DIT:

```
 x = [0,1, 2, 3];
X = fft(x);
disp('Input Signal:');
disp(x);
disp('FFT of the Signal:');
disp(X);
figure;
subplot(2, 1, 1);
stem(abs(X));
title('Magnitude of FFT');
xlabel('Frequency Index');
ylabel('Magnitude');
grid on;
subplot(2, 1, 2);
stem(angle(X));
title('Phase of FFT');
xlabel('Frequency Index');
ylabel('Phase (radians)');
grid on;
```

DIF:

```
clc;
clear;
x = [0,1, 2, 3];
N = length(x);
disp('Input Signal:');
disp(x);
X0 = x(1) + x(3);
X1 = x(2) + x(4);
X2 = x(1) - x(3);
X3 = (x(2) - x(4)) * exp(-1j * 2 * pi * 1 / 4);
stage1 = [X0, X1, X2, X3];
Y0 = stage1(1) + stage1(2);
Y1 = stage1(3) + stage1(4);
Y2 = stage1(1) - stage1(2);
Y3 = (stage1(3) - stage1(4)) * exp(-1j * 2 * pi * 1 / 4);
DIF_output = zeros(1, 4);
DIF_output(1) = Y0;
DIF_output(3) = Y1;
DIF_output(2) = Y2;
DIF_output(4) = Y3;
disp('DIF FFT Output :');
disp(DIF_output);
freq = 0:N-1;
mag = abs(DIF_output);
phase = angle(DIF_output);
figure;
subplot(2,1,1);
stem(freq, mag);
title('Magnitude of FFT');
xlabel('Frequency Index');
ylabel('Magnitude');
grid on;
subplot(2,1,2);
stem(freq,phase);
title('Phase of FFT');
xlabel('Frequency Index');
ylabel('Phase (radians)');
grid on;
```

FIR:

```
N=51;
fc=0.25;
n=(0:(N-1))-(N-1)/2;
h_ideal=sin(2*pi*fc*n)./(pi*n);
h_ideal((N-1)/2+1)=2*fc;
window=hamming(N);
h_windowed=h_ideal.*window;
h_windowed=h_windowed/sum(h_win
dowed);
figure;
subplot(2,1,1);
stem(h_windowed);
title('Fir filter coefficients (Windowing
method)');
xlabel('Sample index');
ylabel('Amplitude');
grid on;
[Hz,f]=freqz(h_windowed,1,512,'half');
subplot(2,1,2);
plot(f*0.5,20*log10(abs(Hz)));
title('Frequency response(Windowing
method)');
xlabel('Frequency(Hz)');
ylabel('Magnitude(dB)');
grid on;
```

IIR BILR:

```
N = 4;
Wp = 0.2;
Fs = 1;
theta = (2*(1:N) + N - 1) * pi / (2 * N);
poles = exp(1j * theta);
poles = (2 * Fs + poles) ./ (2 * Fs - poles);
function [b, a] = compute_tf_coeffs(poles, k)
  a = 1;
  for i = 1:length(poles)
    a = conv(a, [1 -real(poles(i)) -imag(poles(i))]);
  end
  b = k;
end
[bz, az] = compute_tf_coeffs(poles, 1);
disp('Digital filter coefficients:');
disp('b = ');
disp(bz);
disp('a = ');
disp(az);
[Hz, f] = freqz(bz, az, 512, 'half');
f = f * 0.5;
subplot(2,1,1);
plot(f, 20*log10(abs(Hz)));
title('Magnitude Response of Digital Butterworth
Filter');
xlabel('Frequency (Hz)');
ylabel('Magnitude (dB)');
grid on;
subplot(2,1,2);
plot(f, angle(Hz));
title('Phase Response of Digital Butterworth
Filter');
xlabel('Frequency (Hz)');
ylabel('Phase (radians)');
grid on;
```

SPECTRUM FFT:

```
Fs = 1000;
T = 1 / Fs;
L = 1000;
t = (0:L-1) * T;
f1 = 50;
f2 = 120;
x = 0.7 * sin(2 * pi * f1 * t) +
sin(2 * pi * f2 * t);
X = fft(x);
P2 = abs(X / L);
P1 = P2(1:L/2+1);
P1(2:end-1) = 2 * P1(2:end-1);
f = Fs * (0:(L/2)) / L;
figure;
subplot(2, 1, 1);
plot(t, x);
title('Time Domain Signal');
xlabel('Time (s)');
ylabel('Amplitude');
grid on;
subplot(2, 1, 2);
plot(f, P1);
title('Frequency Domain
Spectrum');
xlabel('Frequency (Hz)');
ylabel('|P1(f)|');
grid on;
figure;
phase = angle(X(1:L/2+1));
plot(f, phase);
title('Phase Spectrum');
xlabel('Frequency (Hz)');
ylabel('Phase (radians)');
grid on;
```

```
N=length(t_continuous);
Y_cont=fft(x_continuous);
f_cont=(0:N-
1)*(Fs_continuous/N);
figure;
plot(f_cont,abs(Y_cont));
title('frequency spectrum of
continuous signal');
xlabel('frequency(Hz)');
ylabel('Magnitude');
grid on;

N_low=length(t_low);
Y_low=fft(x_low,N_low);
f_low=(0:N_low-
1)*(Fs_low/N_low);
figure;
plot(f_low,abs(Y_low));
title('Frequency spectrum of low
sampling rate signal');
xlabel('Frequency(Hz)');
ylabel('Magnitude');
grid on;

N_high=length(t_high);
Y_high=fft(x_high,N_high);
f_high=(0:N_high-
1)*(Fs_high/N_high);
figure;
plot(f_high,abs(Y_high));
title('Frequency spectrum of high
sampling rate signal');
xlabel('Frequency(Hz)');
ylabel('Magnitude');
grid on;
```

SAMPLING:

```
Fs_continuous = 1000;
T = 1 / Fs_continuous;
L = 2;
t_continuous = 0:T:(L-T);
f1 = 50;
f2 = 150;
x_continuous =
sin(2*pi*f1*t_continuous) +
sin(2*pi*f2*t_continuous);
figure;
subplot(3, 1, 1);
plot(t_continuous, x_continuous);
title('Continuous Signal');
xlabel('Time (s)');
ylabel('Amplitude');
grid on;
Fs_low = 80;
Fs_high = 300;
t_low = 0:1/Fs_low:(L-1/Fs_low);
t_high = 0:1/Fs_high:(L-
1/Fs_high);
x_low = sin(2 * pi * f1 * t_low) +
sin(2 * pi * f2 * t_low);
x_high=sin(2 * pi* f1 * t_high) +
sin(2 * pi * f2 * t_high);
subplot(3,1,2);
stem(t_low,x_low,'filled');
title('sampled signal(Low sampling
rate,No aliasing)');
xlabel('Time(s)');
ylabel('Amplitude');
grid on;
```

POLES & ZEROS:

```
b = [0.5, -0.5];
a = [1, -1.8, 0.81];
poles = roots(a);
zeros = roots(b);
disp('Poles of the system:');
disp(poles);
disp('Zeros of the system:');
disp(zeros);
is_stable = all(abs(poles) < 1);
if is_stable
disp('The system is BIBO stable.');
else
disp('The system is NOT BIBO stable.');
end
figure;
hold on;
plot(real(poles), imag(poles), 'rx',
'MarkerSize', 10, 'LineWidth', 2);
plot(real(zeros), imag(zeros), 'bo',
'MarkerSize', 10, 'LineWidth', 2);
theta = linspace(0, 2*pi, 100);
x = cos(theta);
y = sin(theta);
plot(x, y, 'k--', 'LineWidth', 1.5);
title('Poles and Zeros of the System');
xlabel('Real Part');
ylabel('Imaginary Part');
axis equal;
grid on;
legend('Poles', 'Zeros', 'Unit Circle');
hold off;
```

IIR INVARIANCE:

```
a=1000;
Fs=8000;
T=1/Fs;
alpha=exp(-a*T);
Bd=1;
Ad=[1,-alpha];
disp('Digital IIR Filter
H(Z)=B(Z)/A(Z):');
disp(['Numerator:B(z)=',num2str(Bd
)]);
disp(['Denomenator:A(Z)=1-
',num2str(alpha),'*Z{-1}']);
N=512;
[H,w]=freqz(Bd,Ad,N,Fs);
subplot(2,1,1);
plot(w,20*log10(abs(H)));
xlabel('frequency(hz)');
ylabel('magnitude(dB)');
title('Magnitude response of IIR
Filter using impulse invariant
method');
grid on;
subplot(2,1,2);
plot(w, angle(H));
xlabel('frequency (Hz)');
ylabel('phase (radians)');
title('Phase response of IIR Filter
using impulse invariant method');
grid on;
```