.7071
y1 = -2.6645e-15
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000 ···
x1 = 0
y2 = -2.6645e-15
y1 = 0.7071
y = 1×50
    0.7071    1.0000    0.7071    0.0000   -0.7071   -1.0000   -0.7071   -0.0000 ···
x1 = 0
y2 = 0.7071
y1 = 1
```

```
plot(1:length(y),y)
```

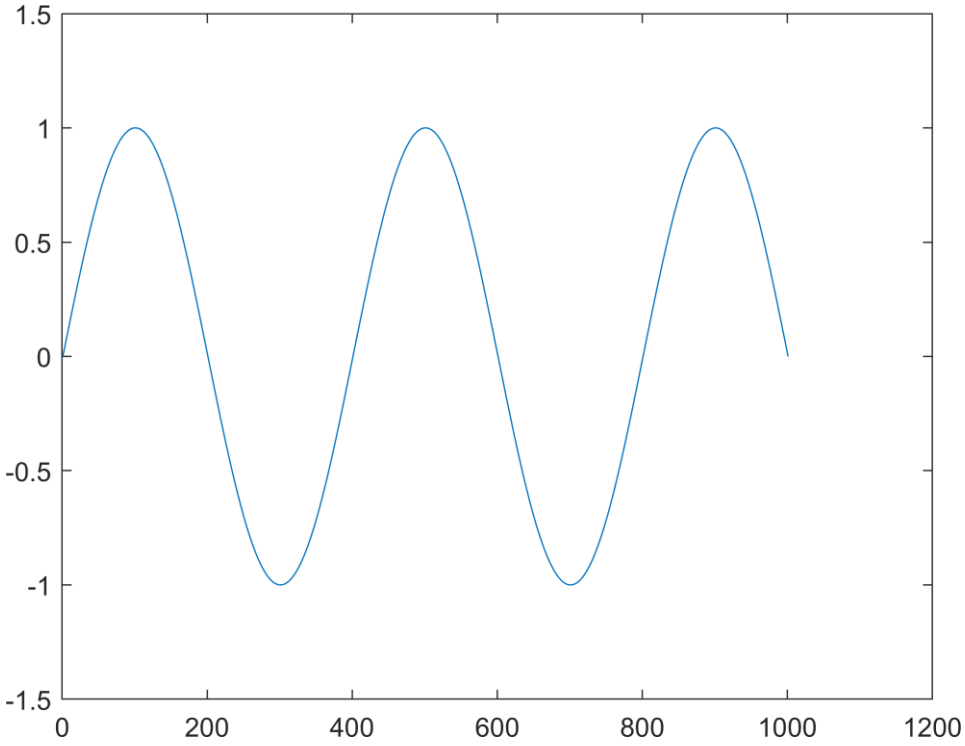



```
% SINE WAVE USING FILTER
% METHOD 1
close all;
clear all;
clc;
```

```
t=0:0.01:1;
f=1000;
fs=8000;
y=zeros(1,length(t));
y(1)=1;
b=[0 sin(2*pi*(f/fs)) 0];
a=[1 -2*cos(2*pi*(f/fs)) 1];
x=filter(b,a,y);
plot(t,x)
```



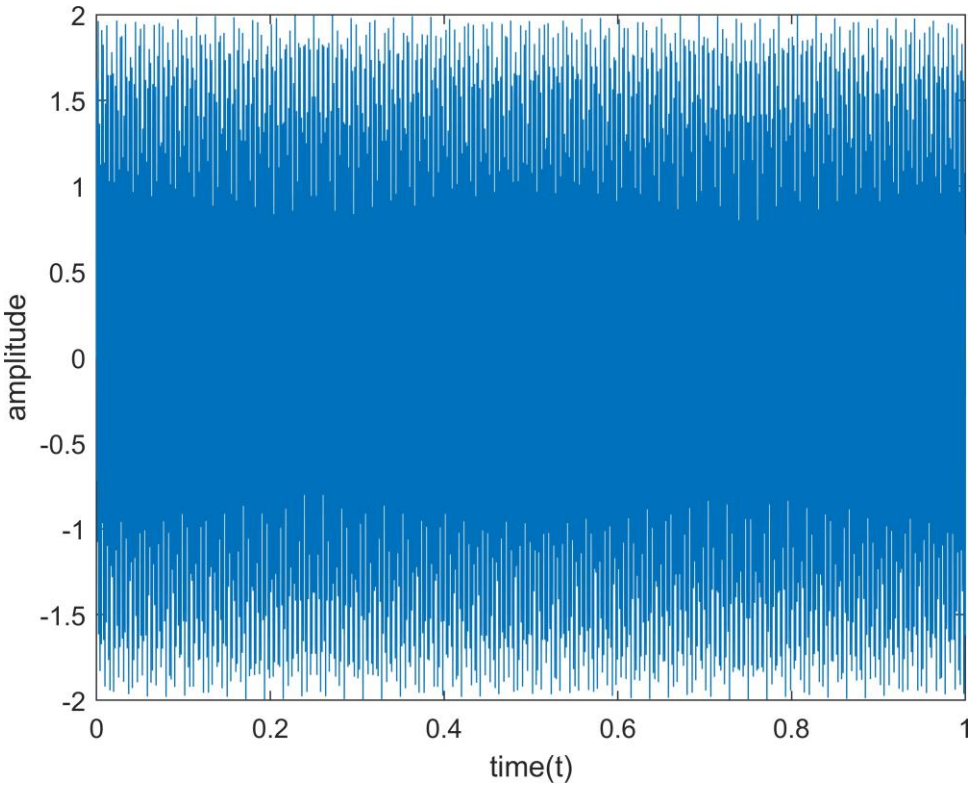

```
% METHOD 2
t=0:0.01:10;
w=2*pi*0.1;
fs=40;
b=[0 sin(w/fs) 0];
a=[1 -2*cos(w/fs) 1];
x=1.*(t==0)+0.*(t==0);
y=filter(b,a,x);
plot(y)
```


1.5
1
0.5
0
-0.5
-1
-1.5
0
200
400
600
800
1000
1200

EXPERIMENT-7 GENERATION OF DTMF SIGNALS (4B9)

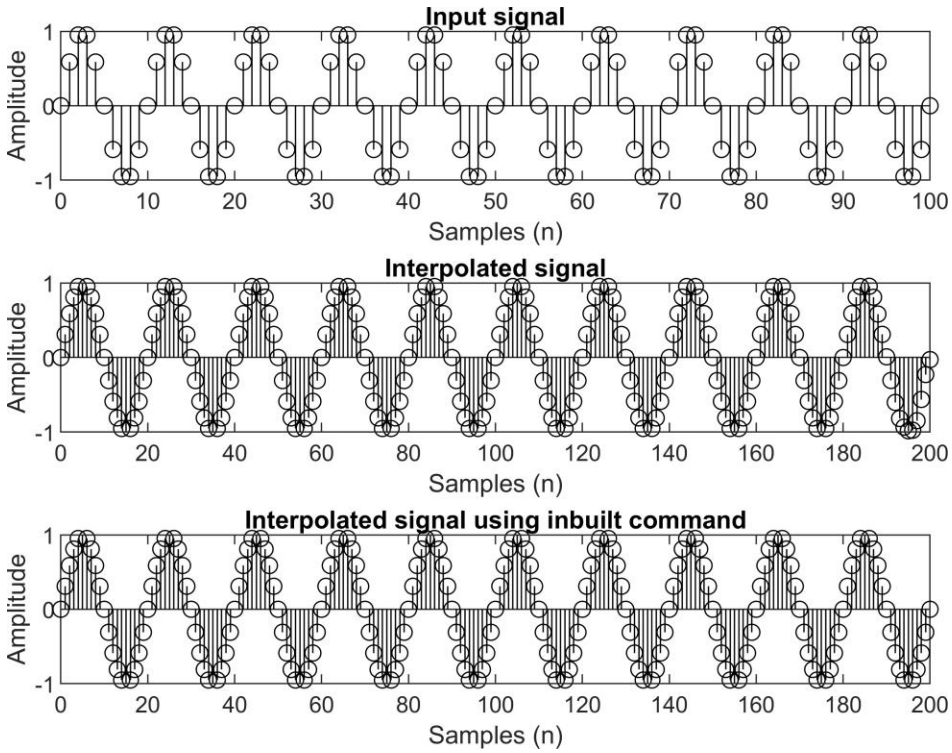
```
clear all
close all
fs=6000;
t=0:1/fs:1;
x=input('Enter the input number: ');
fr1=697;
fr2=770;
fr3=852;
fr4=941;
fa=1209;
fb=1336;
fc=1447;
fd=1633;
y11=sin(2*pi*fr4*t)+sin(2*pi*fc*t);%#
y12=sin(2*pi*fr4*t)+sin(2*pi*fa*t);%*
y10=sin(2*pi*fr4*t)+sin(2*pi*fb*t);%0
y1=sin(2*pi*fr1*t)+sin(2*pi*fa*t);%1
y2=sin(2*pi*fr1*t)+sin(2*pi*fb*t);%2
y3=sin(2*pi*fr1*t)+sin(2*pi*fc*t);%3
y4=sin(2*pi*fr2*t)+sin(2*pi*fa*t);%4
y5=sin(2*pi*fr2*t)+sin(2*pi*fb*t);%5
y6=sin(2*pi*fr2*t)+sin(2*pi*fc*t);%6
y7=sin(2*pi*fr3*t)+sin(2*pi*fa*t);%7
y8=sin(2*pi*fr3*t)+sin(2*pi*fb*t);%8
y9=sin(2*pi*fr3*t)+sin(2*pi*fc*t);%9
if(x==1)
    plot(t,y1)
    xlabel('time(t)')
    ylabel('amplitude')
elseif(x==2)
    plot(t,y2)
    xlabel('time(t)')
    ylabel('amplitude')
elseif(x==3)
    plot(t,y3)
    xlabel('time(t)')
    ylabel('amplitude')
elseif(x==4)
    plot(t,y4)
    xlabel('time(t)')
    ylabel('amplitude')
elseif(x==5)
    plot(t,y5)
    xlabel('time(t)')
    ylabel('amplitude')
elseif(x==6)
    plot(t,y6)
    xlabel('time(t)')
    ylabel('amplitude')
elseif(x==7)
    plot(t,y7)
    xlabel('time(t)')
```

```
    ylabel('amplitude')
elseif(x==8)
    plot(t,y8)
    xlabel('time(t)')
    ylabel('amplitude')
elseif(x==9)
    plot(t,y9)
    xlabel('time(t)')
    ylabel('amplitude')
elseif(x==0)
    plot(t,y10)
    xlabel('time(t)')
    ylabel('amplitude')
elseif(x==10)
    plot(t,y11)
    xlabel('time(t)')
    ylabel('amplitude')
elseif(x==11)
    plot(t,y12)
    xlabel('time(t)')
    ylabel('amplitude')
else
    disp('Enter the correct input')
end
```



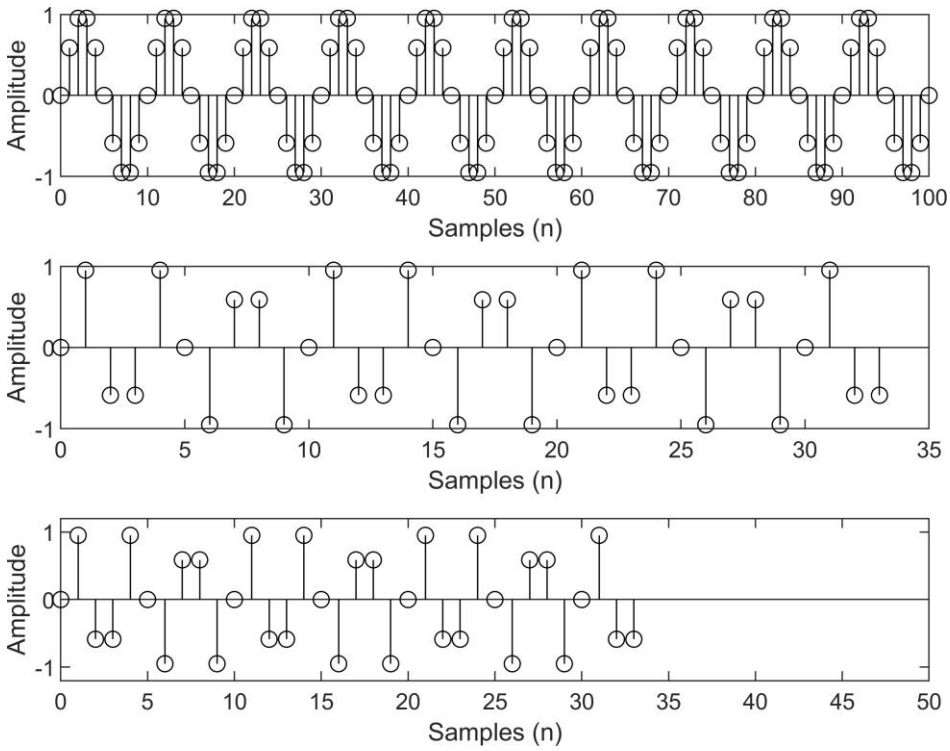
EXPERIMENT-8 IMPLEMENTATION OF DECIMATION AND INTERPOLATION PROCESSES, I/D SAMPLING RATE CONVERTERS (4B9)

```
% INTERPOLATION
clear all
close all
fs=100;
f=10;
L=input('enter interpolation factor = ');
t=0:1/fs:1;
x=sin(2*pi*f*t);
N=length(x);
n=0:N-1;
m=0:(N*L)-1;
x1=zeros(1,L*N);
j=1:L:N*L;
x1(j)=x;
f1=fir1(34,0.48,'low');
z=2*filtfilt(f1,1,x1);
y=interp(x,L);
subplot(3,1,1);
stem(n,x,'k')
title('Input signal')
xlabel('Samples (n)')
ylabel('Amplitude')
subplot(3,1,2)
stem(m,z,'k')
axis ([0 200 -1 1])
title('Interpolated signal')
xlabel('Samples (n)')
ylabel('Amplitude')
subplot(3,1,3)
stem(m,y,'k')
axis ([0 200 -1 1])
title('Interpolated signal using inbuilt command')
xlabel('Samples (n)')
ylabel('Amplitude')
```

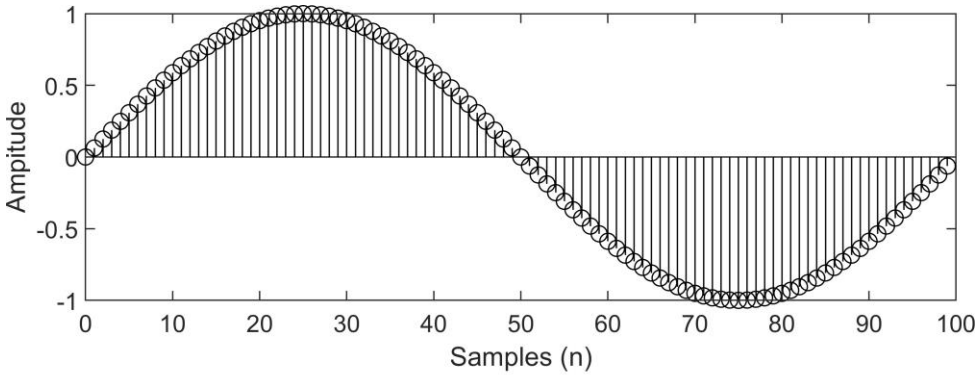


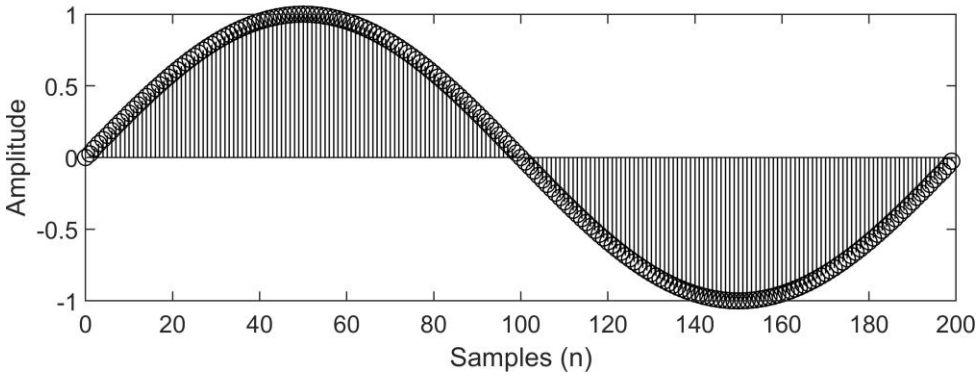

```
% DECIMATION
clear all
close all
fs=100;
fm=10;
D=input('enter decimation factor = ');
t=0:1/fs:1;
x=sin(2*pi*fm*t);
N=length(x);
n=0:N-1;
m=0:(N/D);
y=zeros(1,length(m));
j=1:D:N;
y=x(j);
v=decimate(x,D,'fir');
subplot(3,1,1)
stem(n,x,'k')
xlabel('Samples (n)')
ylabel('Amplitude')
subplot(3,1,2)
stem(m,y,'k')
xlabel('Samples (n)')
ylabel('Amplitude')
subplot(3,1,3)
stem(m,v,'k')
axis([0 50 -1.2 1.2])
xlabel('Samples (n)')
```

```
ylabel('Amplitude')
```




```
% I/D SAMPLING RATE CONVERTERS
clear all
close all
L=input('enter the upsampling factor = ');
D=input('enter the downsampling factor = ');
N=input('enter the length of the input signal = ');
f1=input('enter the frequency of first sinusoidal = ');
n=0:N-1;
x=sin(2*pi*f1*n);
y=resample(x,L,D);
subplot(2,1,1)
stem(n,x(1:N),'k')
xlabel('Samples (n)')
ylabel('Ampitude')
subplot(2,1,2)
m=0:N*L/D-1;
stem(m,y(1:N*L/D),'k')
xlabel('Samples (n)')
ylabel('Amplitude')
```


Ampitude
1
0.5
0
-0.5
-1
0
10
20
30
40
50
60
70
80
90
100
Samples (n)


Amplitude
1
0.5
0
-0.5
-1
0
20
40
60
80
100
120
140
160
180
200
Samples (n)