EXP-1-1-SAMPLING

```
% for graphs we change fs
A=5;
f=50;
T=1/f;
fs=100;
%Sine Signal
t=0:0.001:10*T;
y=A*cos(2*pi*f*t);
subplot(3,1,1);
plot(t,y);
title('Nyquist signal')
xlabel('time(secs)')
ylabel('Amplitude')
%Stem Signal
n=0:1/fs:10*T;
xs=cos(2*pi*f*n);
subplot(3,1,2);
stem(n,xs);
title('Nyquist signal')
xlabel('time(secs)')
ylabel('Amplitude')
%Reconstruction of signal(interp- interpolation of signal)
xr=interp(xs,1);
subplot(3,1,3);
plot(n,xr);
title(' Nyquist signal')
xlabel('time(secs)')
ylabel('Amplitude')
```

EXP-1-2 FREQUENCY DOMAIN

```
A=5;
f=50;
T=1/f;
fs=100;
n=0:1/fs:10*T;
xs=cos(2*pi*f*n);
N=length(xs);
Xs=fft(xs);
Xmag=abs(Xs);
Xmagh=Xmag(1:N/2);
freq=(1:N/2)*fs/N;
plot(freq,Xmagh);
title('ANIKET AGRAWAL (EE008)-Nyquist signal')
xlabel('frequency(HZ)')
ylabel('frequency domain signal')
```

Note: **To solve DFT in MATLAB we use FFT term in code and for IDFT we use IFFT**
Freq: frequency; **mag:** Magnitude; **magh:** half of Magnitude; **abs:** Absolute; **fs:** Sample frequency; **L:** Length; **circshift:** Give the same matrix goes till N-1;

EXP 2-1 DFT Computation with formula

```
x=[1 1 1 1 1 1 0 0];
N=8;
for k=0:N-1
   X(k+1)=0;
   for n=0:N-1
      X(k+1)=X(k+1)+x(n+1)*exp(-i*2*pi*n*k/8)
   end
end
disp(X);
Xmag=abs(X);
Xangle=phase(X);
subplot(2,1,1);
x=0:N-1;
stem(x,Xmag);
title(''Aniket Agrawal (EE008)- Magnitude of X(k)');
xlabel('n');
ylabel('Magnitude');
subplot(2,1,2);
x=0:N-1;
stem(x,Xangle);
title('Aniket Agrawal (EE008)- Phase of X(k)');
xlabel('n');
ylabel('Magnitude');
```

EXP 2-2 Frequency domain analysis of Sinusoidal Analysis

```
f1=50,f2=150,f3=300;
fs=1000;L=1500;
t=(0:L-1)*1/fs;
y=sin(2*pi*f1*t)+sin(2*pi*f2*t)+sin(2*pi*f3*t);
%plot(t(1:100),y(1:100));
Y=fft(y);
N=length(y);
Ymag=abs(Y)/N;
Ymagh=Ymag(1:N/2);
freq=(1:N/2)*fs/N;
plot(freq,Ymagh);
title('Aniket Agrawal (EE008)- Frequency domain signal');
xlabel('Frequency(Hz)');
ylabel('Magnitude');
```

Note: **To solve DFT in MATLAB we use FFT term in code and for IDFT we use IFFT**
Freq: frequency; **mag:** Magnitude; **magh:** half of Magnitude; **abs:** Absolute; **fs:** Sample frequency; **L:** Length; **circshift:** Give the same matrix goes till N-1;

EXP-3

Circular convolution using Summation formula

```
close all;
x1=[1 4 3 2];
x2=[1 2 3 1];
N=4;
for n=0:N-1
   x(n+1)=sum(x1.*circshift(flip(x2),n+1));
end
disp(x);
```

Circular convolution using DFT and IDFT

```
x1=[1 4 3 2];
x2=[1 2 3 1];
X1=fft(x1);
X2=fft(x2);
X3=X1.*X2;
x3=ifft(X3);
disp(x3);
```

Linear convolution using Summation Method

```
x1=[1 2 -1 3 5];
x2=[1 -1];
L=length(x1);
M=length(x2);
N=L+M-1;
for n=1:N
   x(n)=0;
   for k=1:L
      if (n-k+1)>1 && (n-k+1)<=M
      x(n)=x(n)+x1(k)*x2(n-k+1);
      end
   end
end
disp(x);
disp(conv(x1,x2));
```

Linear Convolution using Circular Convolution

```
x1=[1 2 -1 3 5];
x2=[1 -1];
L=length(x1);M=length(x2);N=L+M-1;
xx1=[x1 zeros(M-1,1)];
xx2=[x2 zeros(1,L-1)];
X1=fft(xx1);
X2=fft(xx2);
X3=X1.*X2;
x3=ifft(X3);
disp(x3);
```

Note: **To solve DFT in MATLAB we use FFT term in code and for IDFT we use IFFT**
Freq: frequency; **mag:** Magnitude; **magh:** half of Magnitude; **abs:** Absolute; **fs:** Sample frequency; **L:** Length; **circshift:** Give the same matrix goes till N-1;

EXP-4 Analysis of Audio Files (To read and analyze spectrum of Audio Files)

Code:

To record audio and analyze:

```
clc;
clear;
fs=22050;
x=audiorecorder(fs,16,1);
disp('start');
recordblocking(x,5);
disp('stop');
y=getaudio(x);
sound(y);
subplot(2,1,1);
plot(y);
N=length(y);
Y=fft(y);
Ymag=abs(y)/N;
Ymagh=Ymag(1:N/2);
f=(1:N/2)*fs/N;
subplot(2,1,2);
plot(f,Ymagh);
```

To read an audio file and analyze

```
% To read an audio file and analyze
% Audio file and Matlab file should be in same folder

clc;
clear;
[y,fs]=audioread('audio.mp3');
sound(y);
subplot(2,1,1);
plot(y(:,1));
title('Original Signal-Aniket Agrawal (EE008)')
N=length(y);
Y=fft(y(:,1));
Ymag=abs(Y)/N;
Ymagh=Ymag(1:N/2);
f=(1:N/2)*fs/N;
subplot(2,1,2);
plot(f,Ymagh);
title('Frequency Spectrum - Aniket Agrawal (EE008)')
```

Note: **To solve DFT in MATLAB we use FFT term in code and for IDFT we use IFFT**
Freq: frequency; **mag:** Magnitude; **magh:** half of Magnitude; **abs:** Absolute; **fs:** Sample frequency; **L:** Length; **circshift:** Give the same matrix goes till N-1;

EXP-5

1. **Basic Image Processing**

```
clc;
clear;
I=imread("C:\Users\ADMIN\Desktop\rose.jpg");
subplot(2,2,1);
imshow(I);
title("Original Aniket-EE008");
Ir=imresize(I,0.5);
subplot(2,2,2);
imshow(Ir);
title("Resize Aniket-EE008");
Ig=rgb2gray(I);
subplot(2,2,3);
imshow(Ig);
title("Greyscale Aniket-EE008");
Ibw=imbinarize(Ig);
subplot(2,2,4);
imshow(Ibw);
title("Binary Aniket-EE008");
```

2. **Detecting horizontal and vertical edges**

```
clc;
clear;
I=imread("C:\Users\ADMIN\Desktop\rose.jpg");
subplot(2,2,1);imshow(I);title("Original Aniket-EE008");
Ig=rgb2gray(I);
subplot(2,2,2);imshow(Ig);title("Greyscale Aniket-EE008");
H=[ 1 1 1; 0 0 0; -1 -1 -1 ];
V=[ 1 0 -1; 1 0 -1; 1 0 -1];
Ih=conv2(Ig,H,'same');
Iv=conv2(Ig,V,'same');
subplot(2,2,3);imshow(Ih,[]);title("Horizontal Edges Aniket-EE008");
subplot(2,2,4);imshow(Iv,[]);title("Vertical Edges Aniket-EE008");
```

Note: **To solve DFT in MATLAB we use FFT term in code and for IDFT we use IFFT**

Freq: frequency; **mag:** Magnitude; **magh:** half of Magnitude; **abs:** Absolute; **fs:** Sample frequency; **L:** Length; **circshift:** Give the same matrix goes till N-1;

EXP-6 Noise Reduction in Signals

1. **Eliminate noise from the audio signal**

```
clc;
clear;
%original signal
[y,fs]=audioread('audio.mp3');
sound(y);
subplot(3,2,1);
plot(y(:,1));
title('Original audio signal(Aniket Agrawal (EE008))');
N = length(y);
Y = fft(y(:,1));
Ymag=abs(Y)/N;
Ymagh=Ymag(1:N/2);
f=(1:N/2)*fs/N;
subplot(3,2,2);
plot(f,Ymagh);
title('Freq.spec. or.au.sig.(Aniket Agrawal (EE008))');
%Adding noie
yn=y+randn(N,1)/10;sound(y);
sound(yn);
subplot(3,2,3);
plot(yn(:,1));
title('Audiowithnoise(Aniket Agrawal (EE008))');
Yn = fft(y(:,1));
Ymagn=abs(Yn)/N;
Ymaghn=Ymagn(1:N/2);
subplot(3,2,4);
plot(f,Ymaghn);
title('Frequencyspectrumwith noise(Aniket Agrawal (EE008))');
%filtering noise
n=7;
fc=2000/fs;
[b,a]=butter(n,fc,'low');
yf=filter(b,a,yn);
sound(yf);
subplot(3,2,5);
plot(yf(:,1));
title('audio signalfiltering(Aniket Agrawal (EE008))');
Yf= fft(y(:,1));
Ymagf=abs(Yf)/N;
Ymaghf=Ymagf(1:N/2);
f=(1:N/2)*fs/N;
subplot(3,2,6);
plot(f,Ymaghf);
title('freqspec of au. sig.(Aniket Agrawal (EE008))');
```

Note: **To solve DFT in MATLAB we use FFT term in code and for IDFT we use IFFT**
Freq: frequency; **mag:** Magnitude; **magh:** half of Magnitude; **abs:** Absolute; **fs:** Sample frequency; **L:** Length; **circshift:** Give the same matrix goes till N-1;

2. Eliminate salt and pepper noise from image signal

```
clc;
clear;
I=imread("coins.png");
subplot(2,2,1);imshow(I);title("Original Aniket-EE008");
In=imnoise(I,'salt & pepper',0.02);
subplot(2,2,2);imshow(In);title("Adding salt pepper Aniket-EE008");
Iav=filter2(fspecial('average',3),In)/255;
subplot(2,2,3);imshow(Iav);title("Average Filter Aniket-EE008");
Im=medfilt2(In);
subplot(2,2,4);imshow(Im);title("Median Filter Aniket-EE008");
```

Note: **To solve DFT in MATLAB we use FFT term in code and for IDFT we use IFFT**
Freq: frequency; **mag:** Magnitude; **magh:** half of Magnitude; **abs:** Absolute; **fs:** Sample frequency; **L:** Length; **circshift:** Give the same matrix goes till N-1;

Experiment -7

DTMF generation and Filtering

```
clc; clear;
fs=5000; T=0.4; t=0:1/fs:T;
frow=[697 770 852 941];
fcol=[1209 1336 1477];
dialednum=1234567890;
dialednum=num2str(dialednum);
tonesig=[];
for i=1:length(dialednum)
 switch dialednum(i)
 case '1'
 f1=frow(1); f2=fcol(1);
 case '2'
 f1=frow(1); f2=fcol(2);
 case '3'
 f1=frow(1); f2=fcol(3);
 case '4'
 f1=frow(2); f2=fcol(1);
 case '5'
 f1=frow(2); f2=fcol(2);
 case '6'
 f1=frow(2); f2=fcol(3);
 case '7'
 f1=frow(3); f2=fcol(1);
 case '8'
 f1=frow(3); f2=fcol(2);
```

Note: **To solve DFT in MATLAB we use FFT term in code and for IDFT we use IFFT**
Freq: frequency; **mag:** Magnitude; **magh:** half of Magnitude; **abs:** Absolute; **fs:** Sample frequency; **L:** Length; **circshift:** Give the same matrix goes till N-1;

```
 case '9'
 f1=frow(3); f2=fcol(3);
 case '0'
 f1=frow(4); f2=fcol(2);
 end
 tone=sin(2*pi*f1*t)+sin(2*pi*f2*t);
 tonesig=[tonesig; tone'; zeros(size(tone'))];
end
soundsc(tonesig,fs);
subplot(2,1,1); plot(tonesig);
title('DTMF signal (Aniket-21EE008)');
xlabel('Time');
ylabel('Magnitude');
N=length(tonesig);
X=abs(fft(tonesig))/N;
X=2*X(1:N/2+1);
f=(0:N/2)*fs/N;
subplot(2,1,2); plot(f,X);
title('Frequency spectrum of DTMF signal (Aniket-21EE008)');
xlabel('Frequency');
ylabel('Magnitude');
```

Note: To solve DFT in MATLAB we use FFT term in code and for IDFT we use IFFT

Freq: frequency; mag: Magnitude; magh: half of Magnitude; abs: Absolute; fs: Sample frequency;

L: Length; circshift: Give the same matrix goes till N-1;

```
%Decoding the DTMF signal
a=1;
```

Note: **To solve DFT in MATLAB we use FFT term in code and for IDFT we use IFFT**
Freq: frequency; **mag:** Magnitude; **magh:** half of Magnitude; **abs:** Absolute; **fs:** Sample frequency; **L:** Length; **circshift:** Give the same matrix goes till N-1;

```
L=length(t);
while(a<N)
 z=tonesig(a:a+L);
 Z=abs(fft(z));
 Z=Z(1:L/2+1); fr=(0:L/2)*fs/L;
 [pks,locs]=findpeaks(Z,fr,'MinPeakHeight',20,'MinPeakDistance',50);
 fa=locs(1);
 fb=locs(2);
 if fa<=720 && fb<=1250
 disp('1');
 elseif fa<=720 && fb<=1400
 disp('2');
 elseif fa<=720 && fb<=1500
 disp('3');
 elseif fa<=800 && fb<=1250
 disp('4');
 elseif fa<=800 && fb<=1400
 disp('5');
 elseif fa<=800 && fb<=1500
 disp('6');
 elseif fa<=900 && fb<=1250
 disp('7');
 elseif fa<=900 && fb<=1400
 disp('8');
 elseif fa<=900 && fb<=1500
 disp('9');
 else
```

Note: **To solve DFT in MATLAB we use FFT term in code and for IDFT we use IFFT**
Freq: frequency; **mag:** Magnitude; **magh:** half of Magnitude; **abs:** Absolute; **fs:** Sample frequency; **L:** Length; **circshift:** Give the same matrix goes till N-1;

```
 disp('0');
 end
 a=a+2*L;
endExperiment -7
DTMF generation and Filtering
clc; clear;
fs=5000; T=0.4; t=0:1/fs:T;
frow=[697 770 852 941];
fcol=[1209 1336 1477];
dialednum=1234567890;
dialednum=num2str(dialednum);
tonesig=[];
for i=1:length(dialednum)
 switch dialednum(i)
 case '1'
 f1=frow(1); f2=fcol(1);
 case '2'
 f1=frow(1); f2=fcol(2);
 case '3'
 f1=frow(1); f2=fcol(3);
 case '4'
 f1=frow(2); f2=fcol(1);
 case '5'
 f1=frow(2); f2=fcol(2);
 case '6'
 f1=frow(2); f2=fcol(3);
 case '7'
```

Note: **To solve DFT in MATLAB we use FFT term in code and for IDFT we use IFFT**
Freq: frequency; **mag:** Magnitude; **magh:** half of Magnitude; **abs:** Absolute; **fs:** Sample frequency; **L:** Length; **circshift:** Give the same matrix goes till N-1;

```
 f1=frow(3); f2=fcol(1);
 case '8'
 f1=frow(3); f2=fcol(2);
 case '9'
 f1=frow(3); f2=fcol(3);
 case '0'
 f1=frow(4); f2=fcol(2);
 end
 tone=sin(2*pi*f1*t)+sin(2*pi*f2*t);
 tonesig=[tonesig; tone'; zeros(size(tone'))];
end
soundsc(tonesig,fs);
subplot(2,1,1); plot(tonesig);
title('DTMF signal (Aniket-21EE008)');
xlabel('Time');
ylabel('Magnitude');
N=length(tonesig);
X=abs(fft(tonesig))/N;
X=2*X(1:N/2+1);
f=(0:N/2)*fs/N;
subplot(2,1,2); plot(f,X);
title('Frequency spectrum of DTMF signal (Aniket-21EE008)');
xlabel('Frequency');
ylabel('Magnitude');
```

Note: To solve DFT in MATLAB we use FFT term in code and for IDFT we use IFFT

Freq: frequency; mag: Magnitude; magh: half of Magnitude; abs: Absolute; fs: Sample frequency;

Note: **To solve DFT in MATLAB we use FFT term in code and for IDFT we use IFFT**
Freq: frequency; **mag:** Magnitude; **magh:** half of Magnitude; **abs:** Absolute; **fs:** Sample frequency; **L:** Length; **circshift:** Give the same matrix goes till N-1;

```
L: Length; circshift: Give the same matrix goes till N-1;

%Decoding the DTMF signal

a=1;

L=length(t);

while(a<N)

 z=tonesig(a:a+L);

 Z=abs(fft(z));

 Z=Z(1:L/2+1); fr=(0:L/2)*fs/L;

 [pks,locs]=findpeaks(Z,fr,'MinPeakHeight',20,'MinPeakDistance',50);

 fa=locs(1);

 fb=locs(2);

 if fa<=720 && fb<=1250

 disp('1');

 elseif fa<=720 && fb<=1400

 disp('2');

 elseif fa<=720 && fb<=1500

 disp('3');

 elseif fa<=800 && fb<=1250

 disp('4');

 elseif fa<=800 && fb<=1400

 disp('5');

 elseif fa<=800 && fb<=1500

 disp('6');

 elseif fa<=900 && fb<=1250

 disp('7');

 elseif fa<=900 && fb<=1400

 disp('8');
```

Note: **To solve DFT in MATLAB we use FFT term in code and for IDFT we use IFFT**
Freq: frequency; **mag:** Magnitude; **magh:** half of Magnitude; **abs:** Absolute; **fs:** Sample frequency; **L:** Length; **circshift:** Give the same matrix goes till N-1;

```
 elseif fa<=900 && fb<=1500
 disp('9');
 else
 disp('0');
 end
 a=a+2*L;
end
```

Note: **To solve DFT in MATLAB we use FFT term in code and for IDFT we use IFFT**
Freq: frequency; **mag:** Magnitude; **magh:** half of Magnitude; **abs:** Absolute; **fs:** Sample frequency; **L:** Length; **circshift:** Give the same matrix goes till N-1;

EXP-9-1- Low pass Filter with cut off frequency 0.5pi and order 10

```
clc;clear;
N=10;
M=N+1;
wc=0.5;
win=ones(M,1);
b=fir1(N,wc,'low',win);
disp(b);
[H,W]=freqz(b,1);
plot(W/pi,mag2db(abs(H)));
grid on;
title('Low pass Filter with cut off Frequency 0.5pi and order 10 - Aniket
Agrawal(EE008)');
xlabel('Normalized frequency');
ylabel('Magnitude in dB');
```

EXP-9-2- High pass Filter with cut off frequency 0.48pi and order 12

```
clc;clear;
N=12;
M=N+1;
wc=0.48;
win=ones(M,1);
b=fir1(N,wc,'high',win);
disp(b);
[H,W]=freqz(b,1);
plot(W/pi,mag2db(abs(H)));
grid on;
title('High pass Filter with cut off Frequency 0.48pi and order 12 - Aniket
Agrawal(EE008)');
xlabel('Normalized frequency');
ylabel('Magnitude in dB');
```

EXP-9-3- Band pass Filter with cut off frequency 0.35pi,0.65pi and order 10

```
clc;clear;
N=10;
M=N+1;
wc=[0.35,0.65];
n=0:M-1;
win=(0.54-0.46*cos((2*pi*n)/(M-1)));
b=fir1(N,wc,'bandpass',win);
disp(b);
[H,W]=freqz(b,1);
plot(W/pi,mag2db(abs(H)));
grid on;
title('Band pass Filter with cut off Frequency 0.35pi,0.65pi and order 10 -
Aniket Agrawal(EE008)');
xlabel('Normalized frequency');
ylabel('Magnitude in dB');
```

EXP-9-4- Band Stop Filter with cut off frequency 0.42pi,0.75pi and order 8

Note: **To solve DFT in MATLAB we use FFT term in code and for IDFT we use IFFT**
Freq: frequency; **mag:** Magnitude; **magh:** half of Magnitude; **abs:** Absolute; **fs:** Sample frequency; **L:** Length; **circshift:** Give the same matrix goes till N-1;

```
clc;clear;
N=8;
M=N+1;
wc=[0.42,0.75];
n=0:M-1;
win=(0.54-0.46*cos((2*pi*n)/(M-1)));
b=fir1(N,wc,'stop',win);
disp(b);
[H,W]=freqz(b,1);
plot(W/pi,mag2db(abs(H)));
grid on;
title('Band stop Filter with cut off Frequency 0.35pi,0.65pi and order 10 -
Aniket Agrawal(EE008)');
xlabel('Normalized frequency');
ylabel('Magnitude in dB');
```

Note: **To solve DFT in MATLAB we use FFT term in code and for IDFT we use IFFT**

Freq: frequency; **mag:** Magnitude; **magh:** half of Magnitude; **abs:** Absolute; **fs:** Sample frequency; **L:** Length; **circshift:** Give the same matrix goes till N-1;

Exp-10-1 Low pass Butterworth Filter

```
clc;clear;
wp=0.25*pi;
ws=0.55*pi;
Ap=0.5;As=15;T=1;
Op=(2/T)*tan(wp/2);
Os=(2/T)*tan(ws/2);
[N,Oc]=buttord(Op,Os,Ap,As,'s');
disp(N);disp(Oc);
[b,a]=butter(N,Oc,'s');
tf(b,a)
[bz,az]=bilinear(b,a,1/T);
tf(bz,az,-1,'variable','z^-1')
[H,W]=freqz(bz,az);
plot(W/pi,mag2db(abs(H)));
ylim([-30 5]);
grid on;
title('Low Pass Butterworth Filter(Aniket-EE008)');
xlabel('Normalized Frequency');
ylabel('Gain in dB');
```

Exp-10-2 Low pass ChebyShev Filter

```
clc;clear;
wp=0.25*pi;
ws=0.55*pi;
Ap=0.5;As=15;T=1;
Op=(2/T)*tan(wp/2);
Os=(2/T)*tan(ws/2);
[N,Oc]=cheb1ord(Op,Os,Ap,As,'s');
disp(N);disp(Oc);
[b,a]=cheby1(N,Ap,Op,'s');
tf(b,a)
[bz,az]=bilinear(b,a,1/T);
tf(bz,az,-1,'variable','z^-1')
[H,W]=freqz(bz,az);
plot(W/pi,mag2db(abs(H)));
ylim([-30 5]);
grid on;
title('Low Pass Chebyshev Filter(Aniket-EE008)');
xlabel('Normalized Frequency');
ylabel('Gain in dB');
```

Experiment -7

DTMF generation and Filtering

Note: **To solve DFT in MATLAB we use FFT term in code and for IDFT we use IFFT**
Freq: frequency; **mag:** Magnitude; **magh:** half of Magnitude; **abs:** Absolute; **fs:** Sample frequency; **L:** Length; **circshift:** Give the same matrix goes till N-1;

EXP-11

Linear Convolution

```
#include<stdio.h>
int x[15],h[15],y[15];
int i,j,m,n;
void main()
{
      printf("\n Enter the value for m:");
      scanf("%d", &m);
      printf("\n Enter the value for n:");
    scanf("%d",&n);
    printf(" Enter values for i/p x(n):  \n");
    for(i = 0;i< m;i++)
      scanf("%d",&x[i]);
    printf("enter values for i/p h(n) : \n");
    for(i=0;i<n;i++)
      scanf("%d",&h[i]);
    for(i=m;i<m+n-1;i++)
      x[i]=0;
    for(i=n;i<=m+n-1;i++)
      h[i]=0;
    for(i=0;i<m+n-1;i++)
    {

      y[i]=0;
      for(j=0;j<=i;j++)
            y[i] = y[i]+(x[j]*h[i-j]);
    }

    for(i=0;i<m+n-1;i++)
      printf("\n The value of output y[%d]=%d",i,y[i]);
}
```

Note: **To solve DFT in MATLAB we use FFT term in code and for IDFT we use IFFT**
Freq: frequency; **mag:** Magnitude; **magh:** half of Magnitude; **abs:** Absolute; **fs:** Sample frequency; **L:** Length; **circshift:** Give the same matrix goes till N-1;

Circular Convolution

```c
#include<stdio.h>
int x[30], h[30], y[30], x2[30], a[30];
int m,n,i,j,k;
void main()
{
	printf("\n Enter the length of the first sequence: ");
	scanf("%d", &m);
	printf("\n Enter the length of the second sequence: ");
	scanf("%d", &n);
	printf("\n Enter the first sequence: \n");
	for(i = 0; i < m; i++)
		scanf("%d",&x[i]);
	printf("\n Enter the second sequence: \n");
	for(j = 0; j < n; j++)
		scanf("%d", &h[j]);
	y[0] = 0;
	a[0] = h[0];
	for(j = 1; j <n; j++)
		a[j] = h[n-j];
	for(i = 0; i < n; i++)
		y[0] += x[i]*a[i];
	for(k = 1; k < n; k++)
	{
		y[k] = 0;
		for(j = 1; j < n; j++)
			x2[j] = a[j-1];
		x2[0] = a[n-1];
		for(i = 0; i < n; i++)
		{
			a[i] = x2[i];
			y[k] += x[i] * x2[i];
		}
	}
	printf("The circular convolution is : \n");
	for(i = 0; i < n; i++)
		printf("%d \t", y[i]);
}
```

Note: **To solve DFT in MATLAB we use FFT term in code and for IDFT we use IFFT**

Freq: frequency; **mag:** Magnitude; **magh:** half of Magnitude; **abs:** Absolute; **fs:** Sample frequency; **L:** Length; **circshift:** Give the same matrix goes till N-1;

DFT Calculation

```
#include<stdio.h>
#include<math.h>
int N,K,n,i;
float pi = 3.1416,sumre=0,sumim=0,out_real[8]={0.0},out_imag[8]={0.0};
int x[32];
void main(void)
{
	printf("enter the length of the sequence");
	scanf("%d",&N);
	printf("enter the sequence:");
	for(i=0;i<N;i++)
		scanf("%d",&x[i]);
	for(K=0;K<N;K++)
	{
		sumre = 0;
		sumim = 0;
		for(n=0;n<N;n++)
		{
			sumre = sumre+x[n]*cos(2*pi*K*n/N);
			sumim = sumim-x[n]*sin(2*pi*K*n/N);

		}

		out_real[K]=sumre;
		out_imag[K] = sumim;
		printf("x[%d]) = \t%f\+\t%fi\n",K,out_real[K],out_imag[K]);
	}
}
```

Note: **To solve DFT in MATLAB we use FFT term in code and for IDFT we use IFFT**

Freq: frequency; **mag:** Magnitude; **magh:** half of Magnitude; **abs:** Absolute; **fs:** Sample frequency; **L:** Length; **circshift:** Give the same matrix goes till N-1;