```
%Time reveral
clc; clear all; close all;
%unit step signal
n = -10: 10;
u = zeros(size(n));
u = (n>=0) == 1;
subplot(2,1,1); stem(n,u);
title('Original signal');
xlabel('Time -->'); ylabel('Amplitude -->');
axis([-10 10 0 1.3]);

%Time reversal signal
u_reversed = u(end: -1: 1);
subplot(2,1,2); stem(n,u_reversed);
title('Reversed signal');
xlabel('Time -->'); ylabel('Amplitude -->');
axis([-10 10 0 1.3]);
```

```
%time shifting
clc; clear all; close all;
n = 0:5;
x = input('Enter the sequence ');
subplot(2,1,1); stem(n,x);
title('Original signal');
xlabel('Time -->'); ylabel('Amplitude -->');
axis([-10 10 0 5]);

%Shifted signal
m = n+2;
subplot(2,1,2); stem(m,x);
title('Time shifted signal');
xlabel('Time -->'); ylabel('Amplitude -->');
axis([-10 10 0 5]);
```

```
%time scaling
clc; clear all; close all;
n = -10:10;
x = sin(0.2*pi*n);
a = 2;
subplot(2,1,1); stem(n,x);
title('Original signal');
xlabel('Time -->'); ylabel('Amplitude -->');
axis([-10 10 -1.3 1.3]);

n_scaled = n/a;
x_scaled = sin(0.2*pi*n_scaled);
subplot(2,1,2); stem(n_scaled,x_scaled);
title('Time scaled signal');
xlabel('Time -->'); ylabel('Amplitude -->');
axis([-10 10 -1.3 1.3]);
```

```matlab
%Amplitude scaling
clc;clear all; close all;
n = -10:10;
x = sin(0.2*pi*n);
a = 3;
subplot(2,1,1); stem(n,x);
title('Original signal');
xlabel('Time -->'); ylabel('Amplitude -->');
axis([-10 10 -4 4]);

x_scaled = a*x;
subplot(2,1,2); stem(n,x_scaled);
title('Amplitude scaled signal');
xlabel('Time -->'); ylabel('Amplitude -->');
axis([-10 10 -4 4]);
```

```matlab
%Signal  Addition
clc; clear all; close all;
n = 0:6;
x = input('Enter the first sequence: ');
y = input('Enter the second sequence: ');
subplot(3,1,1); stem(x);
title('First signal');
xlabel('Time -->'); ylabel('Amplitude -->');
axis([0 10 0 10]);
subplot(3,1,2); stem(y);
title('Second signal');
xlabel('Time -->'); ylabel('Amplitude -->');
axis([0 10 0 10]);
z = x+y;
subplot(3,1,3); stem(z);
title('Sum of two signals');
xlabel('Time -->'); ylabel('Amplitude -->');
axis([0 10 0 10]);
```

```
%Multiplication of signal
clc; clear all; close all;
n = 0:0.1:5;
a = 2;
x1 = 6*(a.^n);
subplot(3,1,1); stem(n,x1);
title('First signal x1');
xlabel('Time -->'); ylabel('Amplitude -->');
f = 1.2;
x2 = 2*cos(2*pi*f*n);
subplot(3,1,2); stem(n,x2);
title('SEcond signal x2');
xlabel('Time -->'); ylabel('Amplitude -->');
y = x1.*x2;
subplot(3,1,3); stem(n,y);
title('Multiplication of signals y(n)');
xlabel('Time -->'); ylabel('Amplitude -->');
```

```
%convoluton of two sequences
clc; clear all; close all;
x1 = [1 2 0 1];
x2 = [2 2 1 1];
y = conv(x1,x2);
disp('The convolution output is')
disp(y)
subplot(3,1,1); stem(x1);
title('First input signal');
xlabel('Time -->'); ylabel('Amplitude -->');
subplot(3,1,2); stem(x2);
title('Second input signal');
xlabel('Time -->'); ylabel('Amplitude -->');
subplot(3,1,3); stem(y);
title('Convolution output signal');
xlabel('Time -->'); ylabel('Amplitude -->');
```

```
%linear convolution using DFT
clc; clear all; close all;
x = [1 2];
h = [2 1];
x1 = [x zeros(1,length(h)-1)];
%disp(x1)
h1 = [h zeros(1,length(x)-1)];
%disp(h1)
X = fft(x1);
%disp(X)
H = fft(h1);
%disp(H)
y = X.*H;
%disp(y)
y1 = ifft(y);
disp('The linear comvolution is: ')
disp(y1)
```

```
%covolution using z transform
clc; clear all; close all;
x1 = [2 1 0 -1 3];
x2 = [1 -3 2];
x3 = conv(x1,x2)
```

```
%computation of correlation
clc; clear all; close all;
x1 = [1 3 0 4];
y = xcorr(x1,x1);
disp(y)
subplot(2,1,1); stem(x1);
title('Input signal');
xlabel('Time -->'); ylabel('Amplitude -->');
subplot(2,1,2); stem(y);
title('Autocorrelation of signal');
xlabel('Time -->'); ylabel('Amplitude -->');
```

```
%Z transform and inverse z transform
clc; clear all; close all;
syms n wo
a = n+1;
disp('The input signal is');
disp(a)
b = ztrans(a);
disp('The z-transform is')
disp(b)
c = iztrans(b);
disp('The inverse z-transform is')
disp(c)

a1 = cos(wo*n);
disp('The input signal is');
disp(a1)
b1 = ztrans(a1);
disp('The z-transform is')
disp(b1)
c1 = iztrans(b1);
disp('The inverse z-transform is')
disp(c1)
```

DFT Program 6.1

```
% Fourier series representation of a train of pulses

clc; clear all; close all;

syms t % symbolic Fourier Series computation

T= 1; % TO: period

m = heaviside(t)-heaviside (t-T/4)+ heaviside(t-3*T/4);
x = 2 * m; % periodic signal

% [X,w] = fourierseries(x,TO,20); % X,w: Fourier series
coefficients at harmonicfrequencies

N=20; % N: number of harmonics

% computation of N Fourier series coefficients

for k = 1:N,

X1(k) = int(x*exp(-j* 2 *pi*(k-1)*t/T), t, 0, T)/T;

X(k) = subs(X1(k));

w(k) = (k-1)*2*pi/T; % harmonic frequencies

end

ezplot(x,[0 T]),grid;

title('input sequence')

figure

subplot(2,1,1), stem(w,abs(X));

title('magnitude of fourier series')

subplot(2,1,2), stem(w,angle(X));

title('phase of fourier series')
```

DFT Program 6.2

```
% Fourier series representation of a full wave
rectified wave

clc; clear all; close all;

syms t

TO = 1;

m = heaviside(t)-heaviside(t- TO);

x = abs(cos(pi*t))* m;

N=20;

% computation of N Fourier series coefficients

for k = 1:N,

X1(k) = int(x*exp(-j* 2 *pi *(k-1)*t/TO), t, 0, TO)/TO;
X(k) = subs(X1(k));

w(k) = (k-1)*2*pi/TO; % harmonic frequencies

end

ezplot(x,[0 TO]);grid;

title('input sequence')

figure

subplot(2,1,1);stem(w,abs(X));

title('magnitude of fourier series')

subplot(2,1,2); stem(w,angle(X)); title('phase of
fourier series')
```

DFT Program 6.3

```
% Direct computation of Discrete Fourier transform
(matrix formulation)

clc; clear all; close all;

x=[1-1 2 -2];

L=length(x);

y=x*dftmtx(L);

disp('the discrete fourier transform of the input
sequence is')

disp(y)

L1=length(y);

y1=y* conj(dftmtx(L1))/L1;

disp('the Inverse discrete fourier transform of the
input sequence is')

disp(y1)
```

DFT Program 6.4

```
% Linear convolution using DFT

clc; clear all; close all;

x=[1 2];

h=[2 1];

x1=[x zeros(1,length(h)-1)];

h1=[h zeros(1,length(x)-1)];

X=fft(x1);

H=fft(h1);

y=X.*H;

y1=ifft(y);

disp('the linear convolution of the given sequence')

disp(y1)
```

DFT Program 6.5

```
% Circular convolution using DFT

clc; clear all; close all;

x=[1 2 1 2];

h=[4 3 2 1];

X=fft(x);

H=fft(h);

y=X.*H;

y1=real(ifft(y));

disp('the circular convolution of the given sequence')

disp(y1)
```

DFT Program 6.6

```
% Relation between DFTS of the periodic even and odd
parts of a real sequence

clc; clear all; close all;

x=[1 2 4 2 6 32 6 4 2 zeros(1,247)];

x1=[x(1) x(256:-1:2)];

xe=0.5*(x+x1);

xf=fft(x);

xef=fft(xe);

k=0:255;

subplot(2,1,1),
plot(k/128,real(xf));
title('real part of DFT sequence')

subplot(2,1,2),
plot(k/128,imag(xf));
title('imagenary part of DFT sequence')
figure
subplot(2,1,1),
plot(k/128,real(xef));
 title('real part of DFT even sequence')
subplot(2,1,2), plot(k/128,imag(xef));
 title('imaginary part of DFT evensequence')
```