1 "Under,Niquist,over Sampling"

```
clc;clear all;close all;
tfinal=0.05;
t=0:0.00005:tfinal;
fd=input('Enter analog frequency: ');
xt=cos(2*pi*fd*t);
fs1=1.3*fd;
n1=0:1/fs1:tfinal;
xn1=cos(2*pi*fd*n1);
n1i=0:1/(2*fs1):tfinal;
vq1=interp1(n1,xn1,n1i,'linear');
subplot(3,1,1);
plot(t,xt,'b',n1i,vq1,'r*-');
xlabel('Time
(s)');ylabel('Amplitude');title('UndersamplingPlot');
legend('Analog','Discrete');
fs2=2*fd;
n2=0:1/fs2:tfinal;
xn2=cos(2*pi*fd*n2);
n2i=0:1/(2*fs2):tfinal;
vq2=interp1(n2,xn2,n2i,'linear');
subplot(3,1,2);
plot(t,xt,'b',n2i,vq2,'r*-');
xlabel('Time
(s)');ylabel('Amplitude');title('NyquistPlot');
legend('Analog','Discrete');
fs3=4*fd;
n3=0:1/fs3:tfinal;
xn3=cos(2*pi*fd*n3);
n3i=0:1/(2*fs3):tfinal;
vq3=interp1(n3,xn3,n3i,'linear');
subplot(3,1,3);
plot(t,xt,'b',n3i,vq3,'r*-');
xlabel('Time (s)');ylabel('Amplitude');
title('Oversampling Plot');
legend('Analog','Discrete');
```

Output:Enter analog frequency 200

2A"Linear Convolution"

```
clc;clear all;close all;
n1=input('Enter start point of first sequence: ');
n2=input('Enter end point of first sequence: ');
m1=input('Enter start point of second sequence: ');
m2=input('Enter end point of second sequence: ');
n=n1:n2;m=m1:m2;l=n1+m1:n2+m2;
x=input('Enter the first sequence: ');
h=input('Enter the second sequence: ');
y=conv(x,h);
disp('Linearly Convolved Sequence is');disp(y);
figure;subplot(3,1,1);stem(n,x);
xlabel('---> n');ylabel('---> x(n)');
title('First Sequence');grid on;subplot(3,1,2);
stem(m,h);
xlabel('---> m');ylabel('---> h(n)');
title('Second Sequence');grid on;
subplot(3,1,3);stem(l,y);
xlabel('---> n');ylabel('---> y(n)');
title('Linearly Convolved Sequence');grid on;
```

```
Output:Enter st point of first seq 0
Enter end point of first seq 3
Enter st point of first seq 0
Enter end point of first seq 3
Enter the first sequence [1111]
Enter the second sequence [1 2 3 4]
Linearly Convolved Sequence is
13 6 10 974
```

2B"Circular Convolution"

```
lc;clear all;close all;
x=input('Enter the first sequence: ');
h=input('Enter the second sequence: ');
N1=length(x);N2=length(h);
N=max(N1,N2);
x=[x zeros(1,N-N1)];
h=[h zeros(1,N-N2)];y=zeros(1,N);
for n=0:N-1
for i=0:N-1
j=mod(n-i,N);
y(n+1)=y(n+1)+x(i+1)*h(j+1);
end end
disp('Output sequence of circular convolution:');
disp(y);
n=0:length(y)-1;stem(n,y);
xlabel('Time index');ylabel('Amplitude');
title('Circular Convolution Sequence of x and h');
```

Output:enter the first sequence [1234]
enter the second sequence [567]
output sequence of circular convolution 50 44 34 52

3"C D A Propety"

```
clc;clear all;close all;
x=input('Enter the input sequence x: ');
h=input('Enter the impulse response sequence h:');
y1=conv(x,h);y2=conv(h,x);
disp('y1=x*h is:');disp(y1);
disp('y2=h*x is:');disp(y2);
if isequal(y1,y2)
disp('Commutative property is satisfied');
else
disp('Commutative property is not satisfied');
end
h1=input('Enter impulse response sequence of system h1: ');
h2=input('Enter impulse response sequence of system h2: ');
N1=length(h1);N2=length(h2);
N=max(N1,N2);h1=[h1 zeros(1,N-N1)];
h2=[h2 zeros(1,N-N2)];
y1=conv(x,h1);y2=conv(x,h2);
y3=conv(x,(h1+h2));
disp('x*h1 is:');disp(y1);
disp('x*h2 is:');disp(y2);
disp('x*(h1+h2) is:');disp(y3);
if isequal(y3,y1+y2)
disp('Distributive property is verified');else
disp('Distributive property is not verified');end
q1=conv(h1,h2);
q2=conv(x,q1);
disp('x*(h1*h2) is:');disp(q2);
p1=conv(x,h1);
p2=conv(p1,h2);
disp('(x*h1)*h2 is:');
disp(p2);
if isequal(q2,p2)
disp('Associative property is verified');
else
disp('Associative property is not verified');end
```

Result:

enter the input sequence x [34-1-2]

enter impulse response sequence h [345]

4A"Auto corelation"

```
clc;clear all;close all;

x=input('Enter the sequence x(n): ');

[r,lag]=xcorr(x,x);

disp('Auto Correlation Sequence is');disp(r);

stem(lag,r);

xlabel('--- Lag index');ylabel('--- Amplitude');

title('Auto Correlated Sequence');

energy=sum(x.^2);

centre_index=ceil(length(r)/2);

if r(centre_index)==energy

   disp('Mid value of correlation gives energy');

else disp('Mid value of correlation does not give
energy');end if isequal(r,fliplr(r))

 disp('Autocorrelation is even --- Proved');else

disp('Autocorrelation is even --- Not Proved');end
```

4B"Cross corelation"

```
clc;

clear all;

close all;

x=input('Enter the first sequence');

y=input('Enter the second sequence');

[r,lag]=xcorr(x,y);

disp('Cross Correlation Sequence is');

disp(r);

stem(lag,r);

xlabel('--- Lag index');

ylabel('--- Amplitude');

title('Cross Correlated Sequence');
```

5"Difference Equation"

```
clc;clear all;close all;

N=input('Length of response required=');

b=input('Enter the numerator coefficient');

a=input('Enter the denominator coefficient');

x=[ones(1,N)];n=0:1:N-1;y=filter(b,a,x);

figure;subplot(2,1,1);stem(n,x);

xlabel('--->n');ylabel('--->u(n)');

title('Step Input');

subplot(2,1,2);stem(n,y);

xlabel('--->n');ylabel('--->y(n)');

title('Step Response');

x=cos(0.05*pi*n);

y=filter(b,a,x);

figure;

subplot(2,1,1);

stem(n,x);

xlabel('--->n');

ylabel(' --->x(n)');

title(' Steady State Input');

subplot(2,1,2);

stem(n,y);

xlabel('--->n');

ylabel('--->y(n))');

title(' Steady State Response');
```

Output

```
100      [1]    [1 -0.5]
```

6"N-Point DFT"

```
clc;clear all;close all;

x=input('Enter the sequence x(n)')xlen=length(x);

N=input('Enter the points of DFT');

for k=1:N y(k)=0;

for n=1:xlen

y(k)=y(k)+x(n)*exp(-2*pi*1j*(k-1)*(n-1)/N); end
end

disp('DFT of x using DFT equation:');disp(y);

pm=sqrt(real(y).^2+imag(y).^2);

yl=real(y);z1=imag(y);for k=1:N

if ((yl(k)>=0)&& (z1(k)>=0))p(k)=atan(z1(k)/yl(k));

end if((yl(k)<0)&& (z1(k)>=0))

p(k)=pi+atan(z1(k)/yl(k));end

if ((yl(k)<0)&& (z1(k)<0))p(k)=-pi+atan(z1(k)/yl(k));

end if ((yl(k)>=0)&& (z1(k)<0))

p(k)=atan(z1(k)/yl(k));end end figure;subplot(311);

stem(x);xlabel('--->n');ylabel('--->x(n)');

title('Input Data Sequence x(n)');subplot(312);

stem(pm);xlabel('--->n');ylabel('--->FFTmagnitude');

title('Magnitude Plot');subplot(313);stem(p);

xlabel('--->n');ylabel('--->FFT phase');

title('Phase plot');f=fft(x,N);

disp('The DFT of x using built-in routine:');disp(f);

mag=abs(f);phase=angle(f);figure;subplot(311);

stem(x);xlabel('--->n');ylabel('--->x(n)');

title('Input Data Sequence x(n)');subplot(312);

stem(mag);xlabel('--->n');

ylabel('-->FFT magnitude');

title('Magnitude Plot');subplot(313);

stem(phase);xlabel('--->n');

ylabel('-->FFT phase');title('Phase plot');
```

7"Linearity and Parsevals"

```
clc;clear all;close all;

N=input('How many point DFT do you want?');

x1=input('Enter the first sequence=');

x2=input('Enter the second sequence=');

a=input('Enter the first constant a=');

b=input('Enter the second constant b=');

n1=length(x1);n2=length(x2);

x1=[x1 zeros(1,N-n1)];x2=[x2 zeros(1,N-n2)];

x3=a*x1+b*x2;

rl=fft(x1,N);r2=fft(x2,N);r3=fft(x3,N);

disp('a FFT(x1)');disp(a*rl);disp('b FFT(x2)');

disp(b*r2);disp('FFT(a*x1+b*x2)');disp(r3);

if(r3==(a*rl+b*r2))

disp('Linearity property is satisfied');else

disp('Linearity property is not satisfied');end

N=input('How many point DFT do you want?');

x=input('Enter the sequence=');fl=max(xcorr(x));

disp('calculation of energy by time domain
method');disp(fl);

xl=fft(x,N);x2=abs(xl).^2;x3=sum(x2)/N;

disp('calculation of energy by frequency domain
method');disp(x3);if x3==fix(fl)

fprintf('Parseval''s Theorem is verified\n');else

fprintf('Parseval''s Theorem is not verified\n');end
```

Output:How many point DFT do you want? 4

Enter the first sequence= [[2, 1, 1, 3]]

Enter the second sequence ze = [[1, 1, 3, 4]]

Enter the first constant a = 3

Enter the second constant b=-1

8"Square pulse "

```
clc;clear all;close all;Fs=150;t=-0.5:1/Fs:0.5;w=.2;
x=rectpuls(t,w);subplot(2,1,1);plot(t,x);
title('Square pulse');xlabel('time(ms)');
ylabel('Amplitude');y=fftshift(fft(x));N=length(y);
n=-(N-1)/2:(N-1)/2;f=abs(y);
subplot(212);plot(n,f);title('Magnitude of FFT');
xlabel('Frequency in Hz');
ylabel('Amplitude');
```

"Sine function"

```
clc;clear all;close all;Fs=10;
t=-5:1/Fs:5;x=sinc(t)subplot(2,1,1);
plot(t,x);title('sinc pulse');xlabel('time(ms)');
ylabel('Amplitude');y=fftshift(fft(x));
N=length(y);n=-(N-1)/2:(N-1)/2;f=abs(y);
subplot(212);plot(n,f);title('Magnitude of FFT');
xlabel('Frequency in Hz');ylabel('Amplitude');
```

9/10/11

```
clc;clear all;close all;
Wc=input('Enter the cut-off between 0 and 1:');
n=input('Enter the order of the filter:');
if(rem(n, 2)~=0)
   n=n+1;
end
n1=n+1;
yl=BOXCAR/HAMMING/BARTLETT(n1);
b=fir1(n,Wc,yl);
[h,o]=freqz(b,1,256);
m=20*log10(abs(h));
subplot(321);
plot(o/pi,m);
xlabel('---> Normalised Frequency');
ylabel('---> Gain in db');
title('Lowpass filter using ______ window');
grid on;
subplot(323);
plot(psd(spectrum.periodogram,y,'Fs',Fs,'NFFT',length(y)));
title('spectrum of signal before filtering');
y2=filter(b,1,y);
subplot(325);
plot(psd(spectrum.periodogram,y2,'Fs',Fs,'NFFT',length(y)));
title('spectrum of signal after filtering');
b=fir1(n,Wc,'high',yl);
[ho]=freqz(b,1,256);
m=20*log10(abs(ho));
subplot(322);
plot(o/pi,m);
xlabel('---> Normalised Frequency');
ylabel('---> Gain in db');
title('Highpass filter using _____r window');
grid on;
subplot(324);
plot(psd(spectrum.periodogram,y,'Fs',Fs,'NFFT',length(y)));
title('spectrum of signal before filtering');
y2=filter(b,1,y);
subplot(326);
plot(psd(spectrum.periodogram,y2,'Fs',Fs,'NFFT',length(y)));
title('spectrum of signal after filtering');
```

Output

```
Enter the cut-off between 0 and 1:.48
Enter the order of the filter: 34
```

12”IIR Filter”

```
clc;
clear all;
close all;
pb=input('Enter the Pass Band Edge Frequency ');
sb=input('Enter the stop Band Edge Frequency ');
pbr=input('Enter the Pass Band attenuation ');
sbr=input('Enter the Stop Band attenuation ');
fs=input('Enter the Sampling Frequency ');
pbrad=2*fs*tan((pb*2*pi)/(2*fs));
sbrad=2*fs*tan((sb*2*pi)/(2*fs));
[n,wn]=buttord(pbrad,sbrad,pbr,sbr);
[b,a]=butter(n,wn,'s');
[z,p,k]=tf2zp(b,a);
[zd,pd,kd]=bilinear(z,p,k,fs);
[num,den]=zp2tf(zd,pd,kd);
figure;
freqz(num,den,512,fs);
title('BUTTERWORTH FREQUENCY RESPONSE');
load chirp
figure;
subplot(211);
plot(psd(spectrum.periodogram,y,'Fs',Fs,'NFFT',len
gth(y)));
title('spectrum of signal before filtering');
y2=filter(num,den,y);
subplot(212);
plot(psd(spectrum.periodogram,y2,'Fs',Fs,'NFFT',le
ngth(y)));
title('spectrum of signal after low pass filtering');
[b,a]=butter(n,wn,'high','s');
[z,p,k]=tf2zp(b,a);
[zd,pd,kd]=bilinear(z,p,k,fs);
[num,den]=zp2tf(zd,pd,kd);
figure;
freqz(num,den,512,fs);
title('BUTTERWORTH FREQUENCY RESPONSE');
load chirp
figure;
subplot(211);
plot(psd(spectrum.periodogram,y,'Fs',Fs,'NFFT',len
gth(y)));
title('spectrum of signal before filtering');
y2=filter(num,den,y);
subplot(212);
plot(psd(spectrum.periodogram,y2,'Fs',Fs,'NFFT',le
ngth(y)));
title('spectrum of signal after high pass filtering');
```