GURU GHASIDAS VISHWAVIDYALAYA BILASPUR (C.G)

(A Central University)

Department of Electronics & Communication Engineering

Digital Signal Processing Lab Report

VIth Semester (2021-2022)

SUBMITTED TO:-

Dr. Anil Kumar Soni

SUBMITTED BY:-

ANKIT KUMAR

Roll No-19106609

--: List Of the Experiments : --

1. Generation of digital signals and random sequences also determine their correlations.
2. To verify Linear and Circular convolutions.
3. To compute DFT of sequence and its Spectrum Analysis.
4. To implement 8-point FFT algorithm.
5. To design of FIR filters using rectangular window techniques.
6. To design of FIR filters using triangular window techniques.
7. To design of FIR filters using Kaiser Window.
8. To design of Butterworth IIR filter.
9. To design of Chebyshev IIR filter.
10. To generate the down sample (decimation) by an Integer factor,
11. To generate the up sample (interpolation) by an Integer factor

Experiment No : 1

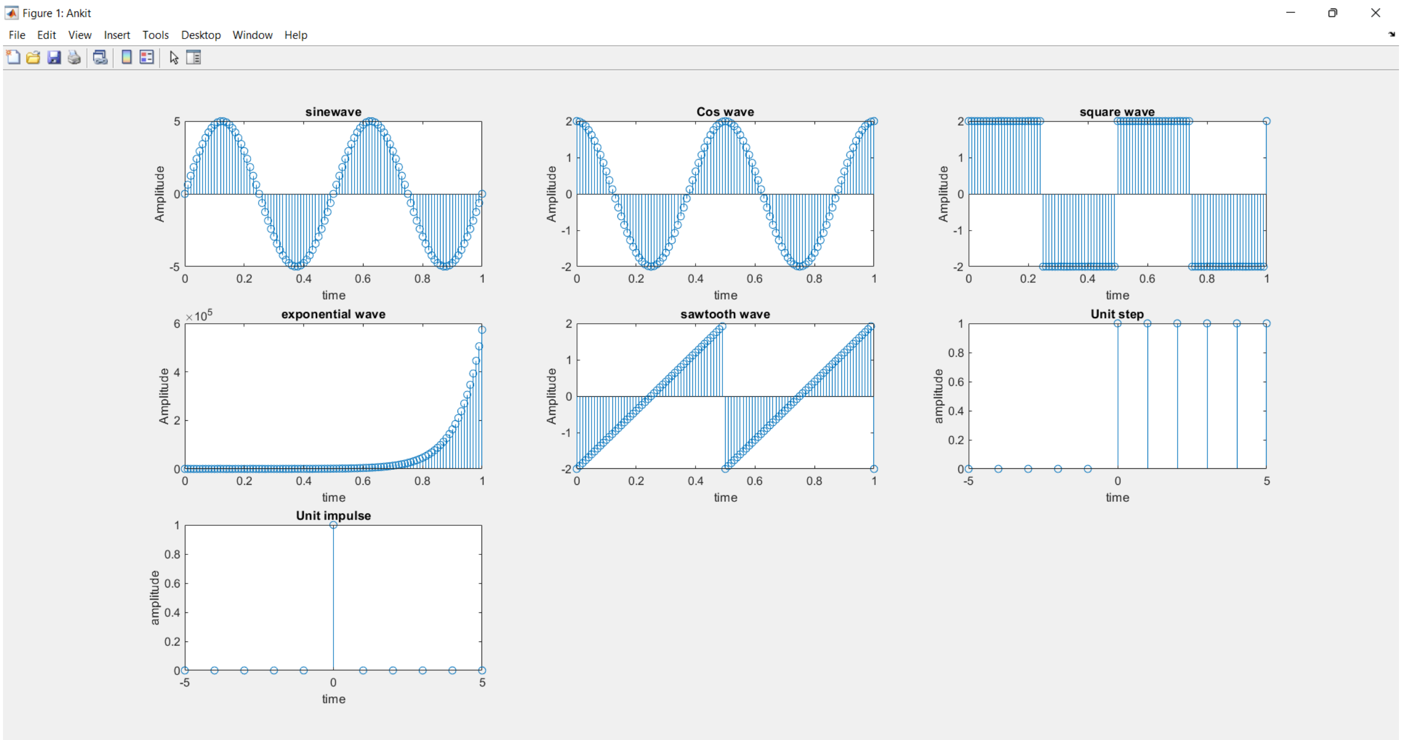
Aim:- Generation of digital signals and random sequences also determine their correlations.

MATLAB CODE:-

```
% Generate the basic signals.

clc; close all; clear all;
% plot results
figure('name','Ankit');
% sine wave
t=0:0.01:1;
% a = input('Enter the sinewave magnitude===');
a=5;
b=a*sin(2*pi*2*t);
subplot(3,3,1); stem(t,b);
xlabel('time'); ylabel('Amplitude'); title('sinewave');

% Cosine wave
t=0:0.01:1;
a=2; c=a*cos(2*pi*2*t);
subplot(3,3,2); stem(t,c);
xlabel('time'); ylabel('Amplitude'); title('Cos wave');

% Square wave
t=0:0.01:1;
a=2; b=a*square(2*pi*2*t);
subplot(3,3,3); stem(t,b);
xlabel('time'); ylabel('Amplitude'); title('square wave');

% Exponential waveform
t=0:0.01:1;
a=2;
b=a*exp(2*pi*2*t); subplot(3,3,4);
stem(t,b);
xlabel('time'); ylabel('Amplitude');
title ('exponential wave');

%sawtooth
t=0:0.01:1;
a=2; b=a*sawtooth(2*pi*2*t);
subplot(3,3,5); stem(t,b);
xlabel('time'); ylabel('Amplitude'); title ('sawtooth wave');

% unit step signal
n=-5:5;
a = [zeros(1,5),ones(1,6)]; subplot(3,3,6);
stem(n,a);
xlabel ('time');
```

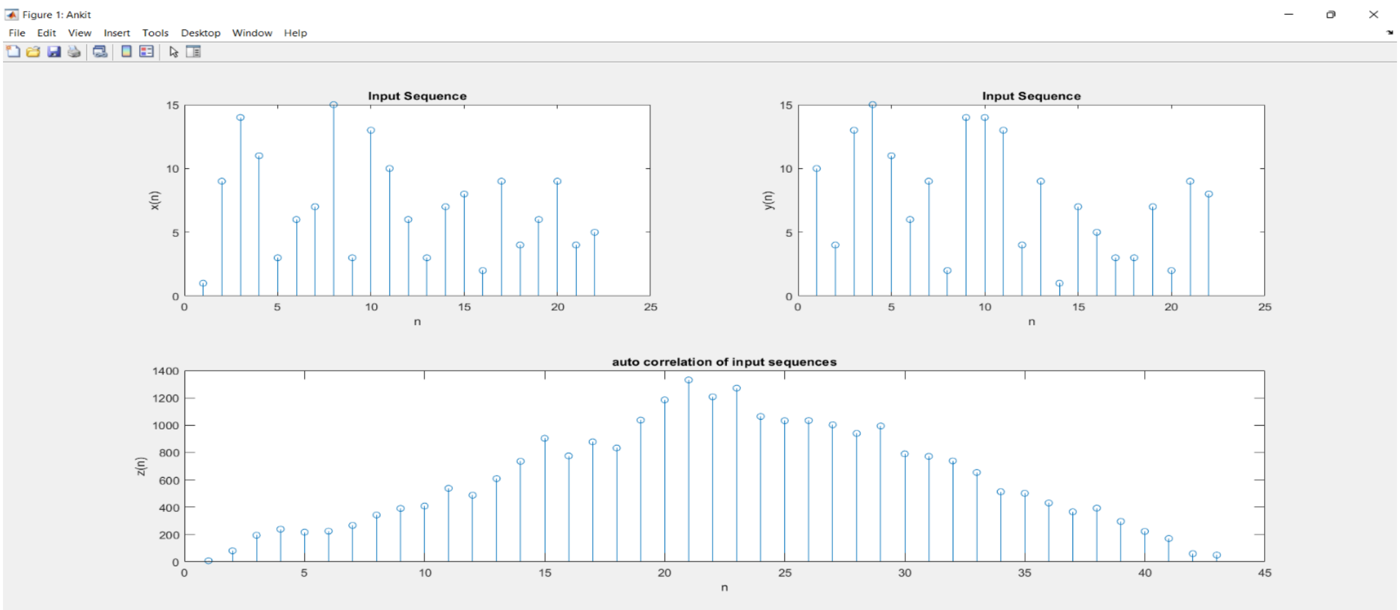
```
ylabel ('amplitude'); title('Unit step');

% unit impulse
n=-5:5;
a = [zeros(1,5),ones(1,1),zeros(1,5)]; subplot(3,3,7);
stem(n,a);
xlabel ('time');
ylabel ('amplitude');
title('Unit impulse');
```

Result:-



```
%Generation of random sequence and thier correlations.
clc;
clear all;
close all;
l=input('Enter Sequence Length')
x=randi(15,1,l);
y=randi(15,1,l);
% plot results
figure('name','Ankit');
subplot(2,2,1);
stem(x);
xlabel('n');
ylabel('x(n)');
title('Input Sequence');
subplot(2,2,2);
stem(y);
xlabel('n');
ylabel('y(n)');
title('Input Sequence');
% autocorrelation of x and y input sequences
z = xcorr(x,y);
disp('The values of z are : '); disp(z);
subplot(2,1,2);
stem(z);
xlabel('n');
ylabel('z(n)');
title('auto correlation of input sequences');
```

Result:-



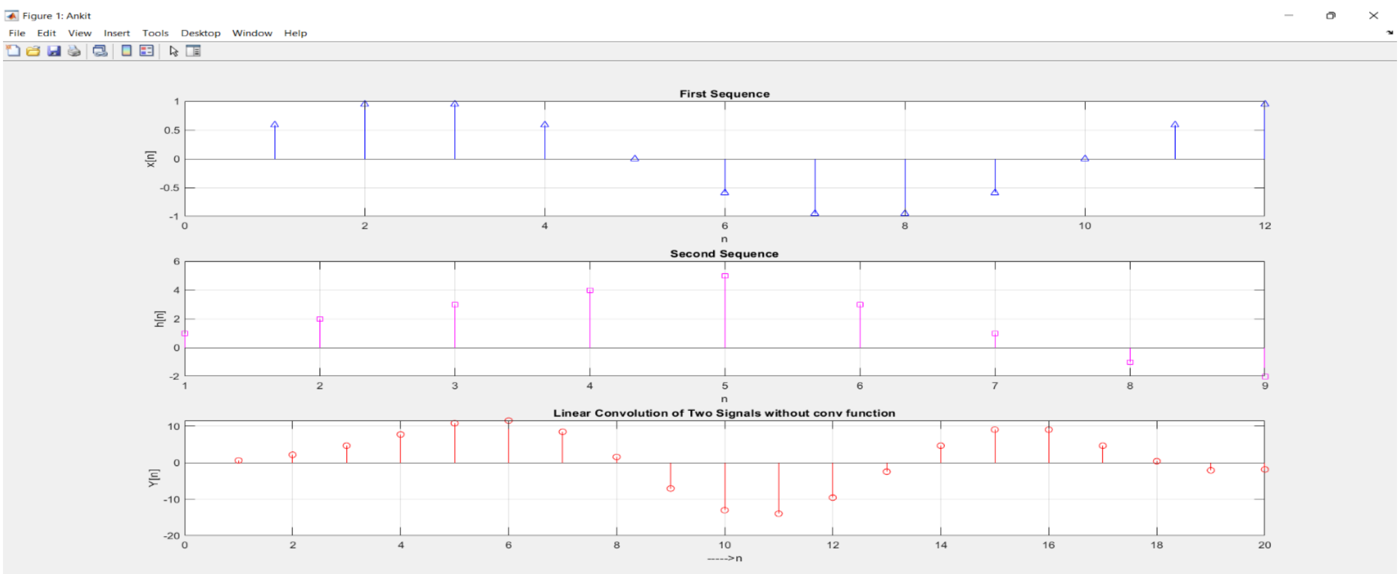
Experiment No : 2

Aim :- To verify Linear and Circular convolutions.

MATLAB CODE:-

```
% To verify Linear convolutions.
clc; clear all; close all;
clearvars
x=sin(2*pi*0.1.*(1:1:12));
h=[1 2 3 4 5 3 1 -1 -2];
% Linear 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('name','Ankit');
subplot(3,1,1); stem(x, '-b^'); xlabel('n');
ylabel('x[n]');
title('First Sequence');
grid on;
subplot(3,1,2); stem(h, '-ms');
xlabel('n'); ylabel('h[n]');
title('Second Sequence');
grid on;
subplot(3,1,3); stem(Y, '-ro');
ylabel('Y[n]'); xlabel('----->n'); grid on;
title('Linear Convolution of Two Signals without conv function');
```

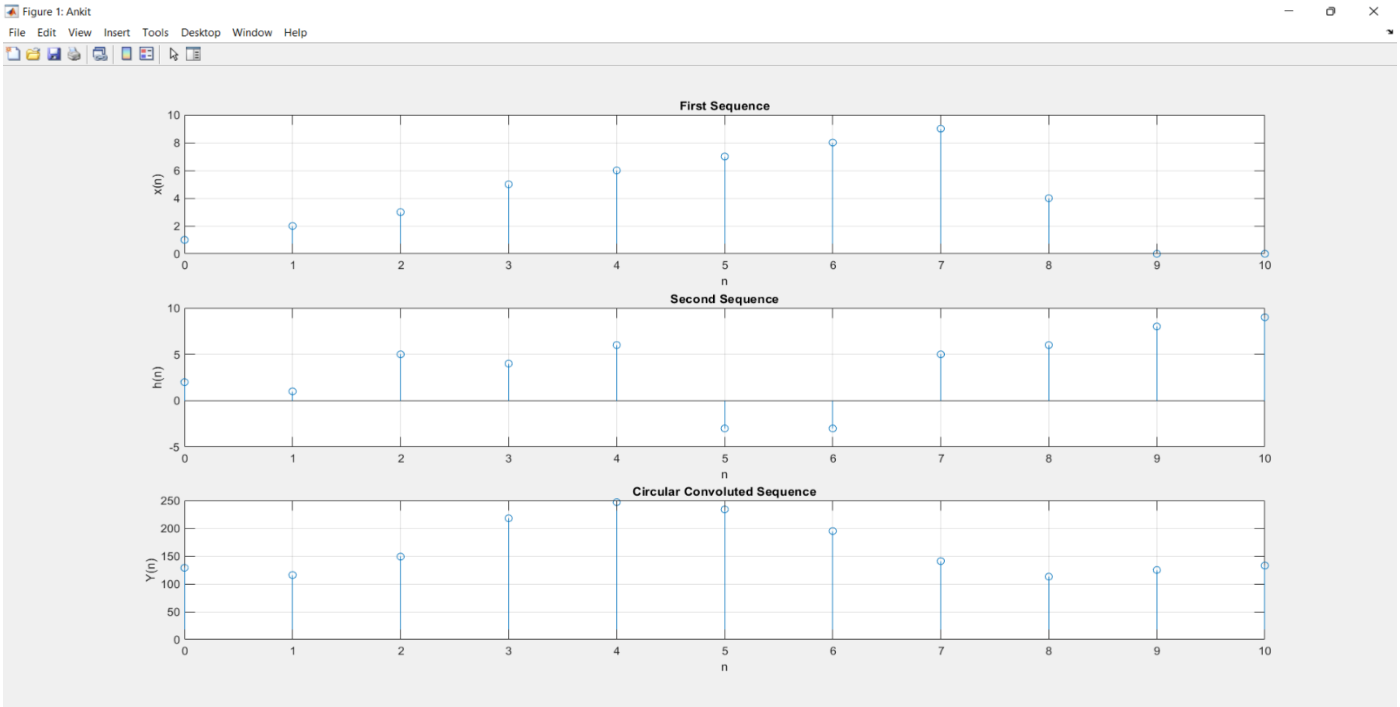
Result :-



```matlab
% To verify Circular convolutions.

clc; close all; clear all;
x=input('Enter x(n):\n');
h=input('Enter h(n):\n');
m=length(x);%length of sequence x(n)
n=length(h);%length of sequence h(n)
N=max(m,n);%length of output sequence y(n)
%For equating both sequence length
x=[x,zeros(1,N-m)];
h=[h,zeros(1,N-n)];
for n=1:N
    Y(n)=0;
    for i=1:N
        j=n-i+1;
        if(j<=0)
            j=N+j;
        end
        Y(n)=[Y(n)+x(i)*h(j)];
    end
end
n=0:N-1;%Range of all Sequences

% plot results
figure('name','Ankit');
subplot(311)
disp('First Sequence x(n) is:')
disp(x)
stem(n,x)
xlabel('n')
ylabel('x(n)')
title('First Sequence')
grid on;
subplot(312)
disp('Second Sequence h(n) is:')
disp(h)
stem(n,h)
xlabel('n')
ylabel('h(n)')
title('Second Sequence')
grid on;
subplot(313)
disp('Convoluted Sequence Y(n) is:')
disp(Y)
stem(n,Y)
xlabel('n')
ylabel('Y(n)')
title('Circular Convoluted Sequence')
```

```
grid on;
```

Result :-



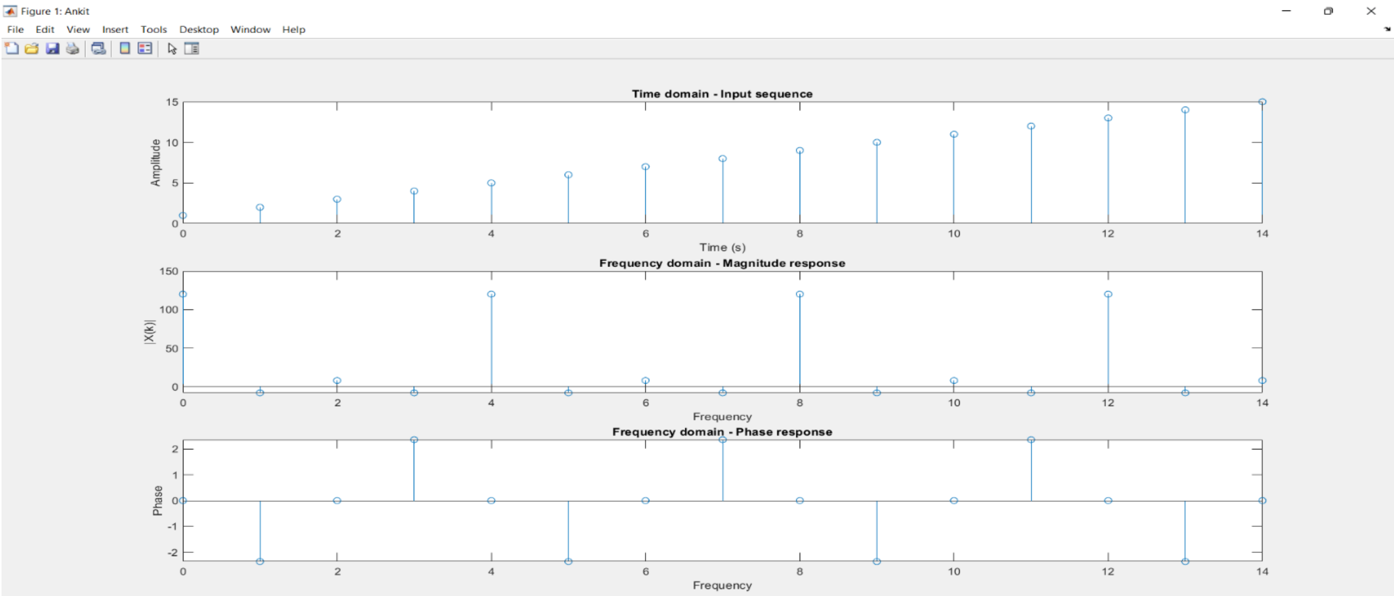
Experiment No : 3

Aim :- To compute DFT of sequence and its Spectrum Analysis.

MATLAB CODE:-

```
% To compute DFT of sequence and its Spectrum Analysis.
clc;
clear all;
close all;
x=input('enter input sequence:') % x = [2 3 -1 4];
N = length(x);
X = zeros(N,1)
for k = 0:N-1
    for n = 0:N-1
        X(k+1) = X(k+1) + x(n+1)*exp(-j*pi/2*n*k)
    end
end
t = 0:N-1
% plot results
figure('name','Ankit');
subplot(311)
stem(t,x);
xlabel('Time (s)');
ylabel('Amplitude');
title('Time domain - Input sequence')
subplot(312)
stem(t,X)
xlabel('Frequency');
ylabel('|X(k)|');
title('Frequency domain - Magnitude response')
subplot(313)
stem(t,angle(X))
xlabel('Frequency');
ylabel('Phase');
title('Frequency domain - Phase response')
X          % to check |X(k)|
angle(X)   % to check phase
```

Result:-



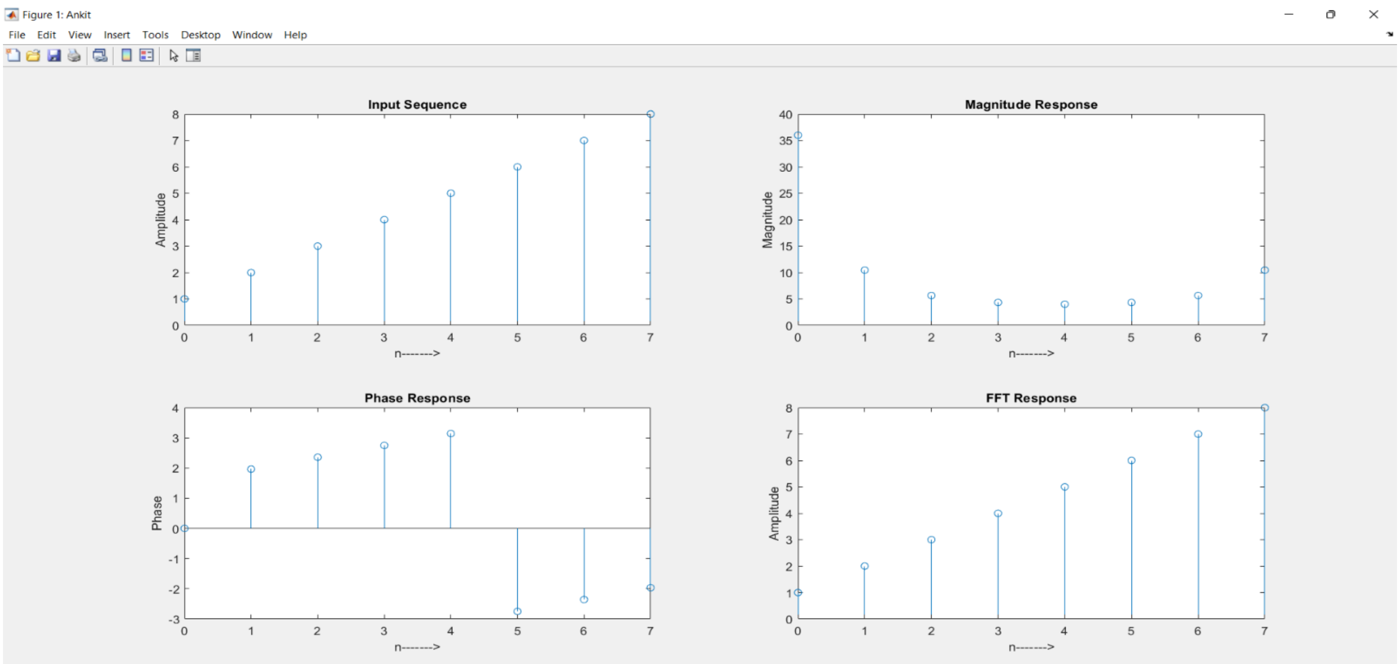
Experiment No : 4

Aim :- To implement 8-point FFT algorithm.

MATLAB CODE:-

```
% To implement 8 point FFT algorithm.
clc;
clear all;
close all;
x = input('Enter the N = 8 sequence : ');
N =length(x);
xK = fft(x,N);
xn = ifft(xK);
n=0:N-1;
% plot results
figure('name','Ankit');
subplot (2,2,1);
stem(n,x);
xlabel('n-------->');
ylabel('Amplitude');
title('Input Sequence');
subplot (2,2,2);
stem(n,abs(xK));
xlabel('n-------->');
ylabel('Magnitude');
title('Magnitude Response');
subplot (2,2,3);
stem(n,angle(xK));
xlabel('n-------->');
ylabel('Phase');
title('Phase Response');
subplot (2,2,4);
stem(n,xn);
xlabel('n-------->');
ylabel('Amplitude');
title('FFT Response');
```

Result :-



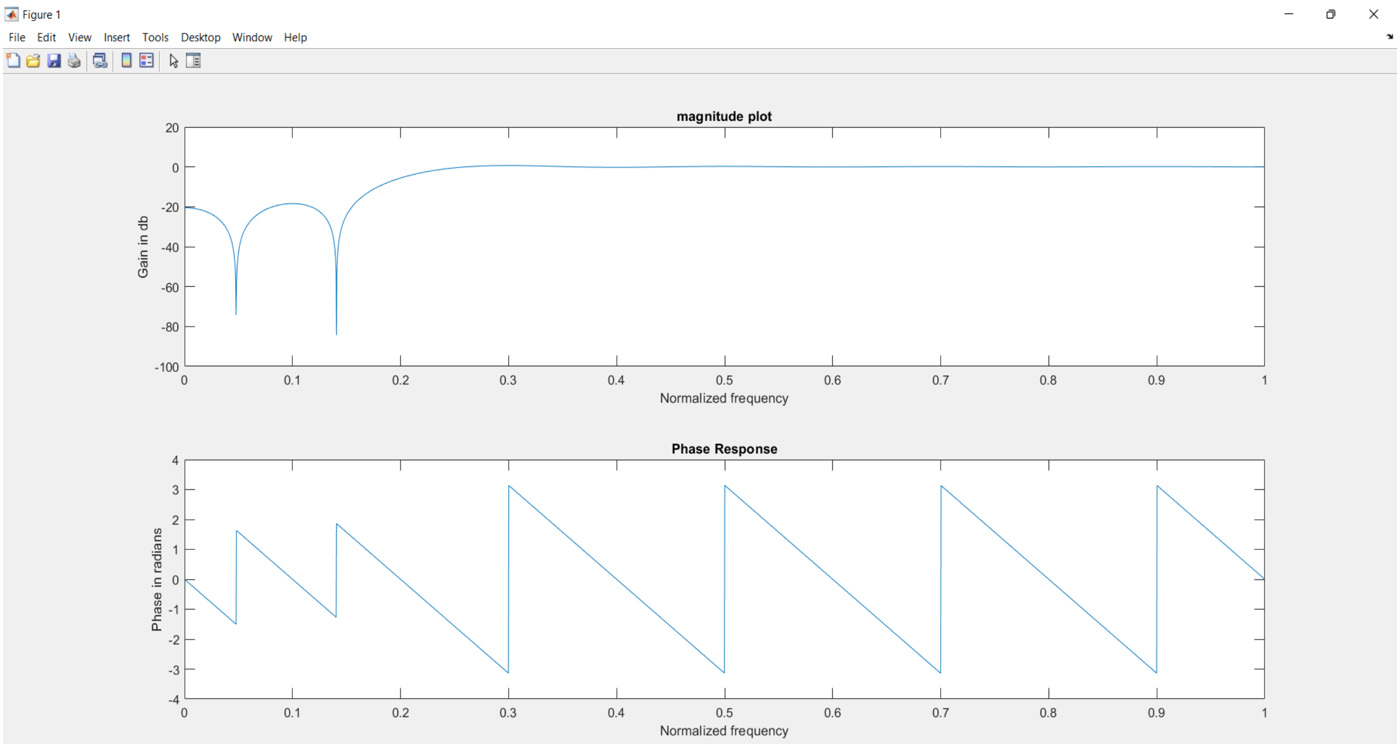
Experiment No : 5

Aim :- To design of FIR filters using rectangular window techniques.

MATLAB CODE:-

```
% Response of high-pass FIR filter using Rectangular window
clc;
clear all;
close all;
n=20;
fp=100;
fq=300;
fs=1000;
fn=2*fp/fs;
window=rectwin(n+1);
b=fir1(n, fn,'high',window);
w=0:0.001:pi;
[h,om]=freqz(b,1,w);
a=20*log10(abs(h));
b=angle(h);
% plot results
figure('name','Ankit');
subplot(2,1,1);plot(w/pi,a);
xlabel('Normalized frequency')
ylabel('Gain in db')
title('magnitude plot')
subplot(2,1,2); plot(w/pi,b);
xlabel('Normalized frequency')
ylabel('Phase in radians')
title('Phase Response')
```

Result:-



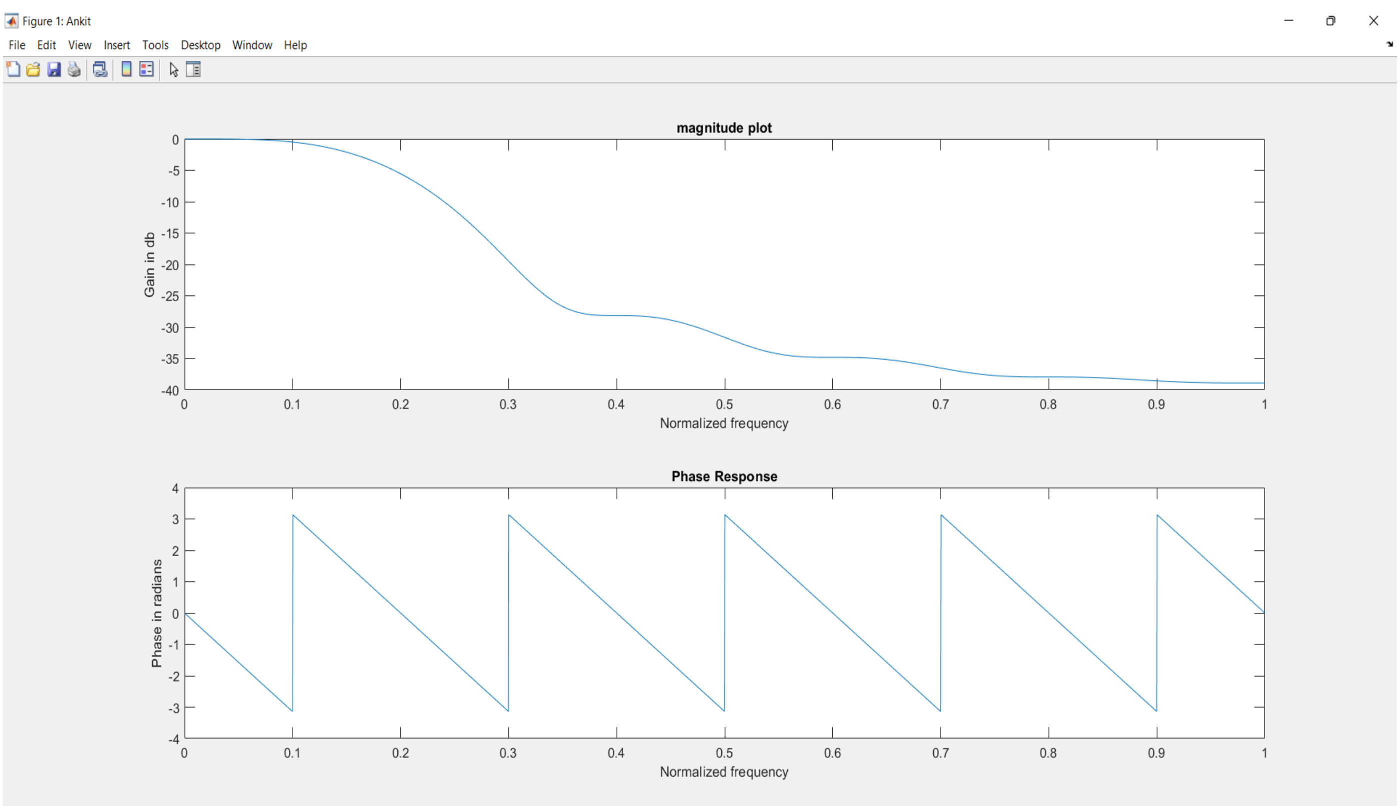
Experiment No : 6

Aim :- To design of FIR filters using triangular window techniques.

MATLAB CODE:-

```
% Response of low-pass FIR filter using Bartlett(Triangular) window
clc;
clear all;
close all;
n=20; fp=100;
fq=300; fs=1000;
fn=2*fp/fs; window=bartlett(n+1);
b=fir1(n,fn,window);
w=0:0.001:pi;
[h,orn]=freqz(b,1,w);
a=20*log10(abs(h));
b=angle(h);
% plot results
figure('name','Ankit');
subplot(2,1,1);plot(w/pi,a);
xlabel('Normalized frequency')
ylabel('Gain in db')
title('magnitude plot')
subplot(2,1,2);plot(w/pi,b);
xlabel('Normalized frequency')
ylabel('Phase in radians')
title('Phase Response')
```

Result :-



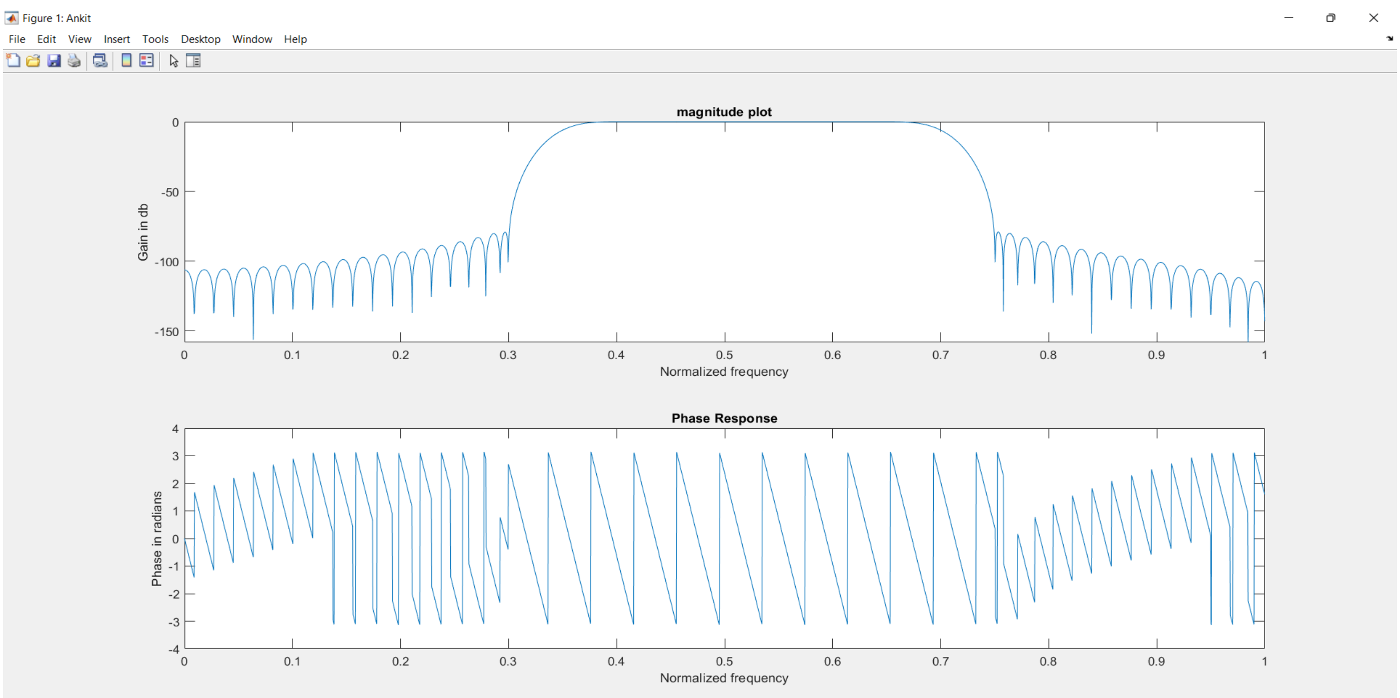
Experiment No : 7

Aim :- To design of FIR filters using Kaiser Window.

MATLAB CODE:-

```
% Response of band pass FIR filter using Kaiser window
clc;
clear all;
close all;
fs = 20000;             % sampling rate
F = [3000 4000 6000 8000]; % band limits
A = [0 1 0];            % band type: 0='stop', 1='pass'
dev = [0.0001 10^(0.1/20)-1 0.0001]; % ripple/attenuation specifications
[M,Wn, beta,typ] = kaiserord(F,A,dev,fs); % window parameters
b = fir1(M,Wn, typ, kaiser(M+1, beta),'noscale'); % filter design
w=0:0.001:pi;
[h,om]=freqz(b,1,w);
a=20*log10(abs(h));
b=angle(h);
% plot results
figure('name','Ankit');
subplot(2,1,1),plot(w/pi,a);
xlabel('Normalized frequency')
ylabel('Gain in db')
title('magnitude plot')
subplot(2,1,2),plot(w/pi,b);
xlabel('Normalized frequency')
ylabel('Phase in radians')
title('Phase Response')
```

Result :-



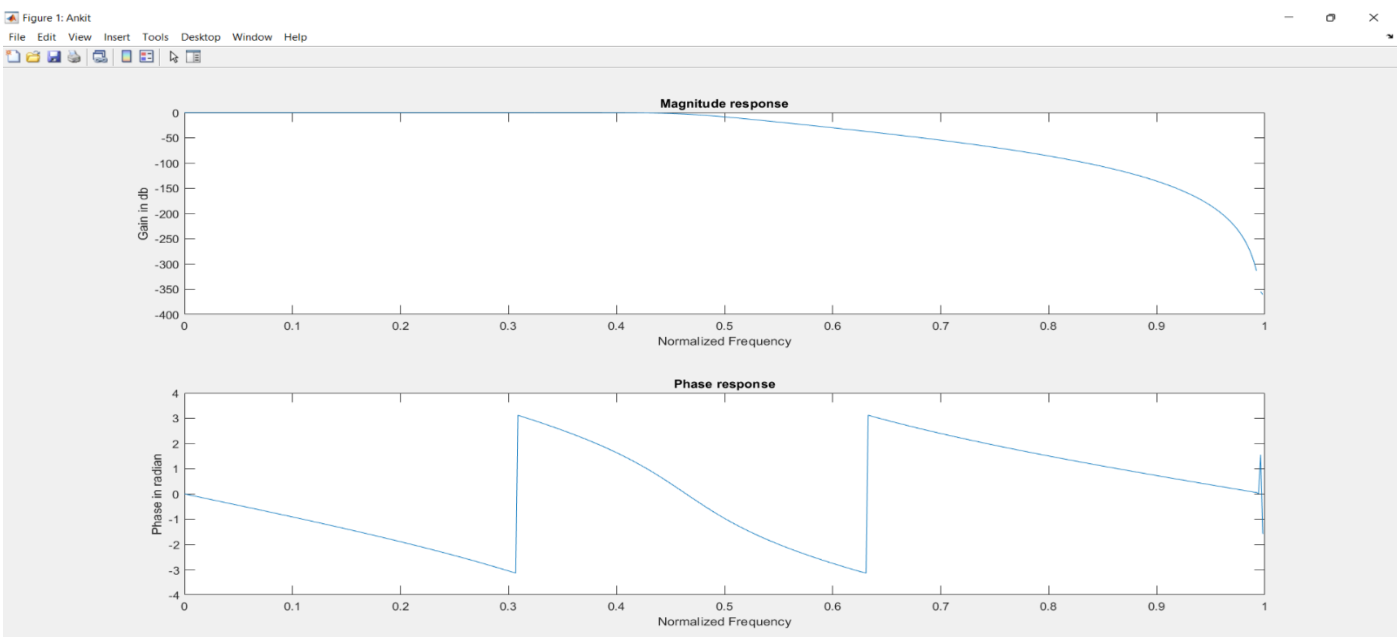
Experiment No : 8

Aim :- To design of Butterworth IIR filter.

MATLAB CODE:-

```
%Butterworth low-pass filter programm
clc;
clear all;
close all;
alphas =30;
alphap = 0.5;
fpass = 1000;
fstop = 1500;
fsam = 5000;
wp=2*fpass/fsam;
ws=2*fstop/fsam;
[n,wn] = buttord(wp,ws,alphap,alphas);
[b,a] = butter(n,wn);
[h,w] = freqz(b,a);
% plot results
figure('name','Ankit');
subplot(2,1,1);
plot(w/pi,20*log10(abs(h)));
xlabel('Normalized Frequency');
ylabel('Gain in db');
title('Magnitude response');
subplot(2,1,2);
plot(w/pi,angle(h));
xlabel('Normalized Frequency');
ylabel('Phase in radian');
title('Phase response');
```

Result :-



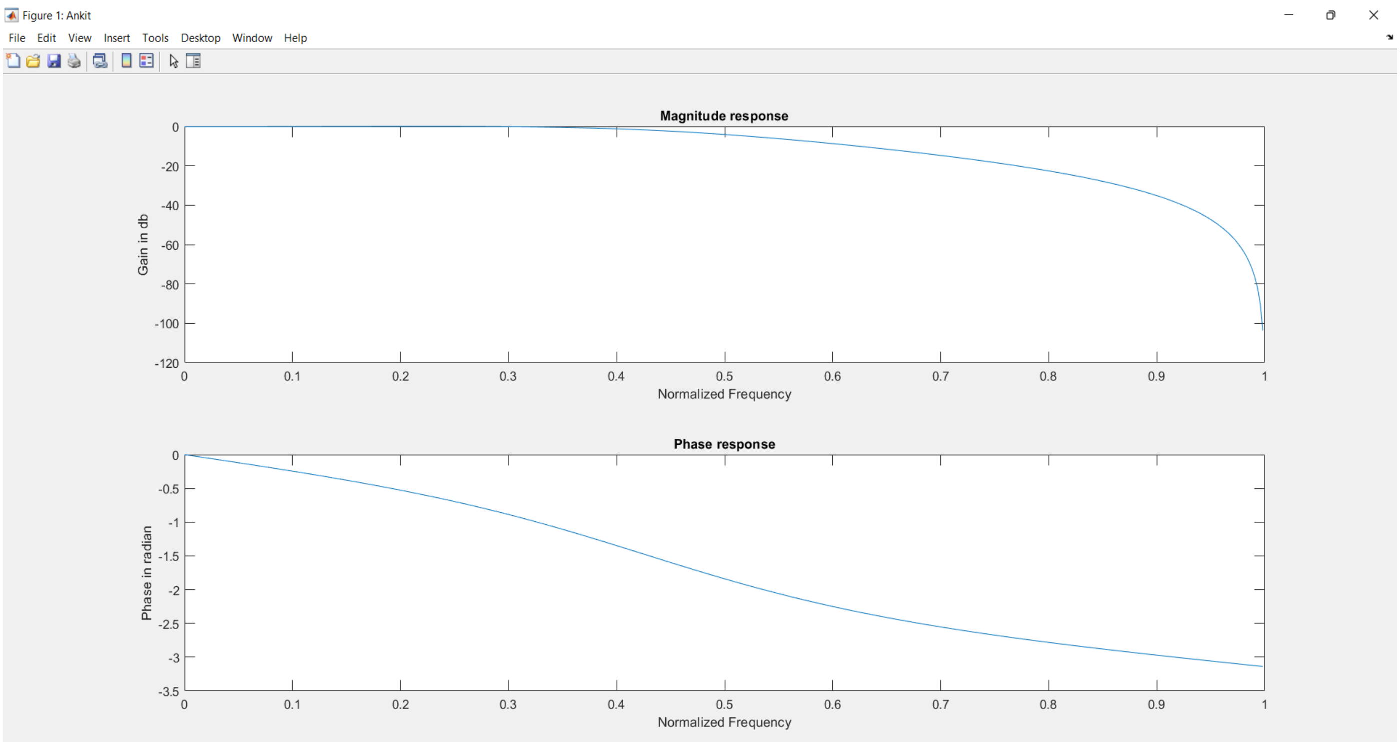
Experiment No : 9

Aim :- To design of Chebyshev IIR filter.

MATLAB CODE:-

```
% Chebyshev low-pass IIR filter programm
clc;
clear all;
close all;
alphap = 0.15;
alphas =0.9;
wp= 0.3*pi;
ws= 0.5*pi;
[n,wn] = cheb1ord(wp/pi,ws/pi,alphap,alphas);
[b,a] = cheby1(n,alphap,wn);
[h,w] = freqz(b,a);
% plot results
figure('name','Ankit');
subplot(2,1,1);
plot(w/pi,20*log10(abs(h)));
xlabel('Normalized Frequency');
ylabel('Gain in db');
title('Magnitude response');
subplot(2,1,2);
plot(w/pi,angle(h));
xlabel('Normalized Frequency');
ylabel('Phase in radian');
title('Phase response');
```

Result :-



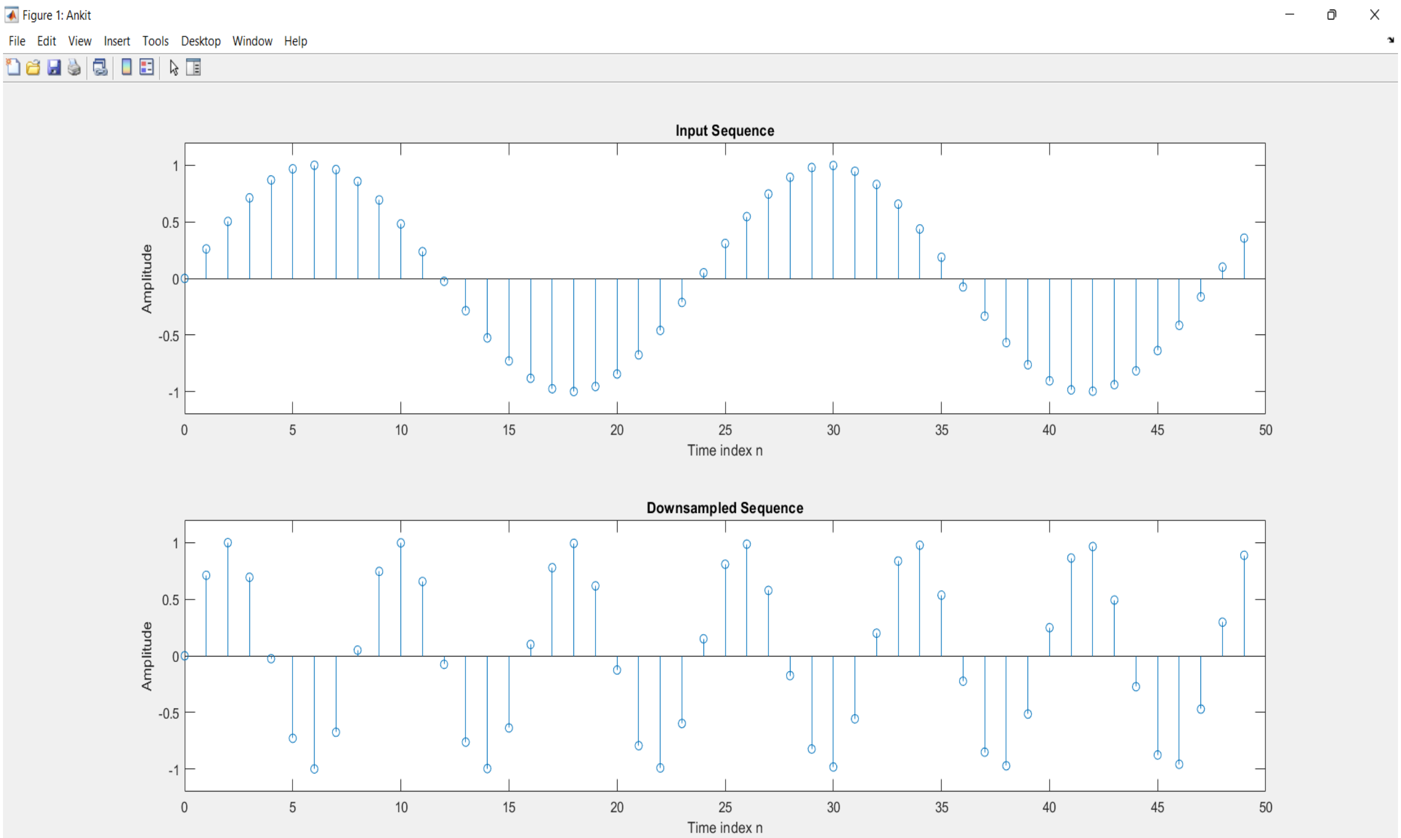
Experiment No : 10

Aim :- To generate the down sample (decimation) by an Integer factor.

MATLAB CODE:-

```
% Down Sampling by an integer factor
clc;
clear all;
close all;
n=0:49;
m= 0:50*3-1;
x = sin(2*pi*0.042*m);
y = x([1:3:length(x)]);
% plot results
figure('name','Ankit');
subplot(2,1,1),stem(n, x(1:50));
axis([0 50 -1.2 1.2]);
xlabel('Time index n');
ylabel('Amplitude');
title('Input Sequence');
subplot(2,1,2),stem(n, y);
axis([0 50 -1.2 1.2]);
xlabel('Time index n');
ylabel('Amplitude');
title('Downsampled Sequence');
```

Result :-



Experiment No : 11

Aim :- To generate the up sample (interpolation) by an Integer factor.

MATLAB CODE:-

```
% Up-Sampling by an integer factor
clc;
clear all;
close all;
n=0:50;
x = sin(2*pi*0.06*n);
y = zeros([1, 2*length(x)]);
y([1:2:length(y)]) = x;

% plot results
figure('name','Ankit');
subplot(2,1,1),stem(n, x);
xlabel('Time index n');
ylabel('Amplitude');
title('Input Sequence');
subplot(2,1,2),stem(n, y(1:length(x)));
xlabel('Time index n');
ylabel('Amplitude');
title('Up-sampled Sequence');
```

Result :-