```
% GENERATION OF BASIC SIGNALS %
% UNIT IMPULSE %
l=input('length')
n=1:l
uimp=[1,zeros(1,l-1)]
subplot(3,3,1)
stem(uimp)
xlabel('Time')
ylabel('Amplitude')
title('unit impulse waveform');

% STEP %
step=ones(1,l)
subplot(3,3,2)
stem(step)
xlabel('Time')
ylabel('Amplitude')
title('step waveform');

% RAMP %
ramp=(n-1).*step
subplot(3,3,3)
stem(ramp)
xlabel('Time')
ylabel('Amplitude')
title('ramp waveform');
```



```
% SINE WAVE %
F=input('frequency')
t=0:0.001:pi
a=sin(2.*pi.*F.*t)
subplot(3,3,4)
plot(a)
xlabel('Time')
ylabel('Amplitude')
title('sine waveform');

% SAW TOOTH %
b=sawtooth(2.*pi.*F.*t)
subplot(3,3,5)
plot(b)
xlabel('Time')
ylabel('Amplitude')
title('sawtooth waveform');
```

```
h=input('sequence 2')
```

% LINEAR CONVOLUTION %

```
a=conv(x,h)
subplot(3,1,1)
stem(a)
xlabel('Time')
ylabel('Amplitude')
title('Linear Convolution')
```

% CROSS CORRELATION %

```
b=xcorr(x,h)
subplot(3,1,2)
stem(b)
xlabel('Time')
ylabel('Amplitude')
title('Cross Correlation')
```

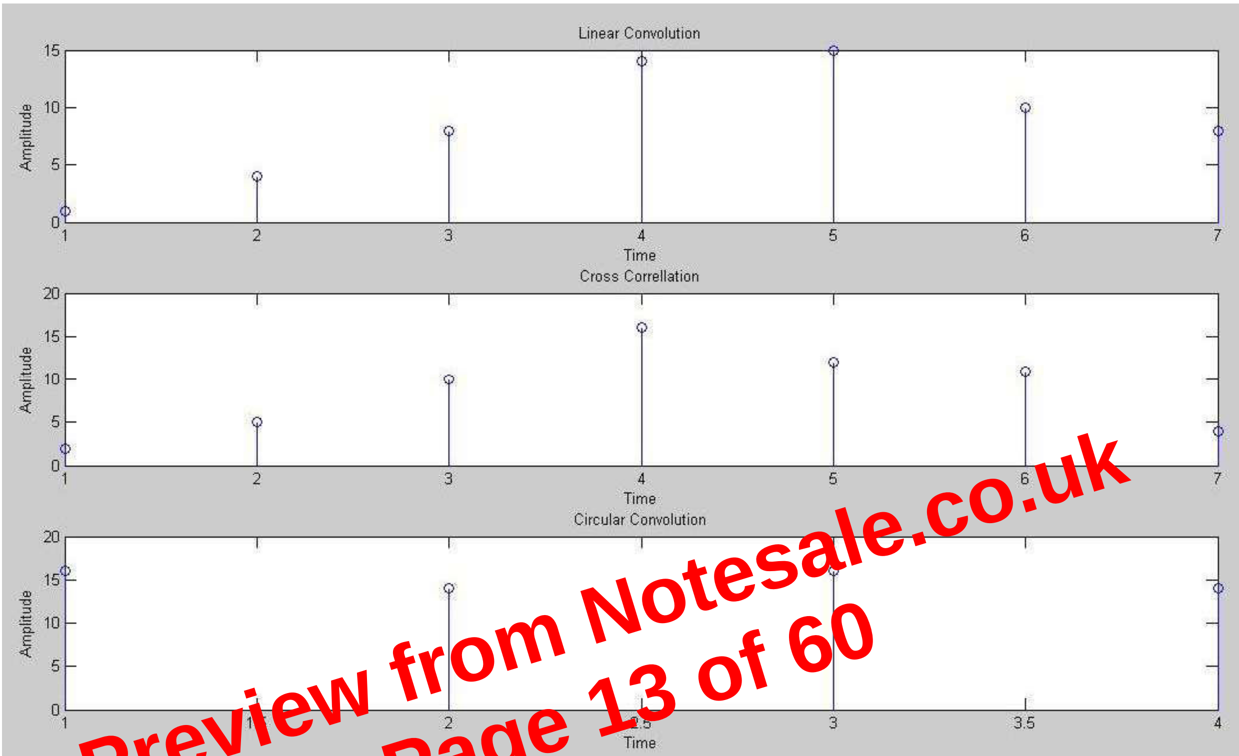

% CIRCULAR CONVOLUTION %

```
n1=length(x);
n2=length(h);
N=max(n1,n2);
x=[x zeros(1,(N-n1))];
for i=1:N
   k=i;
   for j=1:n2
      H(i,j)=x(k)*h(j);
      k=k-1;
      if(k==0)
         k=N;
      end
```

a = 1   4   8   14   15   10   8

b = 2.0000   5.0000   10.0000   16.0000   12.0000   11.0000   4.0000

Output is y(n)= 16   14   16   14



5. **Computation And Verification Of FFT Of A Sequence**

% FFT OF A SEQUENCE %

```
x=input('enter the sequence')
N=input('enter the length of FFT')
y=fft(x,N);
stem(y);
xlabel('real axis');
```

➔ **INPUT**

Order=20

Frequency=20

Sampling frequency=100

OUTPUT

wn = 0.4000

h = **Columns 1 through 12**

-0.0000 0.0057 0.0063 -0.0103 -0.0257 0.0000 0.0570 0.0533 -0.0874 -0.2981 0.6010 -0.2981

Columns 13 through 21

-0.0874 0.0533 0.0570 0.0000 -0.0257 -0.0103 0.0063 0.0057 -0.0000

% GET THE FILTER OUTPUT USING CCSTUDIO %

```
#include<stdio.h>
#include<math.h>
float
h[21]={-0.0000,0.0057,0.0063,-0.0103,-0.0257,0.0000,0.0570,0.0533,-0.0874,-0.2981,0.6010,-0.2981,-0.0874,0.0533,0.0570,0.0000,-0.0257,-0.0103,0.0063,0.0057,-0.0000};

int xlength=60,hlength=20,N;
float x[60],y[70];
main()
{
```

21. **Program to realize and design IIR butterworth high pass filter**

% GET THE FILTER COEFFICIENTS USING MATLAB %

```
wp=input('pass band edge frequency')
ws=input('stop band edge frequency')
rp=input('pass band ripple')
rs=input('stop band attenuation')
[N,wn]=buttord(wp,ws,rp,rs)
[B,A]=butter(N,wn,'high')
freqz(B,A)
```

 INPUT

pass band edge frequency= 0.2
stop band edge frequency = 0.8
pass band ripple =3
stop band attenuation =30

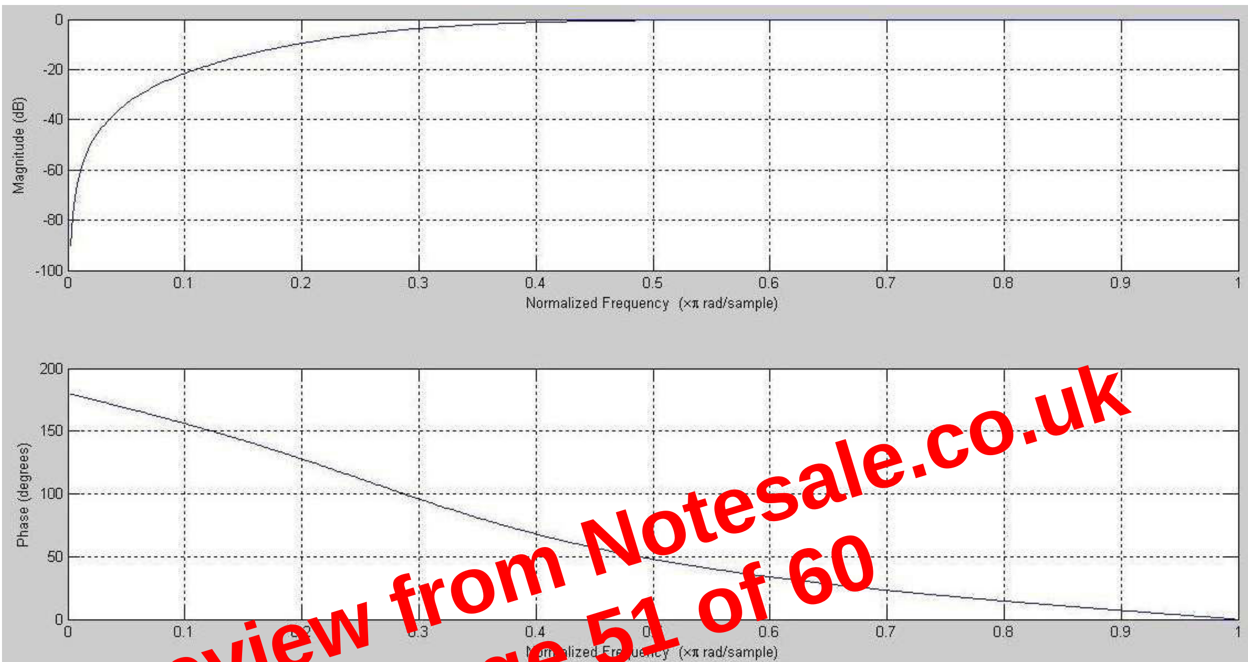
➔ OUTPUT

N = 2

wn = 0.3189

B = 0.4822 -0.9644 0.4822

A = 1.0000 -0.6754 0.2534

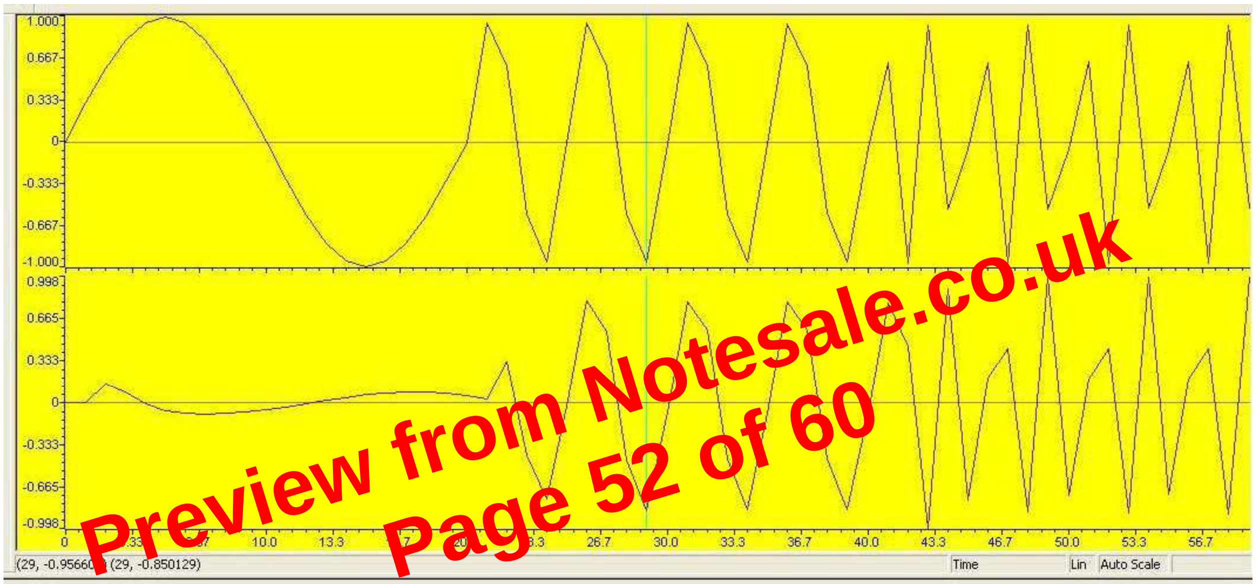


% GET THE FILTER OUTPUT USING CCSTUDIO %

```
#include<stdio.h>
#include<math.h>
float b0=0.4822,b1=-0.9644,b2=0.4822; /*obtained from matlab */
float a0=1.0000,a1=-0.6754,a2=0.2534; /*obtained from matlab */
int n=60; /*no. of input samples */
float dn=0,dn1=0,dn2=0; /* initialise delay buffers */
float x[60],y[70];
main()
{
int k,i;
/* generate input samples of 50, 200 and 400 Hz at sampling freq 1KHz */
for(i=0;i<20;i++)
x[i]=sin(2*3.14*i*50/1000);
for(i=20;i<40;i++)
x[i]=sin(2*3.14*i*200/1000);
for(i=40;i<60;i++)
x[i]=sin(2*3.14*i*400/1000);
```

```
/* generate output samples */
for(k=0;k<n;k++)
{
y[k]=b0*dn+b1*dn1+b2*dn2;
/* update delay buffers */
dn2=dn1;
dn1=dn;
dn=x[k]-a1*dn1-a2*dn2;
}
}
```

➔ **OUTPUT**



22. Program to realize and design IIR chebyshev type 1 low pass filter

% GET THE FILTER COEFFICIENTS USING MATLAB %

```
wp=input('pass band edge frequency')
ws=input('stop band edge frequency')
rp=input('pass band ripple')
rs=input('stop band attenuation')
[N,wn]=cheb1ord(wp,ws,rp,rs)
[B,A]=cheby1(N,rp,wn)
freqz(B,A)
```

➔ **INPUT**

pass band edge frequency = 0.2

stop band edge frequency = 0.8

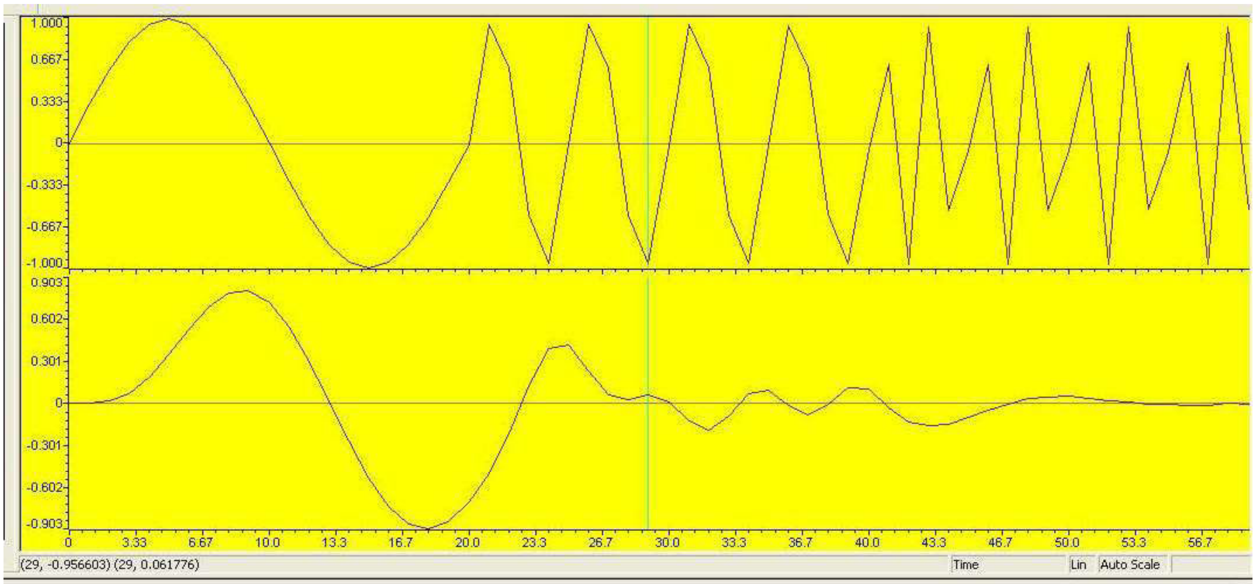
pass band ripple = 3

stop band attenuation = 30

```
{
int k,i;
/* generate input samples of 50, 200 and 400 Hz at sampling freq 1KHz */
for(i=0;i<20;i++)
x[i]=sin(2*3.14*i*50/1000);
for(i=20;i<40;i++)
x[i]=sin(2*3.14*i*200/1000);
for(i=40;i<60;i++)
x[i]=sin(2*3.14*i*400/1000);

/* generate output samples */
for(k=0;k<n;k++)
{
y[k]=b0*dn+b1*dn1+b2*dn2;
/* update delay buffers */
dn2=dn1;
dn1=dn;
dn=x[k]-a1*dn1-a2*dn2;
}
}
```


OUTPUT



23.**Program to realize and design IIR chebyshev type 1 high pass filter**

% GET THE FILTER COEFFICIENTS USING MATLAB %


```
wp=input('pass band edge frequency')
ws=input('stop band edge frequency')
rp=input('pass band ripple')
rs=input('stop band attenuation')
[N,wn]=cheb1ord(wp,ws,rp,rs)
[B,A]=cheby1(N,rp,wn,'high')
freqz(B,A)
```

➔ **INPUT**

pass band edge frequency = 0.2

stop band edge frequency = 0.8

pass band ripple= 3

stop band attenuation=30