1. **GENERATION OF ELEMENTARY DISCRETE TIME SEQUENCES**

*//Generate an impulse sequence*

x1=-10:10;

y1=[zeros(1,10),ones(1,1),zeros(1,10)];

subplot(3,2,1);

plot2d3(x1,y1);

xlabel('sequence n ◊ ');

ylabel('x(n)');

title('UNIT IMPULSE');

*//Generate an unit step sequence*

x2=-20:20;

y2=[zeros(1,20),ones(1,1),ones(1,20)];

subplot(3,2,2);

plot2d3(x2,y2);

xlabel('sequence n ◊');

ylabel('u(n)');

title('UNIT STEP');

*//Generate an unit ramp sequence*

n=input('enter the value');

x3=0:n-1;

N=0:n-1;

subplot(3,2,3);

plot2d3(x3,N);

xlabel('sequence n ◊');

ylabel('Ur(n)');

title('UNIT RAMP');

*//Generate a real decaying exponential signal*

a=0.8;

n=0:20;

x=a.^n;

subplot(3,2,4);

plot2d3(n,x);

xlabel('sequence n ◊');

ylabel('x(n)');

title('exponential signal');

*//Generate a Sinusoidal signal*

a=0.8;

n=0:50;

x=a\*sin(2\*%pi\*1000\*n\*(1/20000));

subplot(3,2,6);

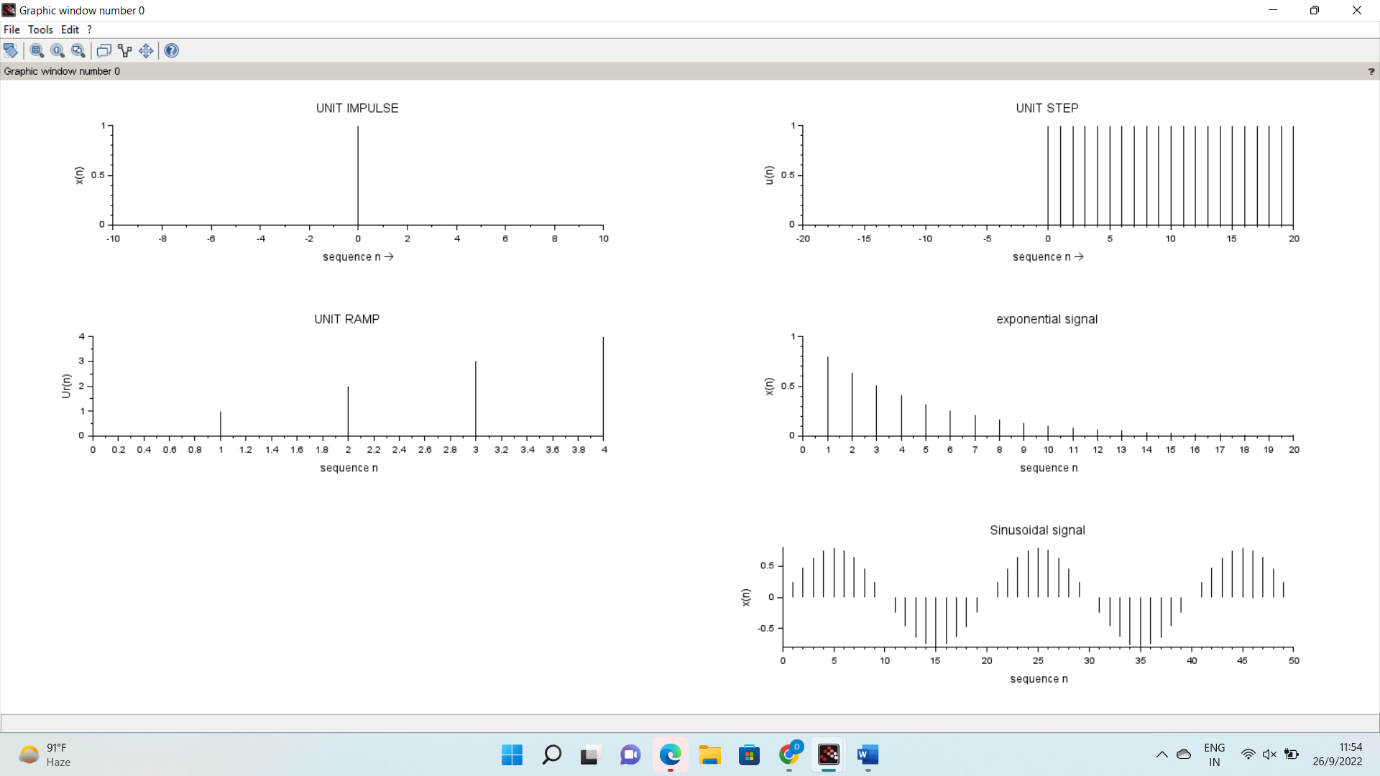
plot2d3(n,x);

xlabel('sequence n ?');

ylabel('x(n)');

title('Sinusoidal signal');

**OUTPUT**



**2. LINEAR CONVOLUTION**

x=input('enter the four point input sequence :');

h=input('enter the four point impulse sequence:');

y=conv(x,h,"full");

disp(y);

subplot(3,1,1);

plot2d3(x);

xlabel('sequence n--> :');

ylabel('x(n)');

title('1.input sequence');

subplot(3,1,2);

plot2d3(h);

xlabel('sequence n--> :');

ylabel('h(n)');

title('2.impulse sequence');

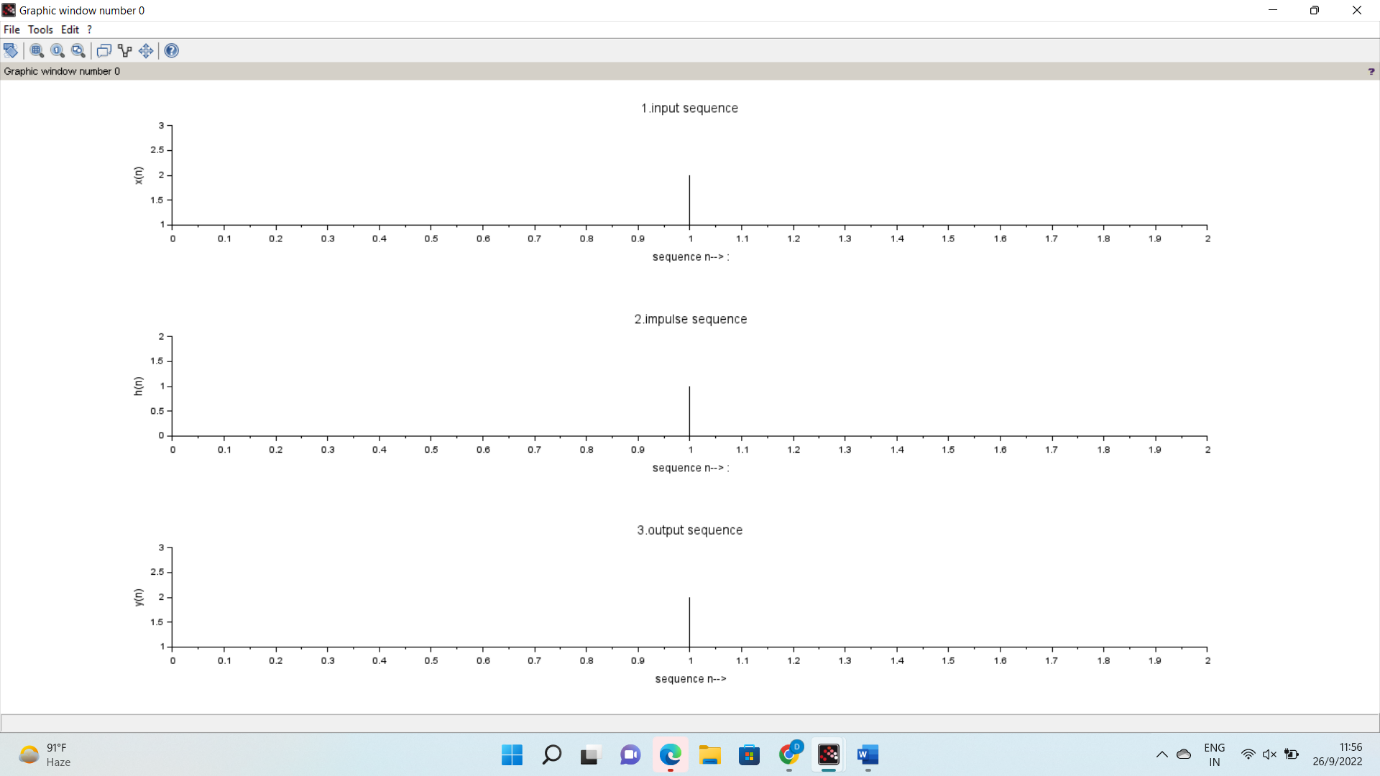
subplot(3,1,3);

plot2d3(y);

xlabel('sequence n-->');

ylabel('y(n)');

title('3.output sequence');

**OUTPUT**

**3. CIRCULAR CONVOLUTION**

// a) CIRCULAR CONVOLUTION

x=input('enter the four point input sequence :');

h=input('enter the four point impulse sequence:');

y=conv(x,h,"same");

disp(y);

subplot(3,1,1);

plot2d3(x);

xlabel('sequence n--> :');

ylabel('x(n)');

title('1.input sequence');

subplot(3,1,2);

plot2d3(h);

xlabel('sequence n--> :');

ylabel('h(n)');

title('2.impulse sequence');

subplot(3,1,3);

plot2d3(y);

xlabel('sequence n-->');

ylabel('y(n)');

title('3.output sequence');

// b) CIRCULAR CONVOLUTION (Using DFT and IDFT – zero padding)

*// circular convolution using DFT and IDFT*

clc;

close;

x=input('enter the input sequence :');

N=input('enter the length of the input sequence :');

h=input('enter the impulse sequence :');

M=input('enter the length of the impulse sequence :');

if N>M then

H=[h,zeros(1,N-M)];

else

H=h;

end

if M>N then

X=[x,zeros(1,M-N)];

else

X=x;

end

x1=fft(X,-1);

disp(x1);

h1=fft(H,-1);

disp(h1);

Y=x1.\*h1;

y=fft(Y,1)

disp(y);

subplot(3,1,1);

plot2d3(x);

xlabel('sequence n--> :');

ylabel('x(n)');

title('1.input sequence');

subplot(3,1,2);

plot2d3(h);

xlabel('sequence n--> :');

ylabel('h(n)');

title('2.impulse sequence');

subplot(3,1,3);

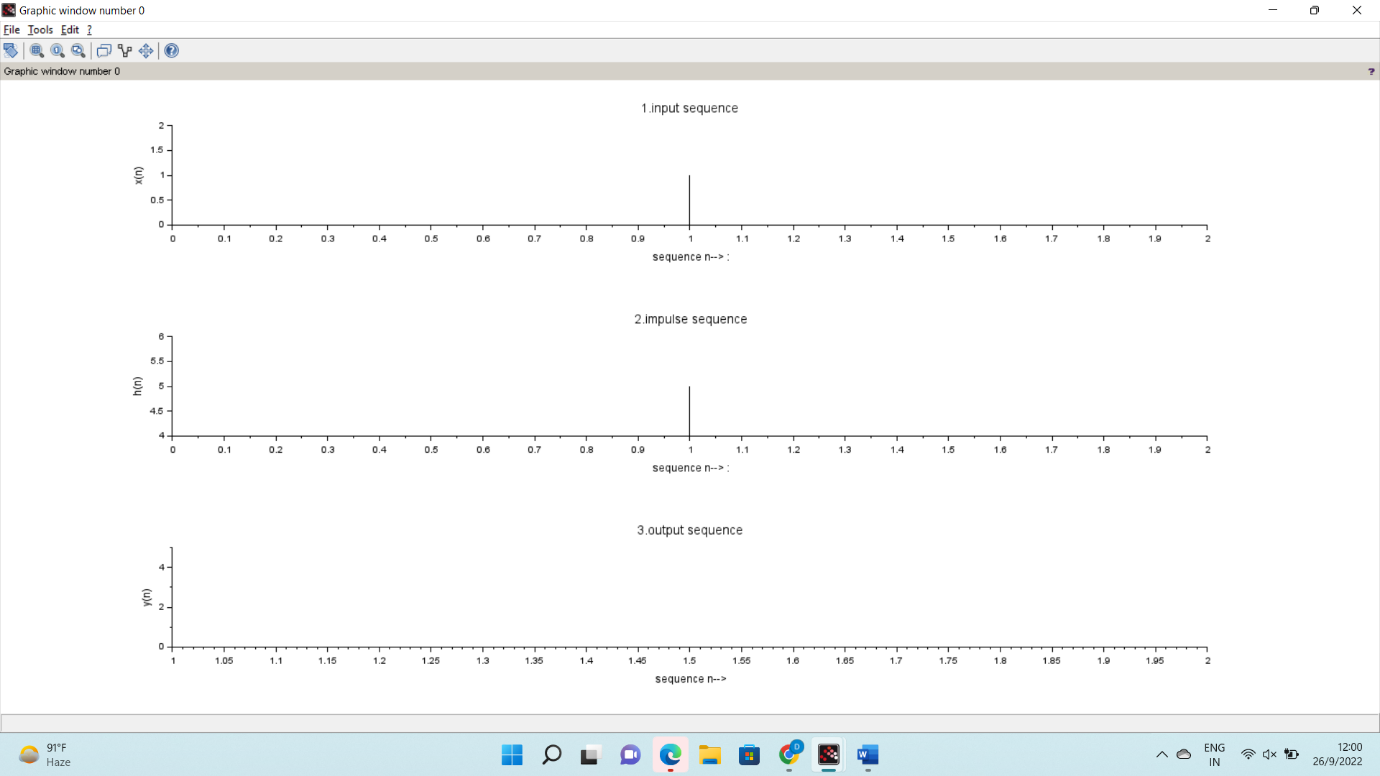
plot2d3(y);

xlabel('sequence n-->');

ylabel('y(n)');

title('3.output sequence');

**OUTPUT**

****

**4. AUTOCORRELATION AND CROSS CORRELATION OF SEQUENCES**

%Auto Correlation%

x= input('Enter any sequence x(n)');

figure(1);

subplot(2,1,1);

stem(x);

xlabel('Time period');

ylabel('Amplitude');

title('Input sequence');

Rxx=xcorr(x)

subplot(2,1,2);

stem(Rxx);

xlabel('Time period');

ylabel('Amplitude');

title('Auto correlation');

%Cross correlation%

x=input('Enter any sequence x(n)');

figure(2);

subplot(3,1,1);

stem(x);

xlabel('Time period');

ylabel('Amplitude');

title('Input sequence');

h=input('Enter any sequence h(n)');

subplot(3,1,2);

stem(h);

xlabel('Time period');

ylabel('Amplitude');

title('Impulse sequence');

Rxy =xcorr(x,h)

subplot(3,1,3);

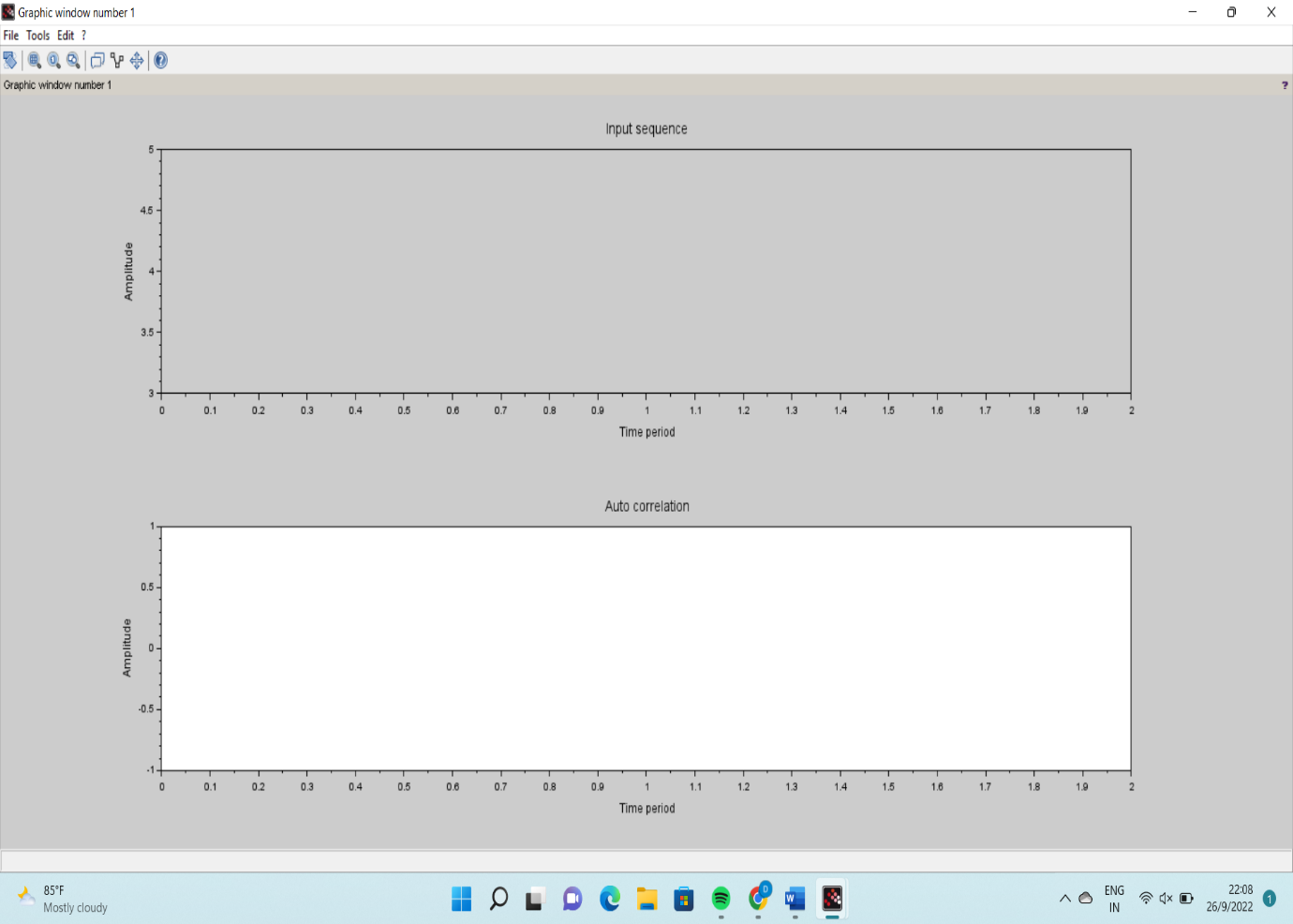
stem(Rxy);

xlabel('Time period');

ylabel('Amplitude');

title('Cross correlation');

**OUTPUT**

****

**5. FREQUENCY ANALYSIS USING DFT**

clc;

xn =[1 2 3 4 4 3 2 1]

n1=0:1:length(xn)-1;

subplot(2,2,1);

plot2d3(n1,xn);

xlabel('Time n');

ylabel('Amplitude');

title('Input Sequence');

Xk = fft(xn);

disp(Xk);

K1=0:1:length(Xk)-1;

magnitude=abs(Xk)

disp(magnitude);

subplot(2,2,2);

plot2d3(K1,magnitude);

xlabel('frequency(Hz)');

ylabel('magnitude(gain)');

title('magnitude spectrum');

angle = atan(imag(Xk),real(Xk))

disp(angle)

subplot(2,2,3);

plot2d3(K1,angle);

xlabel('frequency(Hz)');

ylabel('Phase');

title('Phase spectrum')

y= ifft(Xk)

disp(y);

n2=0:1:length(y)-1;

subplot(2,2,4)

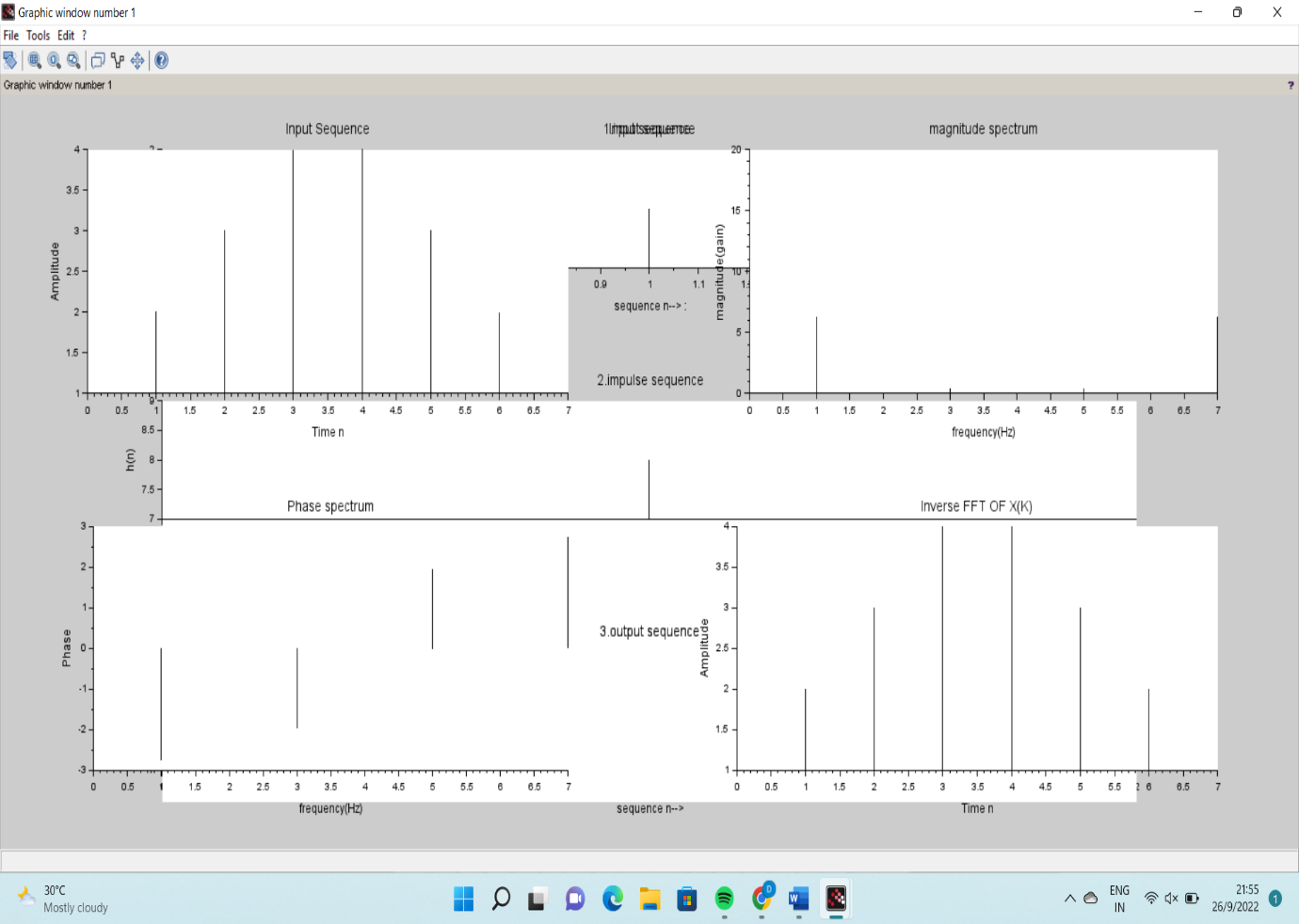
plot2d3(n2,y)

xlabel('Time n');

ylabel('Amplitude');

title('Inverse FFT OF X(K)');

**OUTPUT**

****