EXP NO: 01 **GENERATION OF SIGNALS**

DATE: 17/07/2025

AIM:

To generate various periodic and aperiodic signals such as circuits impulse, step, ramp, , increasing exponential signal, decreasing exponential signal, Sinusoidal signal, cosine signal ,parabolic signal using MATLAB.

APPARATUS REQURIED:

MATLAB software.

THEORY:

IMPLUSE SIGNAL:

A impulse function is a special function that is often used by engineers to model certain events.

$$\delta[n] = 1 \text{ at } n = 0$$

$$\delta[n] = 0 \text{ at } n \neq 0$$

UNIT STEP SIGNAL:

A continuous time unit step signal is denoted by u(t), mathematically it can be expressed as

$$u(t) = 1 \text{ for } t>0$$

$$u(t) = 0 \text{ for } t<0$$

UNIT RAMP SIGNAL:

A continuous time ramp signal is denoted by r(t), mathematically it can be expressed as

$$r(t) = 1 \text{ for } t=1$$

$$r(t) = 0 \text{ for } t<0$$

INCREASING EXPONENTIAL SIGNAL:

The complex exponential is a complex valued signal that simultaneously encapsulate both a cosine signal and a sine signal by posting them on the real and imaginary components of complex signal.

The continuous time complex exponential is defined as:

s (t) =C e^at

DECREASING EXPONENTIAL SIGNAL:

An exponential decay signal is a signal that decreases (decays) exponentially with time. It is a special case of the real exponential signal, where the exponent is negative.

s (t) =C e^-at

SINE WAVE FORM:

A sinusoidal has the same shape as the graph of the sine function used in Trigonometry. A sinusoidal waveform is denoted by

X[n] = A sin (n)

COSINE WAVE FORM:

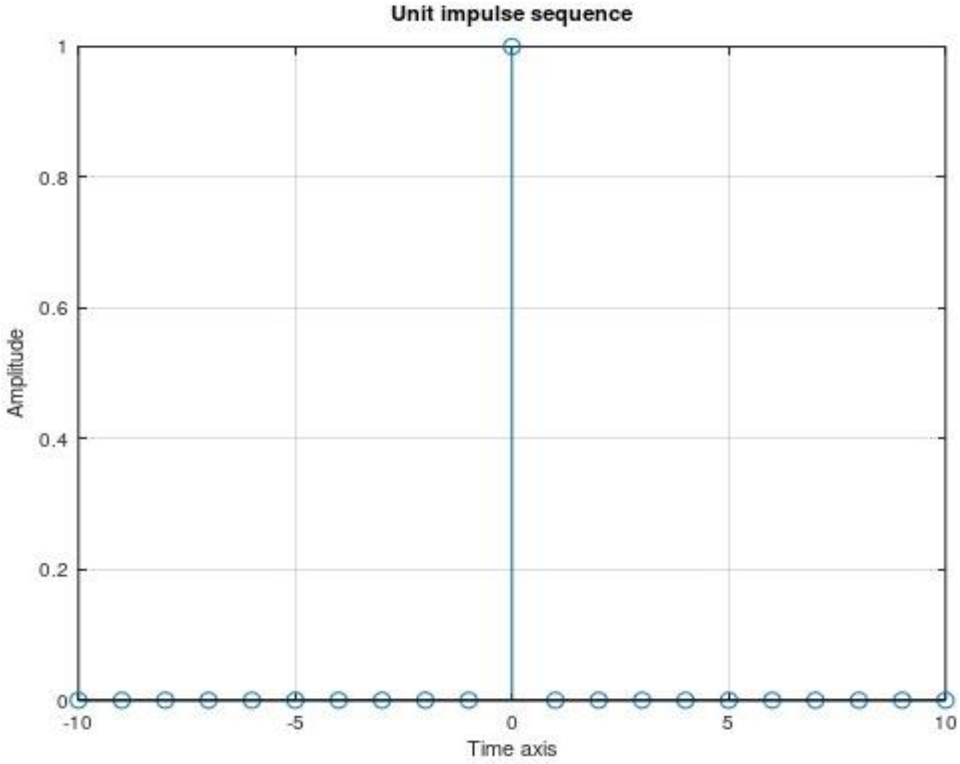
A cosine wave is a signal waveform with a shape identical to that of a sine wave, except each point on the cosine wave occurs exactly ¼ cycles earlier than the corresponding point on the sine wave.

PARABOLIC FUNCTION:

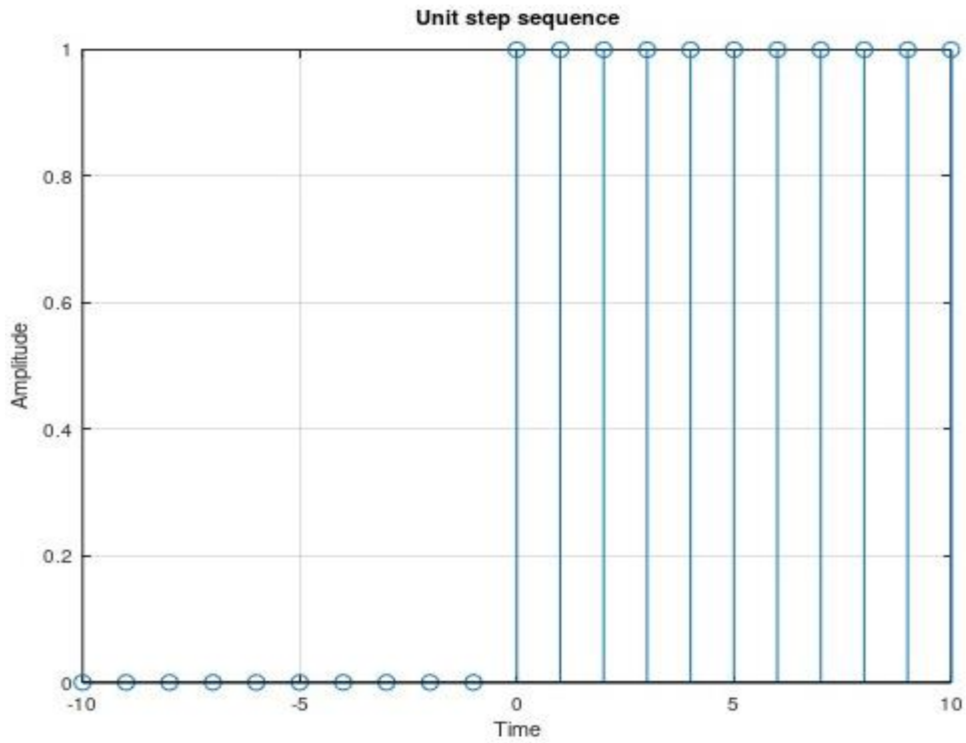
A parabolic function is a second-degree polynomial function that describes a curve opening either upward or downward in the form of a parabola.

X(t) = At^2 + Bt + C

OUTPUT : **FOR UNIT IMPULSE SIGNAL**



FOR UNIT STEP SIGNAL



MATLAB CODE :

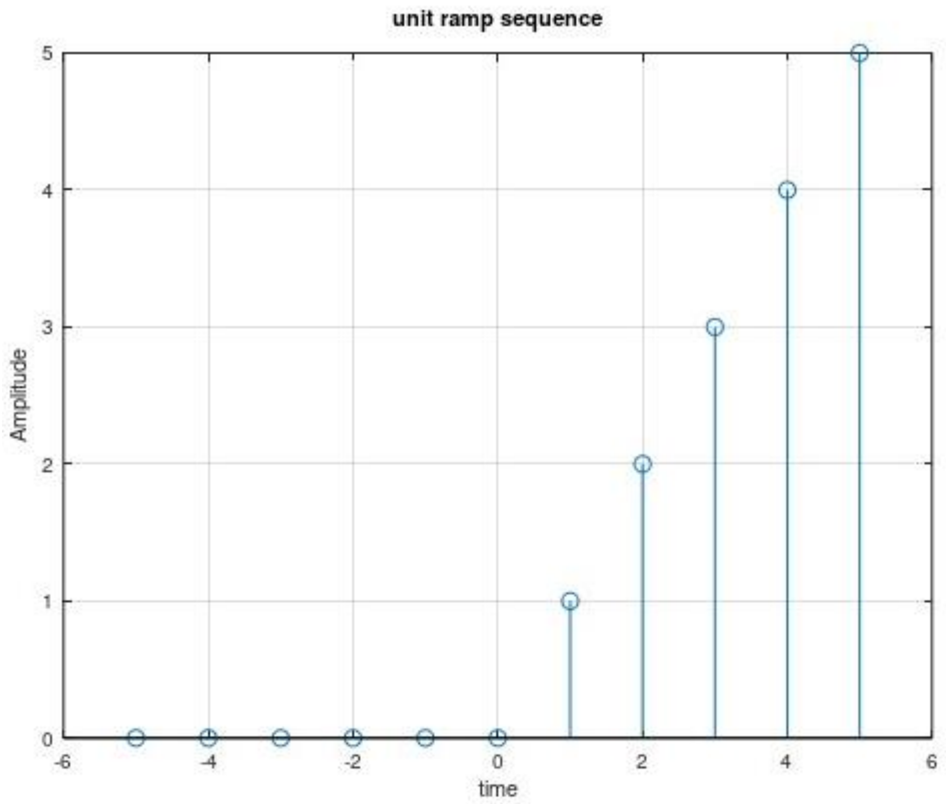
FOR UNIT IMPULSE SIGNAL:

```
clc;
clear all;
close all;
n = -10:1:10;
x = [zeros(1,10), 1, zeros(1,10)];
stem (n,x);
xlabel ('Time axis');
ylabel ('Amplitude');
title ('Unit impulse sequence');
grid on;
```

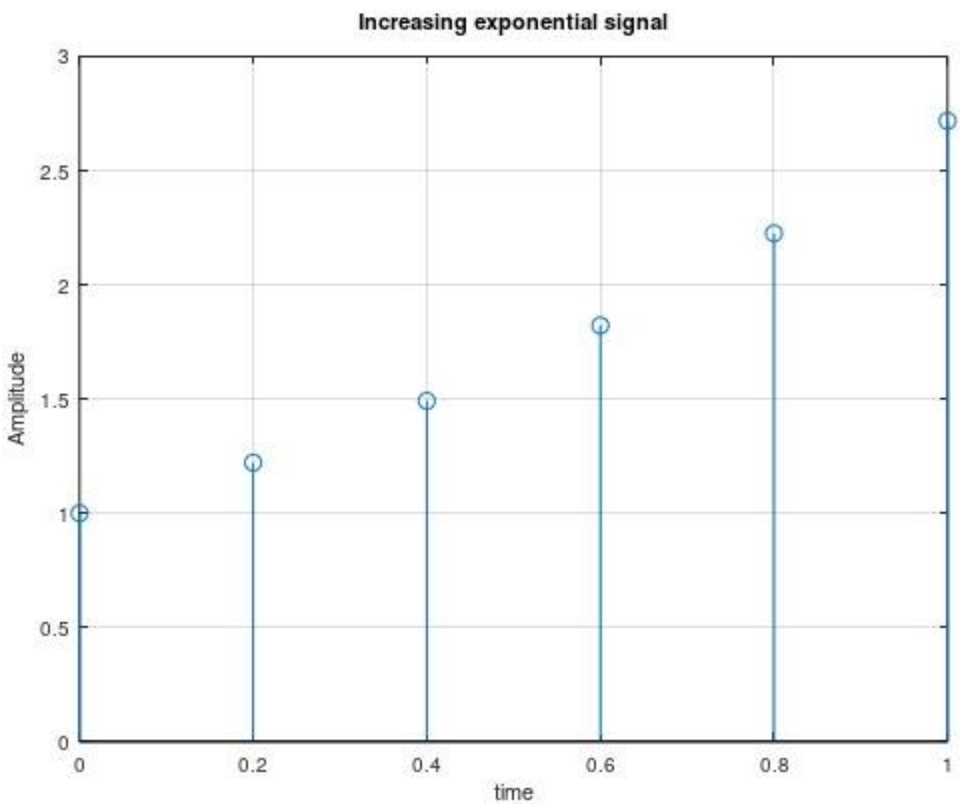
FOR UNIT STEP SIGNAL:

```
clc;
clear all;
close all;
n = -10:1:10;
x= [zeros(1,10), ones(1,11)];
stem (n,x);
xlabel ('Time');
ylabel ('Amplitude');
title ('Unit step sequence');
grid on;
```

OUTPUT : **FOR UNIT RAMP SIGNAL**



FOR INCREASING EXPONENTIAL SIGNAL



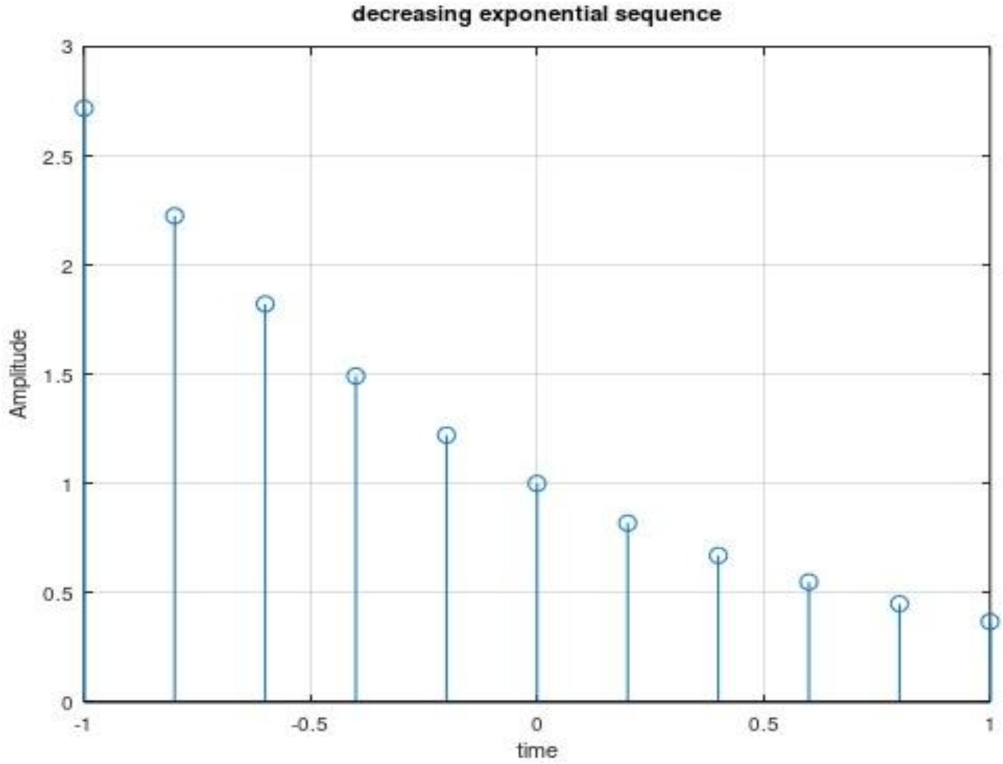
FOR UNIT RAMP SIGANAL:

```
clc;
clear all;
close all;
x = [zeros(1,5) 0:5];
n = -5:5;
stem(n,x);
xlabel ('time');
ylabel ('Amplitude');
title ('unit ramp sequence');
grid on;
```

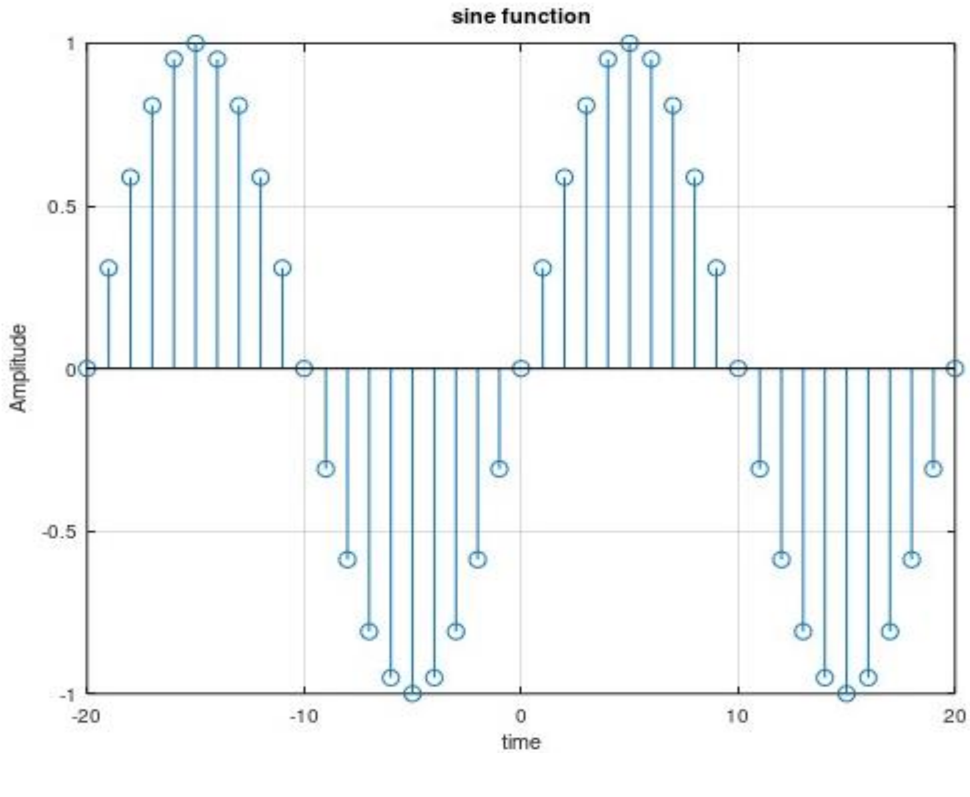
FOR INCREASING EXPONENTIAL SIGNAL:

```
clc;
clear all;
close all;
n = 0:0.2:1;
stem (n,exp(n));
xlabel ('time');
ylabel ('Amplitude');
title ('Increasing exponential signal');
grid on;
```

OUTPUT: FOR DECREASING EXPONENTIAL SIGNAL



FOR SINUSOIDAL SIGNAL



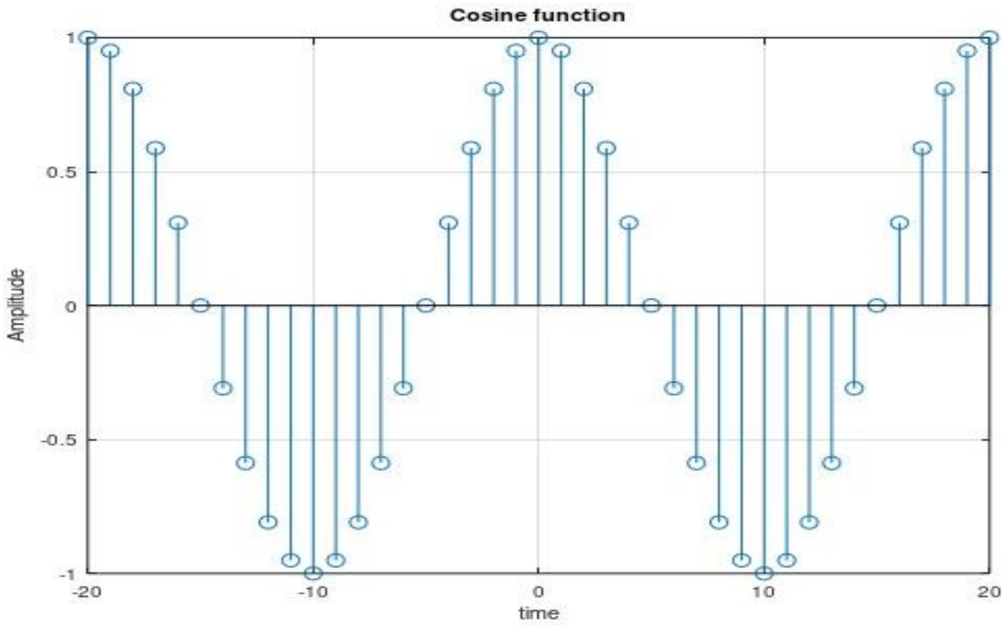
FOR DECREASING EXPONENTIAL SIGNAL:

```
clc;
clear all;
close all;
n = -1:0.2:1;
stem (n,exp(-n));
xlabel('time');
 ylabel ('Amplitude');
title('decreasing exponential sequence');
grind on;
```

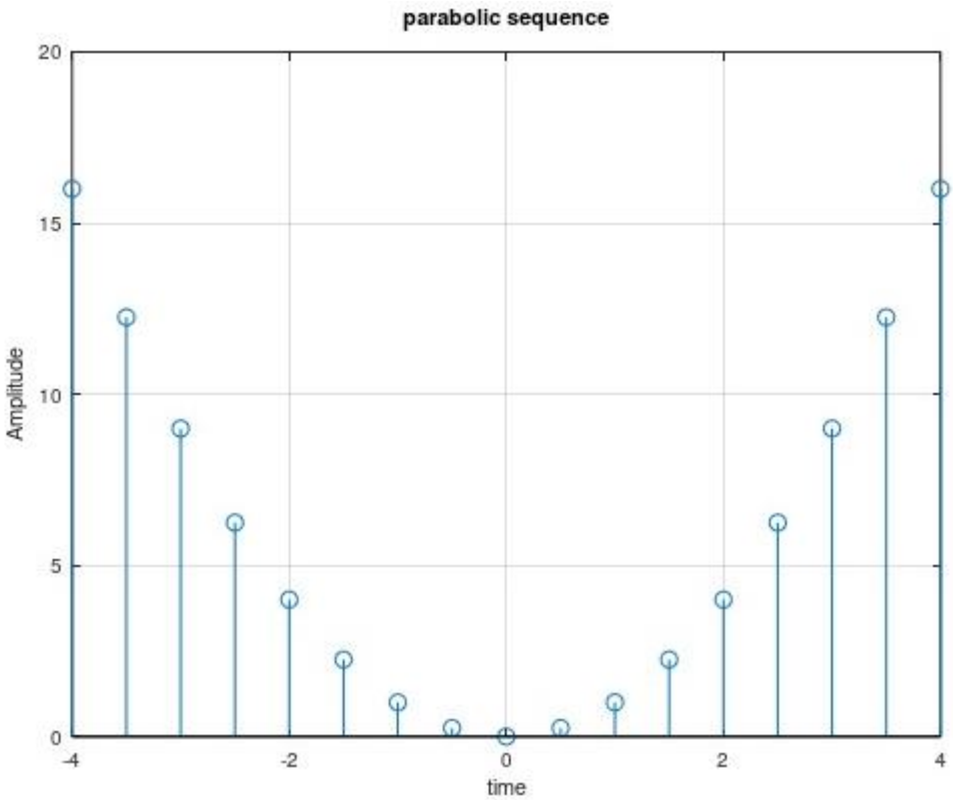
FOR SINUSOIDAL SIGNAL:

```
clear all;
close all;
n = -20:1:20;
f = 0.05
s = 2*pi*f;
x = sin (s*n);
stem (n,x);
xlabel ('time');
ylabel ('Amplitude');
title('sine function');
grid on;
```

OUTPUT: **FOR COSINE SIGNAL**



FOR PARABOLIC SIGNAL



FOR COSINE SIGNAL:

```
clc;
clear all;
close all;
n = -20:1:20;
f = 0.05;
s = 2*pi*f;
x = cos(s*n);
stem (n,x);
xlabel ('time');
ylabel ('Amplitude');
title ('Cosine function');
grid on;
```

PARABOLIC SIGNAL:

```
clc;
clear all;
close all;
n = -4:0.5:4;
x = n.^2;
stem (n,x);
xlabel ('time');
ylabel ('Amplitude');
title ('parabolic sequence');
grid on;
```

INFERENCE:

CONTENTS	MARKS AWAREDED	MARKS OBTAINED
PREPARATION	40	
OBSERVATION	30	
RECORD	20	
VIVA VOCE	10	
TOTAL	100	

RESULT:

MATLAB programs to generate various sequences and waves were written and the results were plotted successfully.

EXP NO: 02 **LINEAR CONVOLUTION**

DATE: 24/07/2025

AIM:

To calculate and plot linear convolution of two signals by using MATLAB software.

APPARATUS REQUIRED:

MATLAB Software.

THEORY:

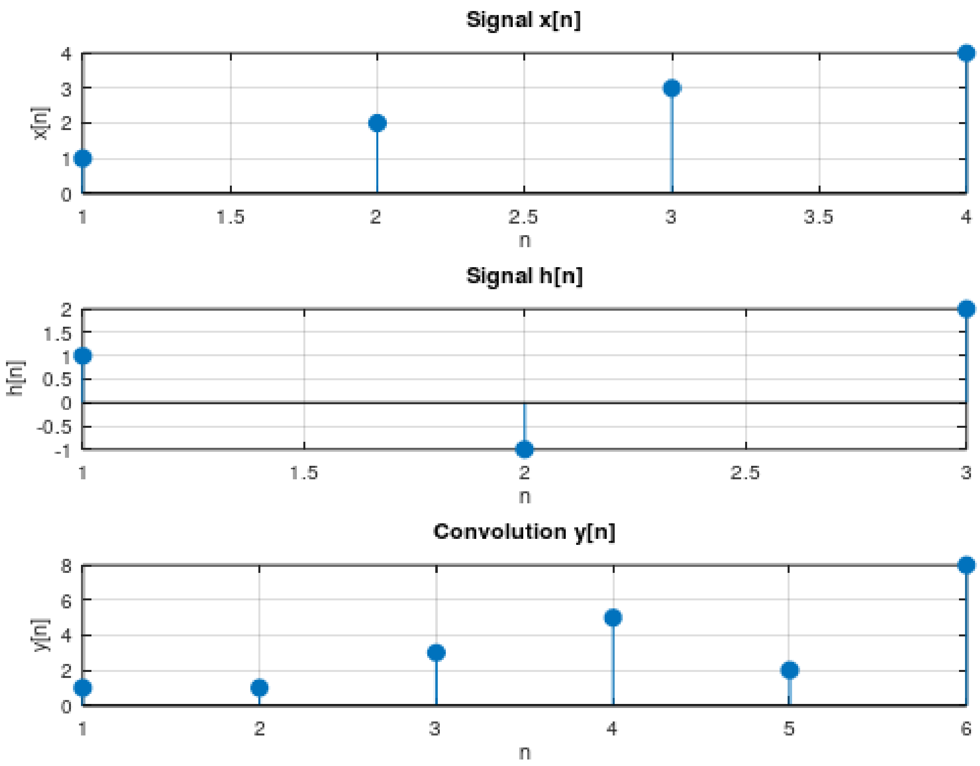
LINEAR CONVOLUTION:

Linear convolution takes two function of an independent variable,which will call time and convolves them using the convolution sum formula. It is the basic operation to calculate the output for any linear time invarient system given its input and its impulse response.

OUTPUT :

The convolution result is:

1   1   3   5   2   8



MAT LAB CODE FOR LINEAR CONVOLUTION:

```
x = [1, 2, 3, 4];
h = [1, -1, 2];
y = conv(x, h);
disp('The convolution result is:');
disp(y);
figure;
subplot(3, 1, 1);
stem(x, 'filled');
title('Signal x[n]');
xlabel('n');
ylabel('x[n]');
grid on;

subplot(3, 1, 2);
stem(h, 'filled');
title('Signal h[n]');
xlabel('n');
ylabel('h[n]');
grid on;

subplot(3, 1, 3);
stem(y, 'filled');
title('Convolution y[n]');
xlabel('n');
ylabel('y[n]');
grid on;
```

INFERENCE:

CONTENTS	MARKS AWAREDED	MARKS OBTAINED
PREPARATION	40	
OBSERVATION	30	
RECORD	20	
VIVA VOCE	10	
TOTAL	100	

RESULT:

Thus, the program for linear convolution was executed and the output is verified using MAT LAB

EXP NO: 03 **CIRCULAR CONVOLUTION**

DATE: 24/07/2025

AIM:

To calculate and plot circular convolution of two signals using MATLAB software.

SOFTWARE REQUIRED:

MATLAB software.

THEORY:

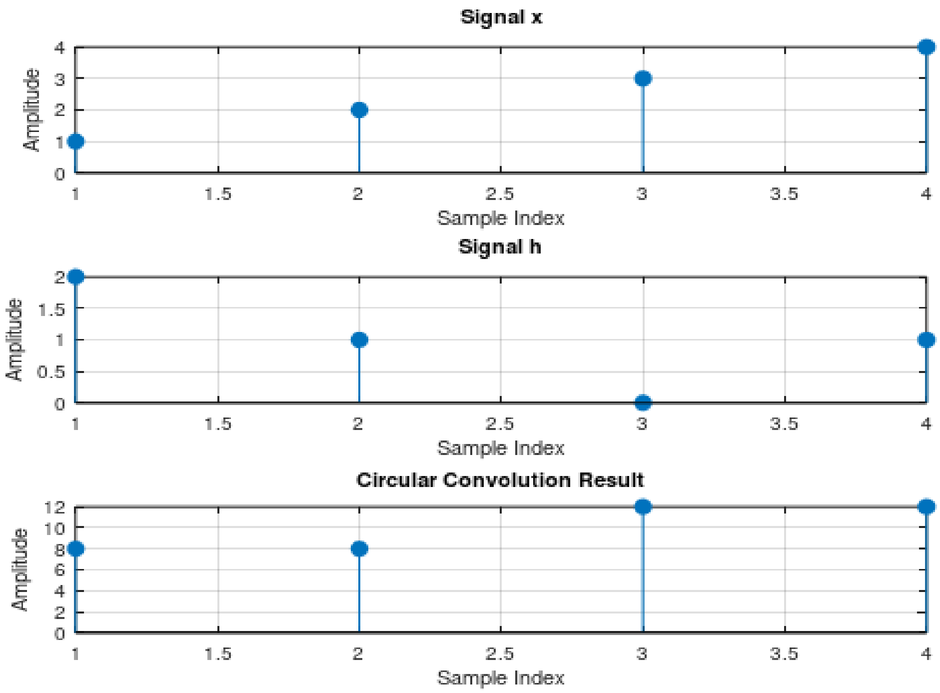
CIRCULAR CONVOLUTION:

The circular convolution, also known as cyclic convolution, of two aperiodic functions (i.e. Schwartz functions) occurs when one of them is convolved in the normal way with a periodic summation of the other function. That situation arises in the context of the circular convolution theorem. The identical operation can also be expressed in terms of the periodic summations of both functions, if the infinite integration interval is reduced to just one period. That situation arises in the context of the discrete-time Fourier transform (DTFT) and is also called periodic convolution

OUTPUT FOR CIRCULAR CONVOLUTION USING MATRIX METHOD:

Circular Convolution Result:

8 8 12 12



MAT LAB CODE FOR CIRCULAR CONVOLUTION:

```
x=[1,2,3,4];
h=[2,1,0,1];
N=length(x);
if length(h)<N
    h=[h,zeros(1,N-length(h))];
elseif length(h) > N
    x=[x,zeros(1,length(h)-N)];
    N=length(x);
end
X=fft(x);
H=fft(h);
Y=X .* H;
y=ifft(Y);
disp('Circular Convolution Result:');
disp(y);
figure;
subplot(3,1,1);
stem(x, 'filled');
title('Signal x');
xlabel('Sample Index');
ylabel('Amplitude');
grid on;

subplot(3,1,2);
stem(h, 'filled');
title('Signal h');
xlabel('Sample Index');
ylabel('Amplitude');
grid on;

subplot(3,1,3);
stem(y, 'filled');
title('Circular Convolution Result');
xlabel('Sample Index');
ylabel('Amplitude');
grid on;
```

INFERENCE:

CONTENTS	MARKS AWAREDED	MARKS OBTAINED
PREPARATION	40	
OBSERVATION	30	
RECORD	20	
VIVA VOCE	10	
TOTAL	100	

RESULT:

Thus, the program for circular convolution was executed and the output is verified using MAT LAB