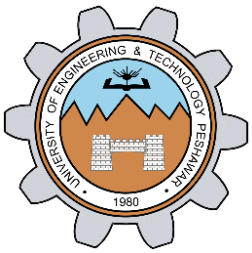
Convolution and LTI systems

LAB # 08



Spring 2023

CSE301L Signals & Systems Lab

Submitted by: **Ali Asghar**

Registration No. : **21PWCSE2059**

Class Section: **C**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Submitted to:

Engr. Sumayyea Salahuddin

Date:

June 1, 2023

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Lab Objective(s):

Objectives of this Lab are;

- Making Signals Causal and Non-Causal
- Convolution
- Properties of Convolution

Task # 01:

Sample the signal given in above example to get its discrete-time counterpart (take 10 samples/sec as sampling rate). Make the resultant signal causal. Display the lollipop plot of each signal.

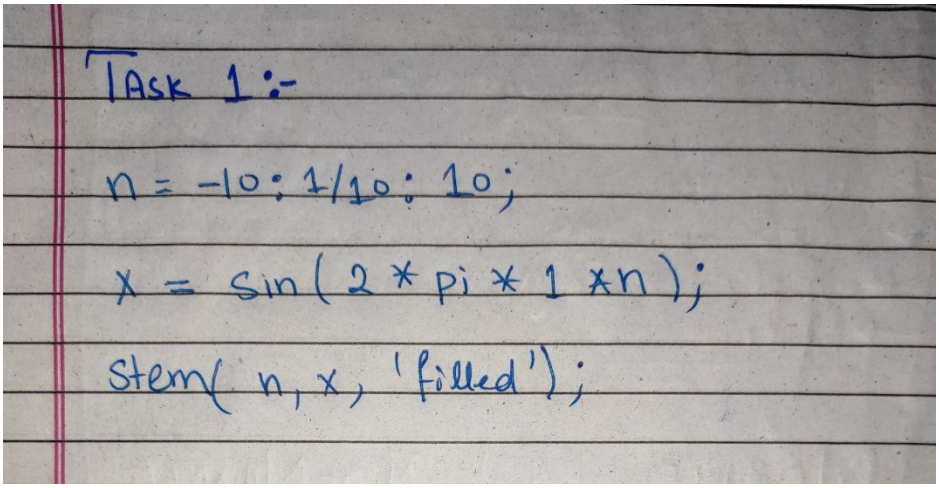
Problem Analysis:

We can sample a signal in MATLAB.

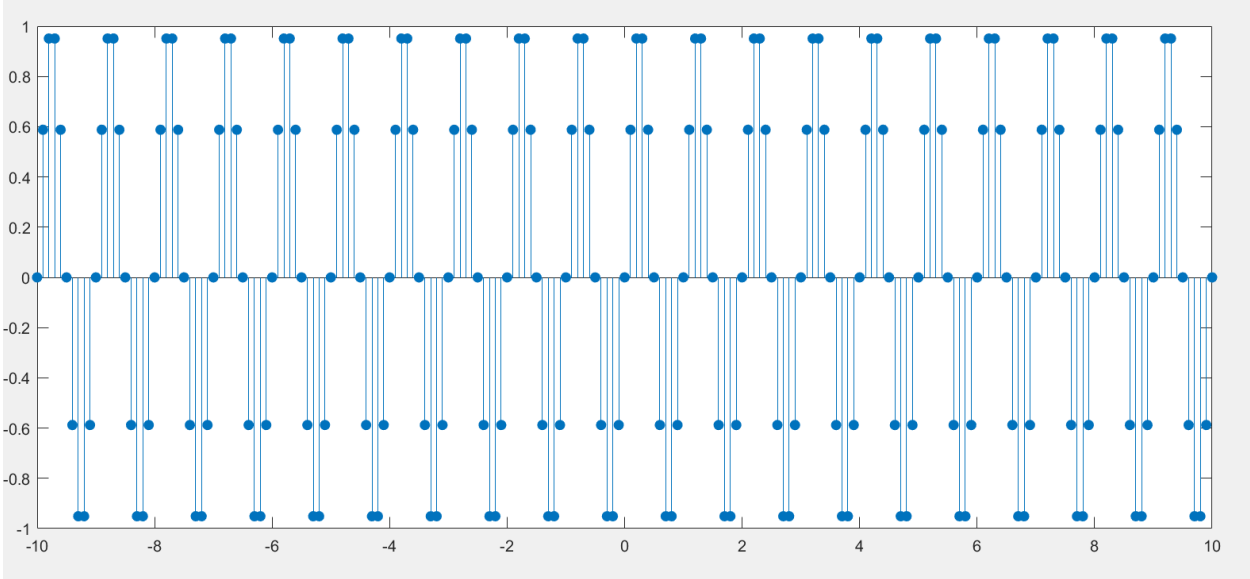
Algorithm:

- Write the code
- Execute it
- Observe the output

Code:



Output / Graphs / Plots / Results:



Discussion and Conclusion:
We can sample a signal in MATLAB.

Task # 02:

A signal is said to be anti-causal if it exists for values of n<0. Make the signal given in above example anti-causal.

Problem Analysis:
We need to make an anti-causal signal in MATLAB.

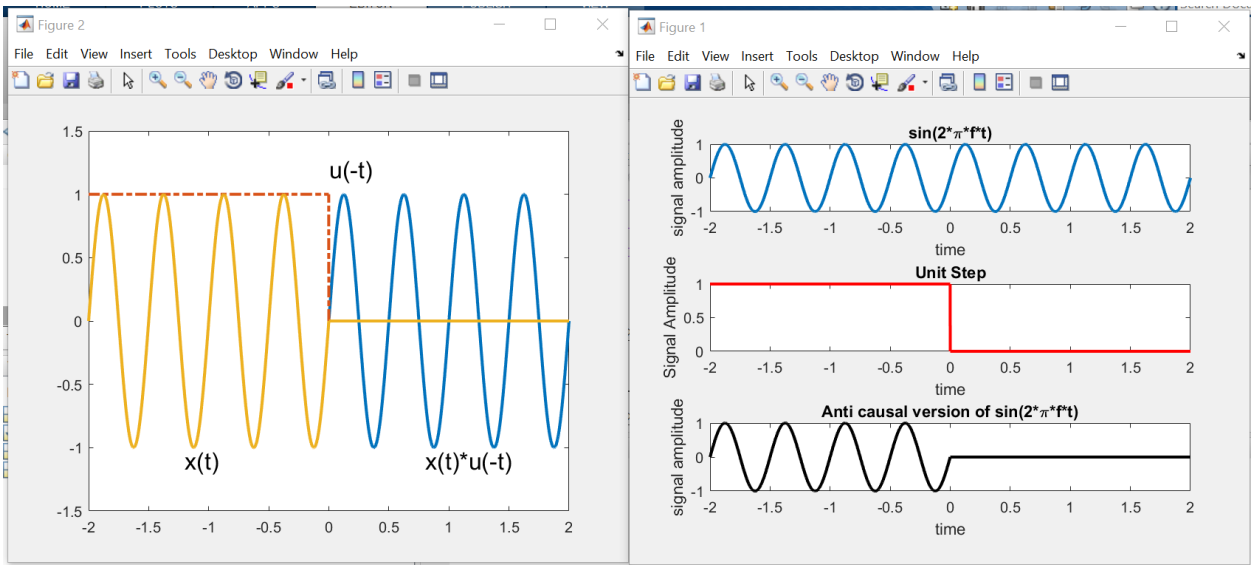
Algorithm:

- Write the code
- Execute it
- Observe the output

Code:

```
Task 2 :-

t = -2 : 1/1000 : 2;
x1 = sin(2*pi*2*t);

subplot(3, 1, 1);
plot(t, x1, 'LineWidth', 2);
xlabel('time');   ylabel('signal amplitude');
title('sin(2*\pi*f*t)');
u = (t<0);
x2 = x1.*u;

subplot(3, 1, 2);
plot(t, x2, 'K', 'LineWidth', 2);
xlabel('time');   ylabel('signal amplitude');
title('Anti causal version of sin(2*\pi*f*t)');
figure;
plot(t, x1, t, u, '-.', t, x2, 'LineWidth', 2);
text(0, 1.2, 'u(-t)', 'FontSize', 16);
text(-1.2, -1.1, 'x(t)', 'FontSize', 16);
text(0.8, -1.1, 'x(t)*u(-t)', 'FontSize', 16);
axis([-2 2 -1.5 1.5]);
```

Output / Graphs / Plots / Results:



Discussion and Conclusion:
We can make an anti-causal signal in MATLAB by using Unit Step Function.

Task # 03:

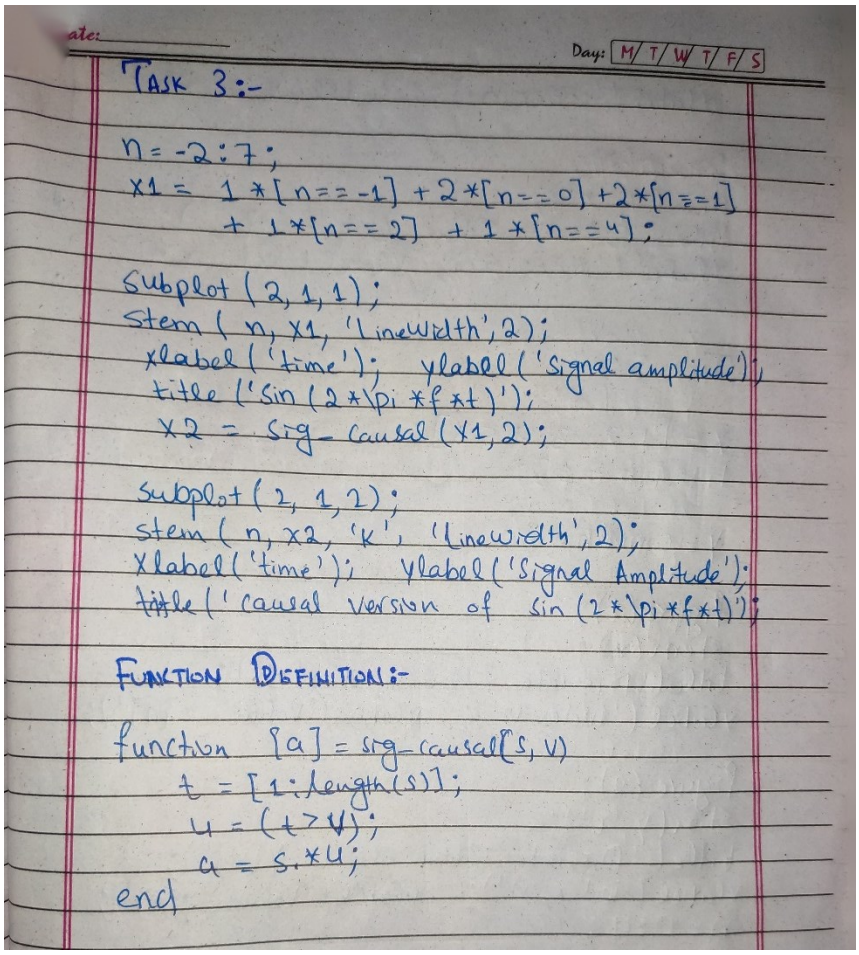
Create a function by name of sig_causal in matlab that has two input arguments: (i) a discrete-time
signal, and (ii) a position vector. The function should make the given signal causal and return the resultant signal to the calling program.

Problem Analysis:
We can perform different operations of signals in MATLAB. Here we make a causal signal than change its causality.

Algorithm:

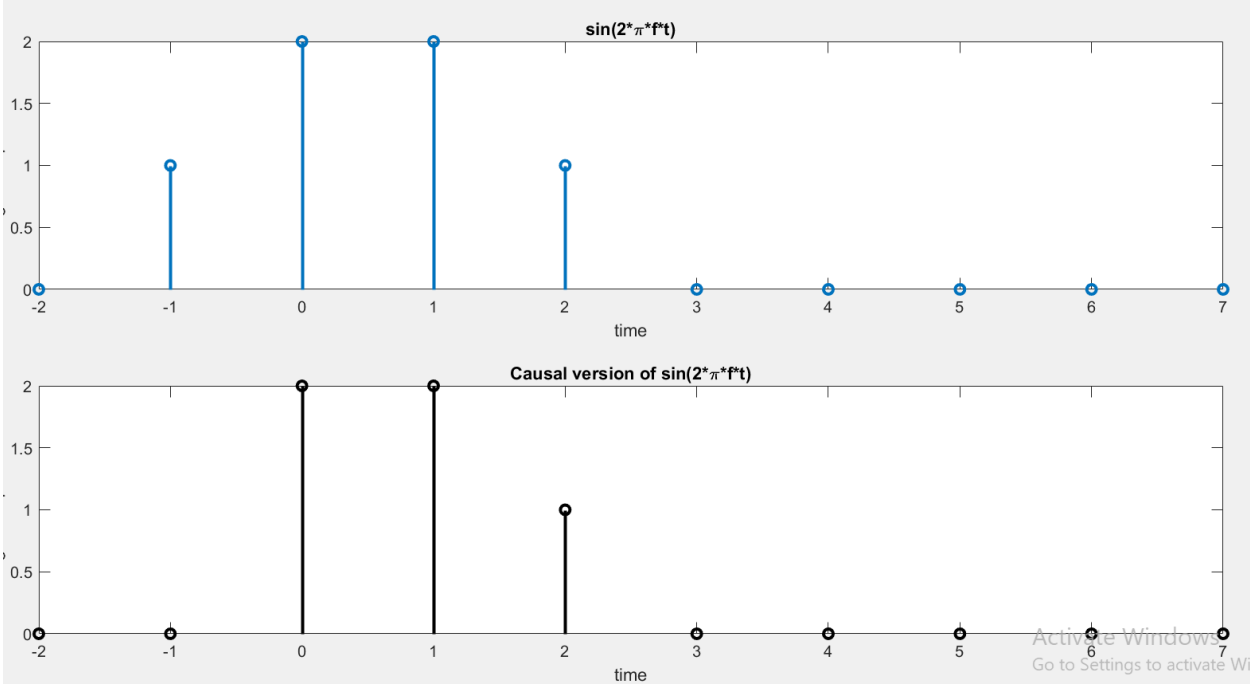
- Write the code
- Execute it
- Observe the output

Code:



```
Date:                                   Day: M T W T F S

TASK 3:-

n = -2:7;
x1 = 1*[n==-1] + 2*[n==0] + 2*[n==1]
     + 1*[n==2] + 1*[n==4];

subplot(2,1,1);
stem(n, x1, 'LineWidth', 2);
xlabel('time'); ylabel('Signal amplitude');
title('Sin(2*\pi*f*t)');
x2 = sig_causal(x1, 2);

subplot(2,1,2);
stem(n, x2, 'k', 'LineWidth', 2);
xlabel('time'); ylabel('Signal Amplitude');
title('causal version of sin(2*\pi*f*t)');

FUNCTION DEFINITION:-

function [a] = sig_causal(s, v)
    t = [1:length(s)];
    u = (t>v);
    a = s.*u;
end
```

Output / Graphs / Plots / Results:



Discussion and Conclusion:
We can perform different operations of signals in MATLAB.

Task # 04:

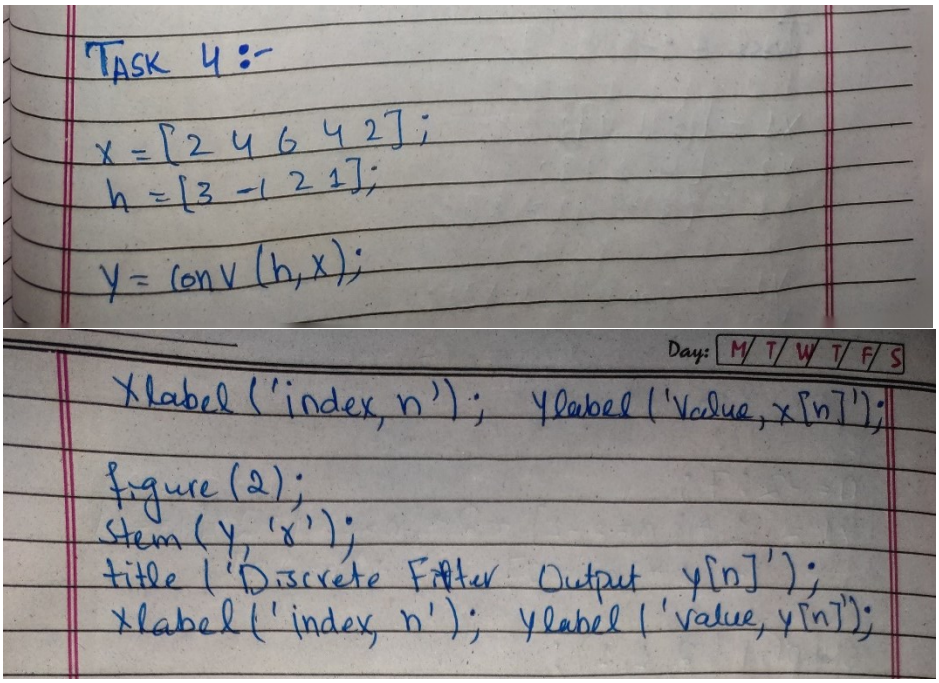
Convolve the following signals:
x =[2 4 6 4 2];
h =[3 -1 2 1];
Plot the input signal as well as the output signal.

Problem Analysis:
We can perform different operations of signals in MATLAB. We perform Convolution in MATLAB.

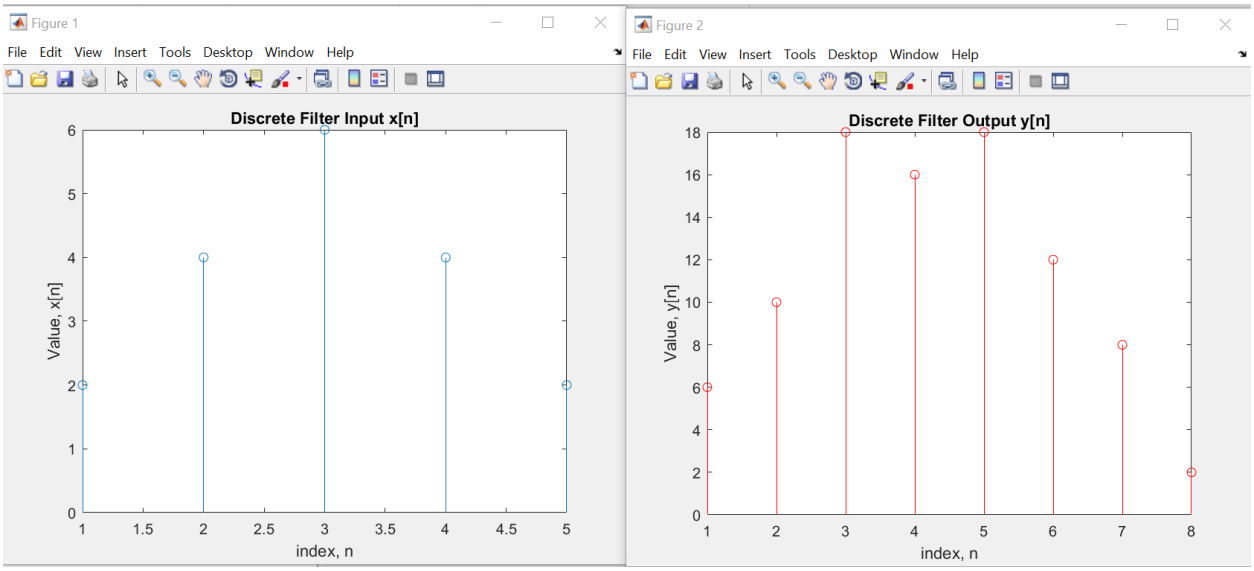
Algorithm:

- Write the code
- Execute it
- Observe the output

Code:



Output / Graphs / Plots / Results:



Discussion and Conclusion:

We can perform different operations of signals in MATLAB.

Task # 05:

Convolution is associative. Given the three signal x1[n], x2[n], and x3[n]

As: x1[n]= [3 1 1]

x2[n]= [4 2 1]
x3[n]=[3 2 1 2 3]

Show that (x1[n] * x2[n]) * x3[n] = x1[n] * (x2[n] * x3[n]).

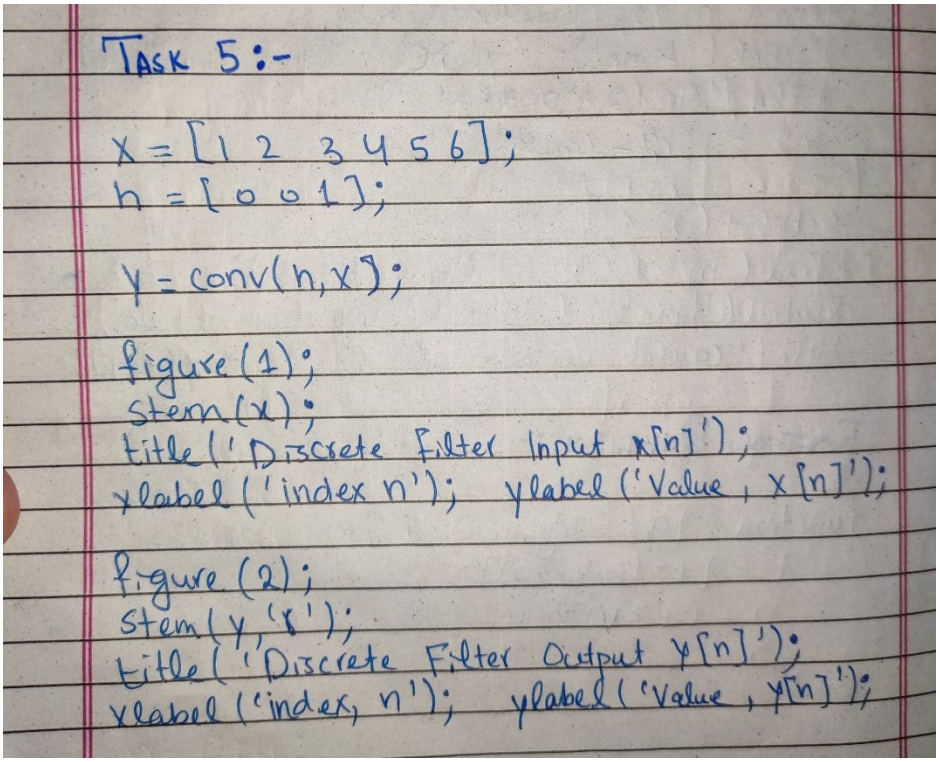
Problem Analysis:

We can perform different operations of signals in MATLAB. We perform Convolution in MATLAB.

Algorithm:

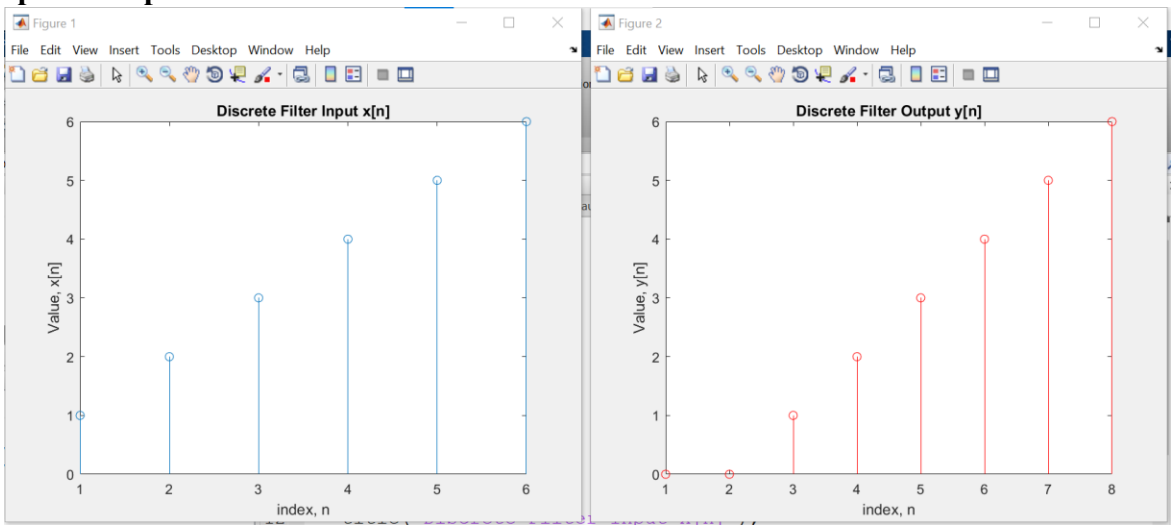
- Write the code
- Execute it
- Observe the output

Code:



```
TASK 5:-

x = [1 2 3 4 5 6];
h = [0 0 1];

y = conv(h,x);

figure(1);
stem(x);
title('Discrete Filter Input x[n]');
xlabel('index n'); ylabel('Value, x[n]');

figure(2);
stem(y,'r');
title('Discrete Filter Output y[n]');
xlabel('index, n'); ylabel('Value, y[n]');
```

Output / Graphs / Plots / Results:



Discussion and Conclusion:
We can perform different operations of signals in MATLAB.

Task # 06:

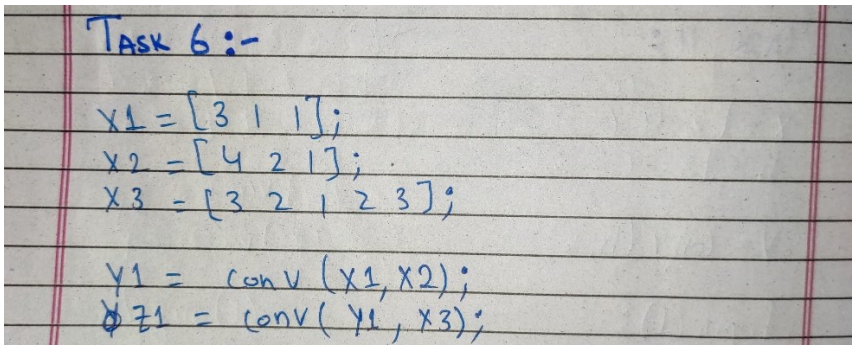
Convolution is associative. Given the three signal x1[n], x2[n], and x3[n] as:
x1[n]= [3 1 1]
x2[n]= [4 2 1]
x3[n]=[3 2 1 2 3]
Show that (x1[n] * x2[n]) * x3[n] = x1[n] * (x2[n] * x3[n]).

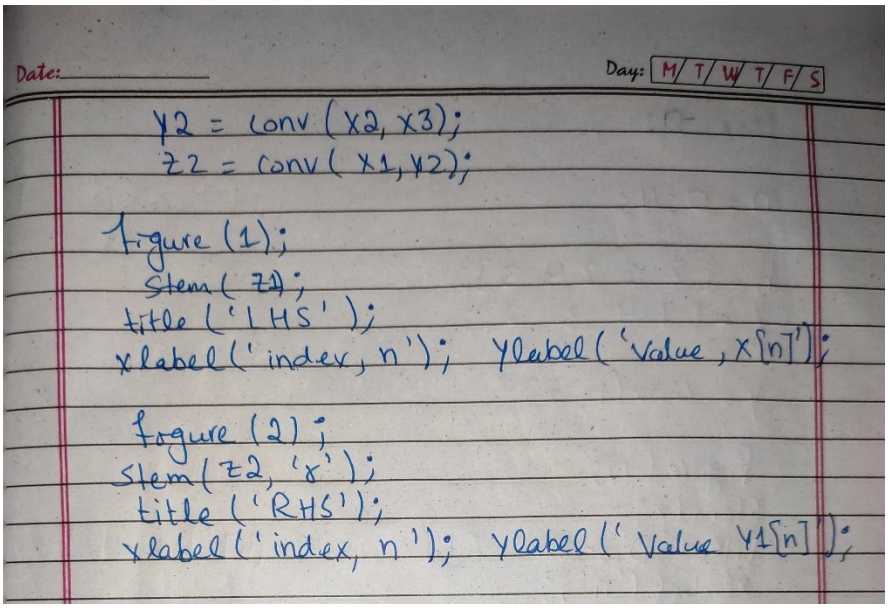
Problem Analysis:
We can perform different operations of signals in MATLAB. We perform Convolution in MATLAB.

Algorithm:

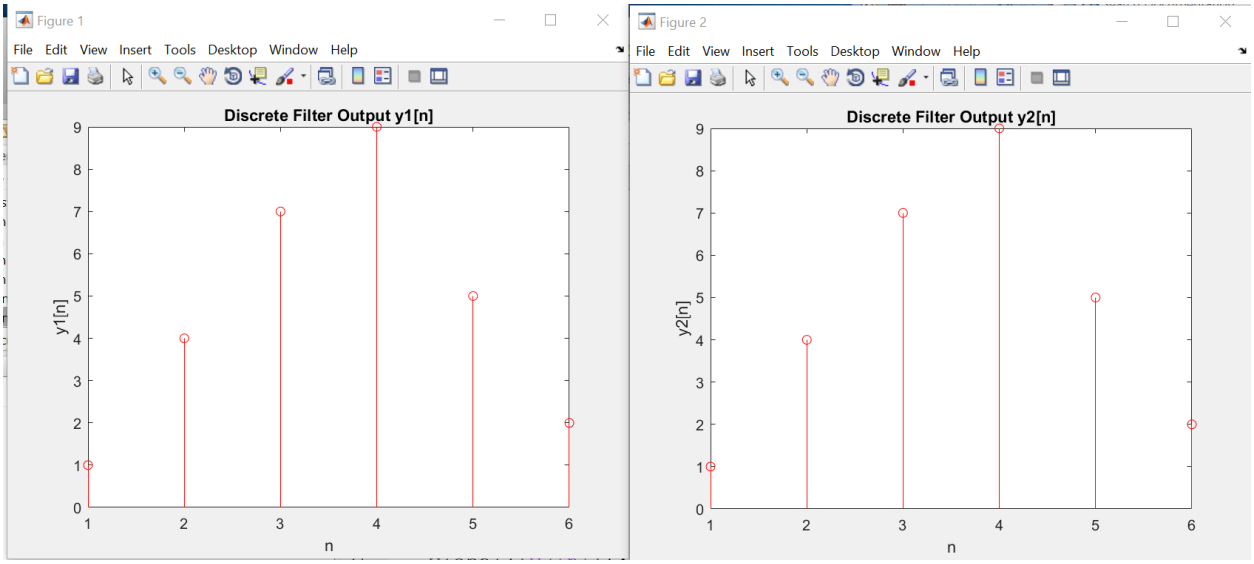
- Write the code
- Execute it
- Observe the output

Code:





Output / Graphs / Plots / Results:



Discussion and Conclusion:
We can perform different operations of signals in MATLAB.

Task # 07:

Convolution is commutative. Given x[n] and h[n] as: X[n]=[1 3 2 1] H[n]=[1 1 2]
Show that x[n] * h[n] = h[n] * x[n].

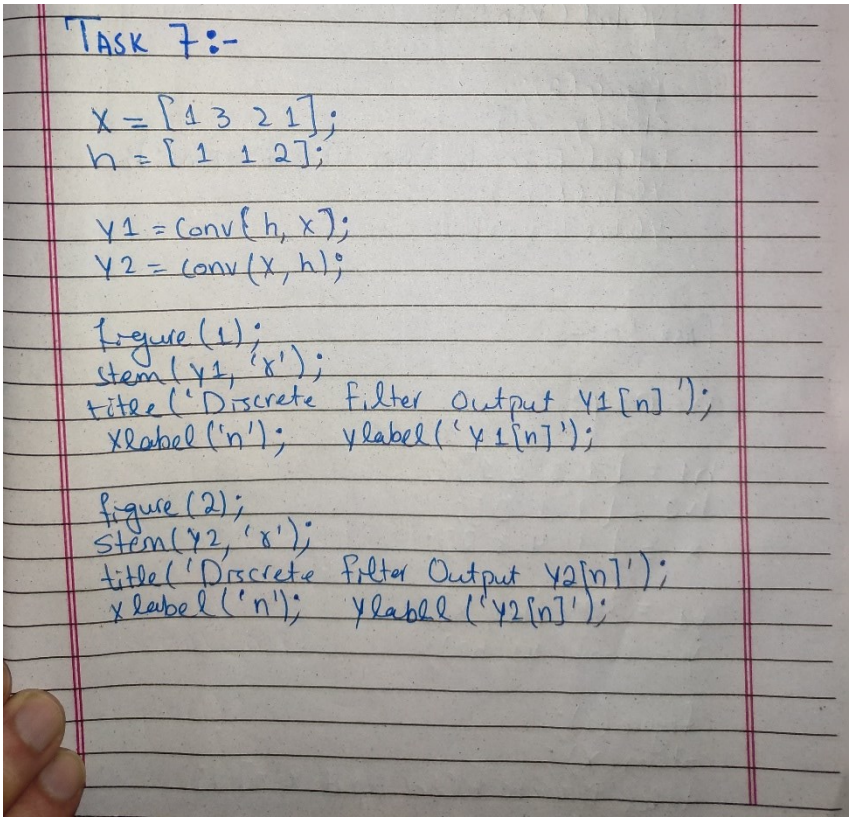
Problem Analysis:

We need to show that Convolution is commutative.

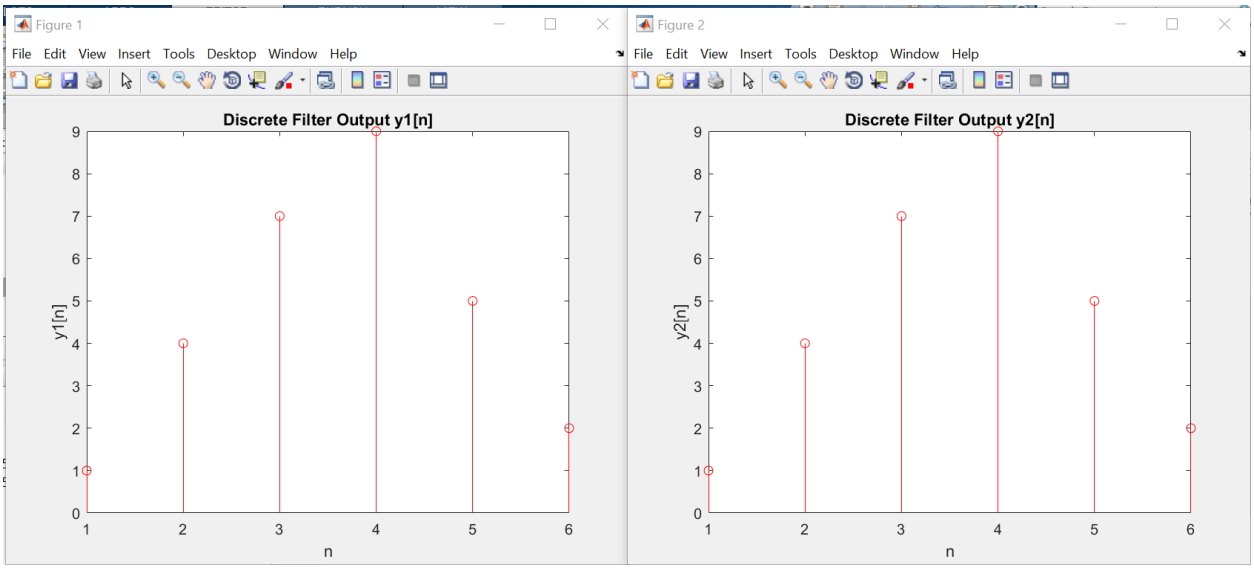
Algorithm:

- Write the code
- Execute it
- Observe the output

Code:



Output / Graphs / Plots / Results:



Discussion and Conclusion:
We can perform different operations of signals in MATLAB.

Task # 08:

Determine h[n] for the system:

$$y[n] = \sum_{k=0}^{10} kx[n - k]$$

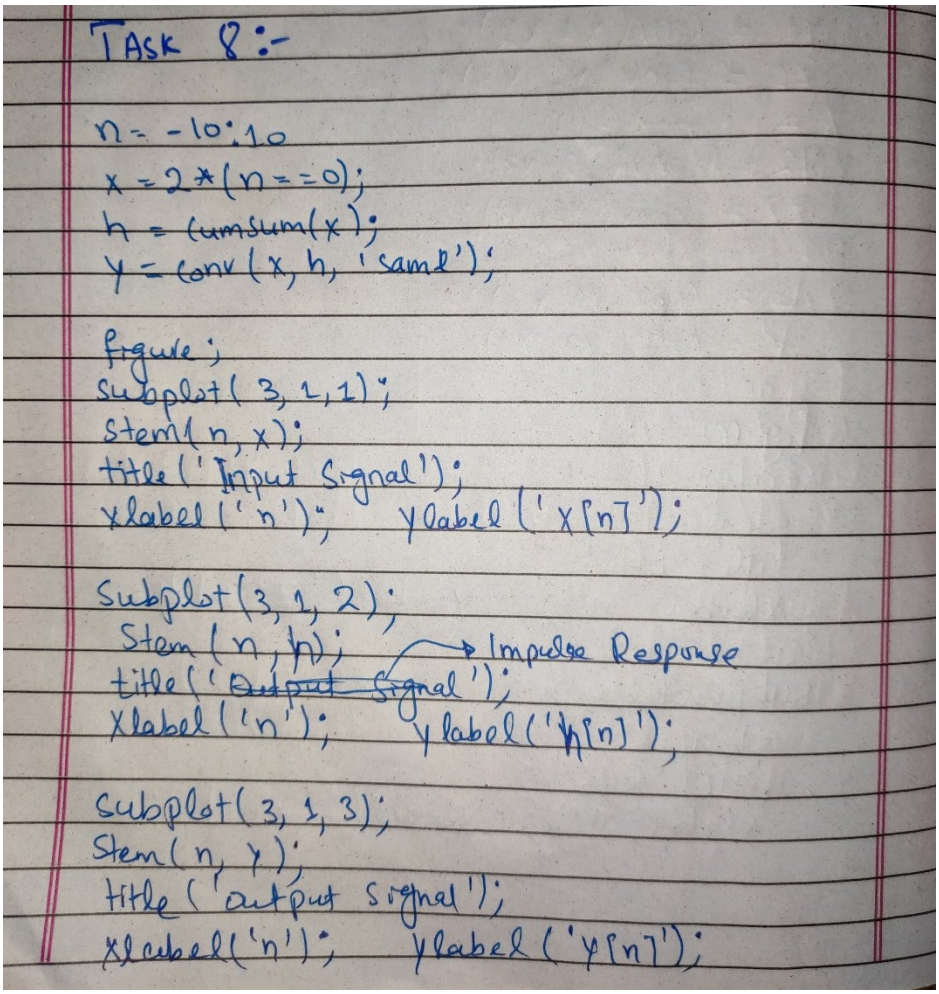
When x[n] = 2δ[n]. Plot the input signal, impulse response, and output signal.

Problem Analysis:
We can perform different operations of signals in MATLAB. We perform Convolution in MATLAB.

Algorithm:

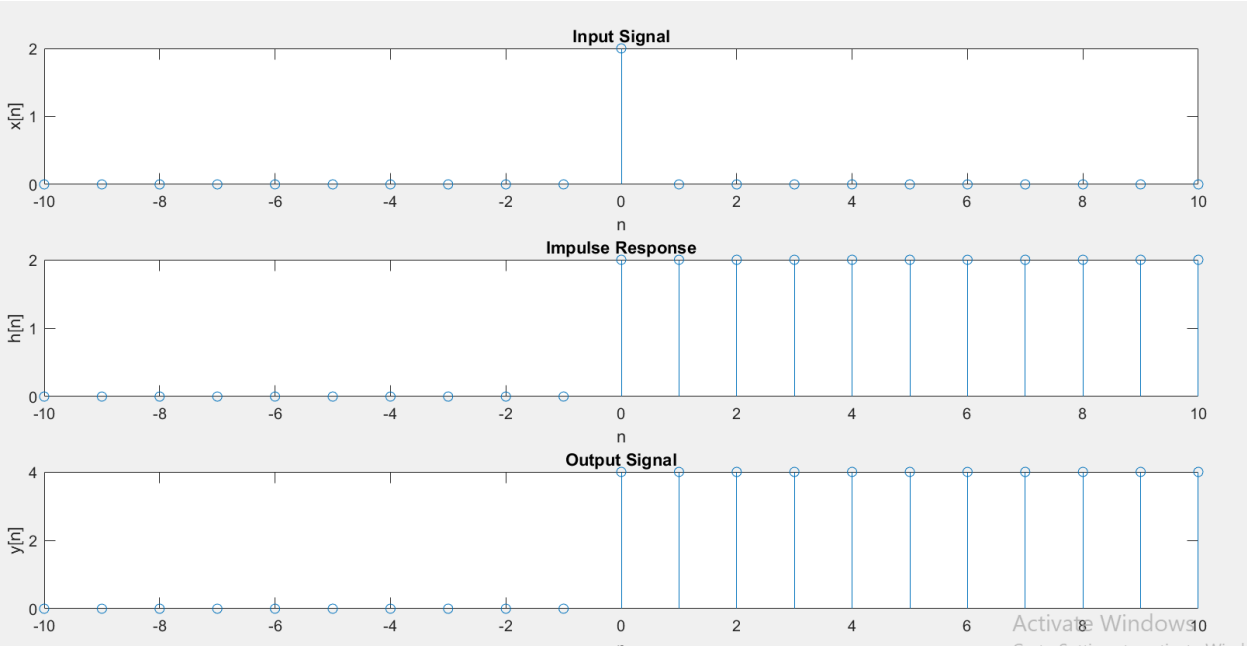
- Write the code
- Execute it
- Observe the output

Code:



```
TASK 8:-

n = -10:10
x = 2*(n==0);
h = cumsum(x);
y = conv(x, h, 'same');

figure;
subplot(3,1,1);
stem(n, x);
title('Input Signal');
xlabel('n'); ylabel('x[n]');

subplot(3,1,2);
stem(n, h);                    -> Impulse Response
title('Output Signal');
xlabel('n'); ylabel('h[n]');

subplot(3,1,3);
stem(n, y);
title('output signal');
xlabel('n'); ylabel('y[n]');
```

Output / Graphs / Plots / Results:



Discussion and Conclusion:
We can perform different operations of signals in MATLAB.

Task # 09:

Given the impulse response of the systems as: h[n]= 2δ[n] + δ[n-1]+ 2δ[n-2]+ 4δ[n-3]+ 3δ[n-4] If the input x[n] = δ[n]+ 4δ[n-1] +3δ[n-2] + 2δ[n-3] is applied to the system, determine the output of the system.

Problem Analysis:
We can perform different operations of signals in MATLAB. We perform Convolution in MATLAB.

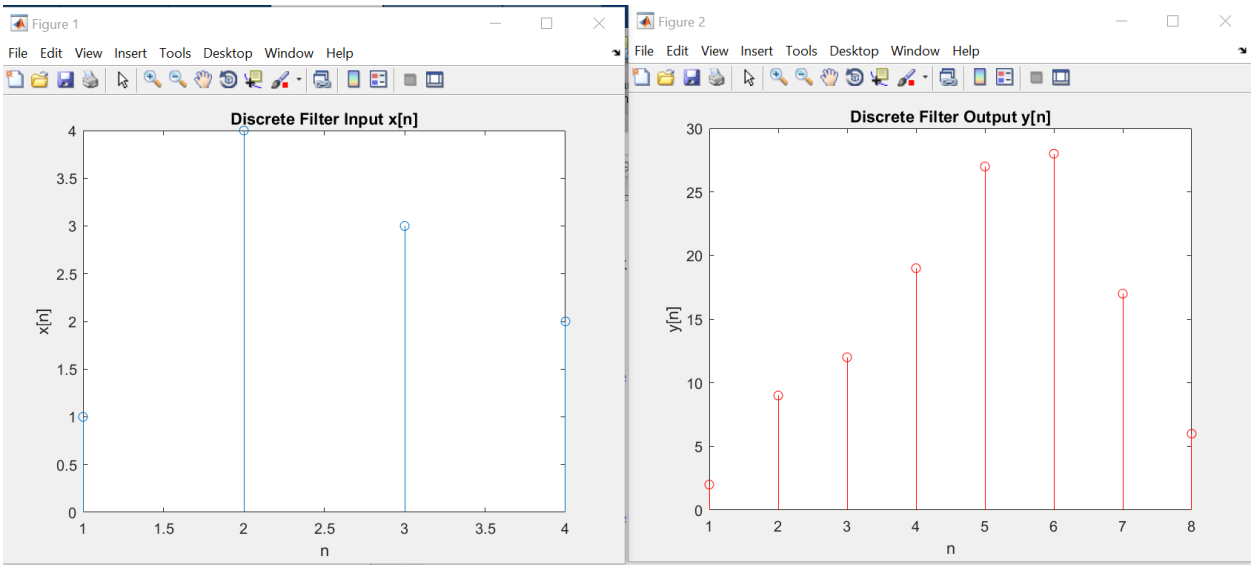
Algorithm:

- Write the code
- Execute it
- Observe the output

Code:

```
TASK 9:-

n = 0:4;

h = [2 1 2 4 3];
x = [1 4 3 2];

y = conv(h, x);

figure(1);
stem(x);
title('Discrete Filter Input x[n]');
xlabel('n');
ylabel('x[n]');

figure(2);
stem(y, 'r');
title('Discrete Filter Output y[n]');
xlabel('n');
ylabel('y[n]');
```

Output / Graphs / Plots / Results:

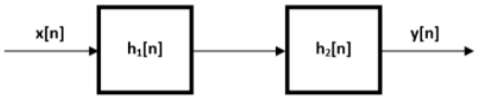


Discussion and Conclusion:

We can perform different operations of signals in MATLAB.

Task # 10:

Two systems are connected in cascade:



h1[n]=[1 3 2 1]
h2[n]=[1 1 2]
If the input x[n] = δ[n]+ 4δ[n-1] +3δ[n-2] +2δ[n-3] is applied, determine the output.

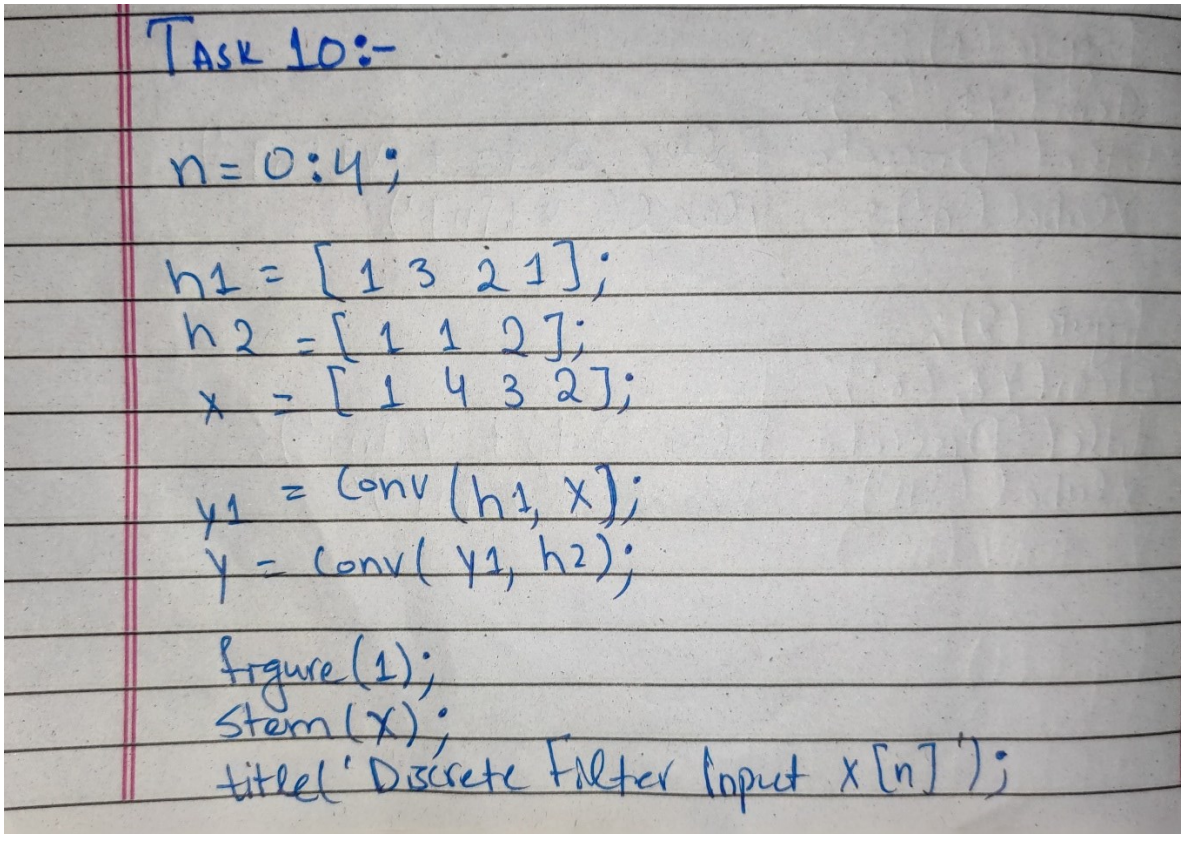
Problem Analysis:

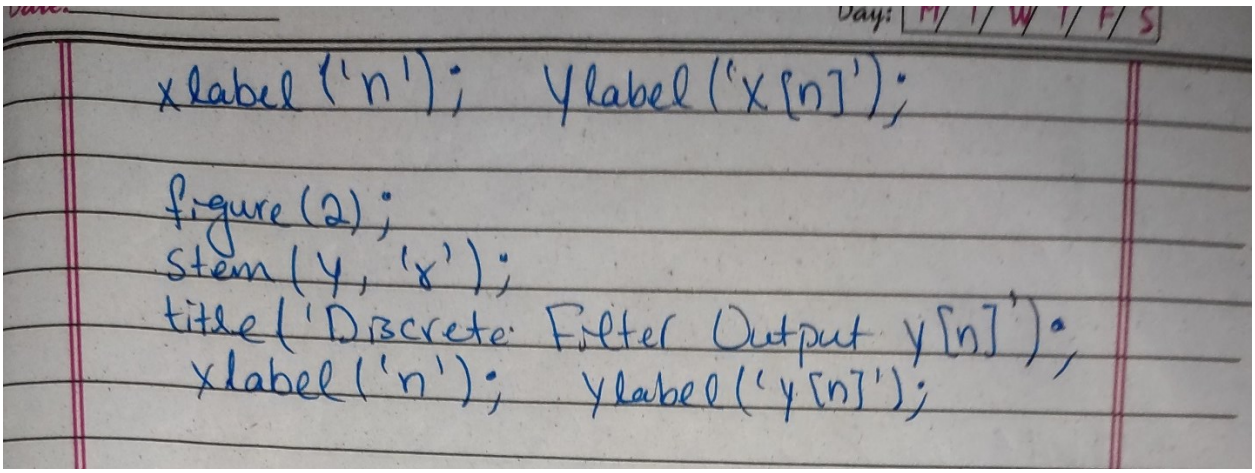
We can perform different operations of signals in MATLAB. We perform Convolution in MATLAB.

Algorithm:

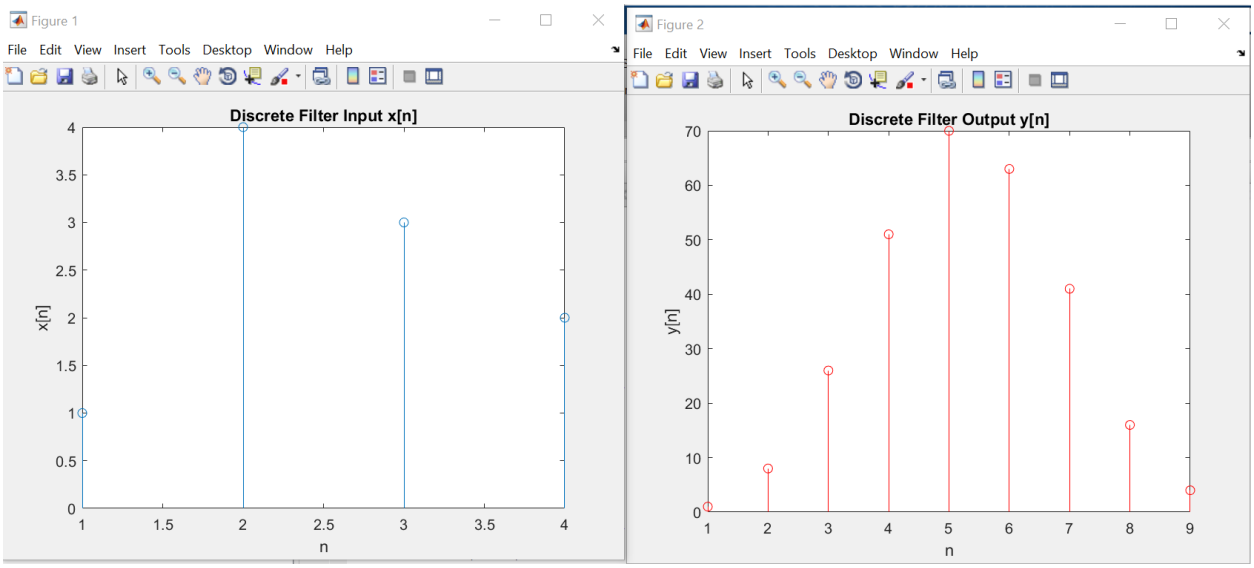
- Write the code
- Execute it
- Observe the output

Code:





Output / Graphs / Plots / Results:



Discussion and Conclusion:
We can perform different operations of signals in MATLAB.

Task # 11:

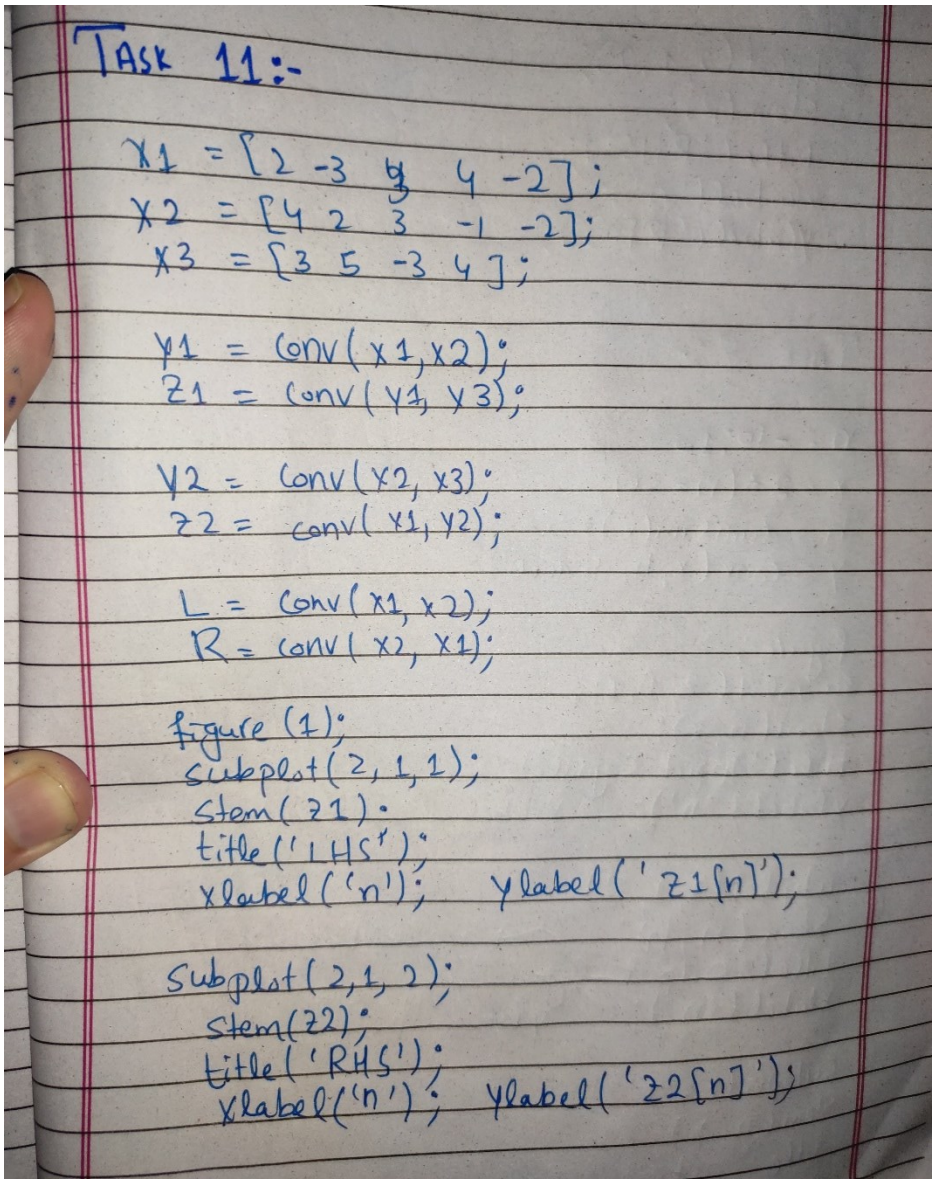
Given the signals:
x1[n]= 2δ[n] -3δ[n-1]+ 3δ[n-2] +4δ[n-3] -2δ[n-4]
x2[n]= 4δ[n]+ 2δ[n-1]+ 3δ[n-2] - δ[n-3] -2δ[n-4]
x3[n]= 3δ[n]+ 5δ[n-1]-3δ[n-2] +4 δ[n-3]

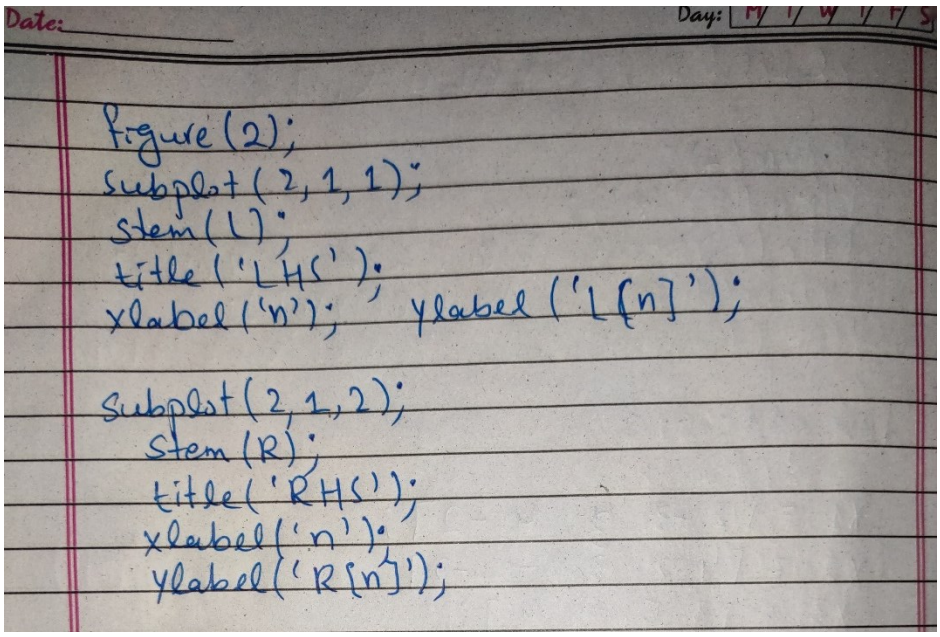
Verify that
x1[n] * (x2[n] * x3[n]) = (x1[n] * x2[n]) * x3[n]
x1[n] * x2[n]= x2[n] * x1[n]

Problem Analysis:
We can perform different operations of signals in MATLAB. We perform Convolution in MATLAB.

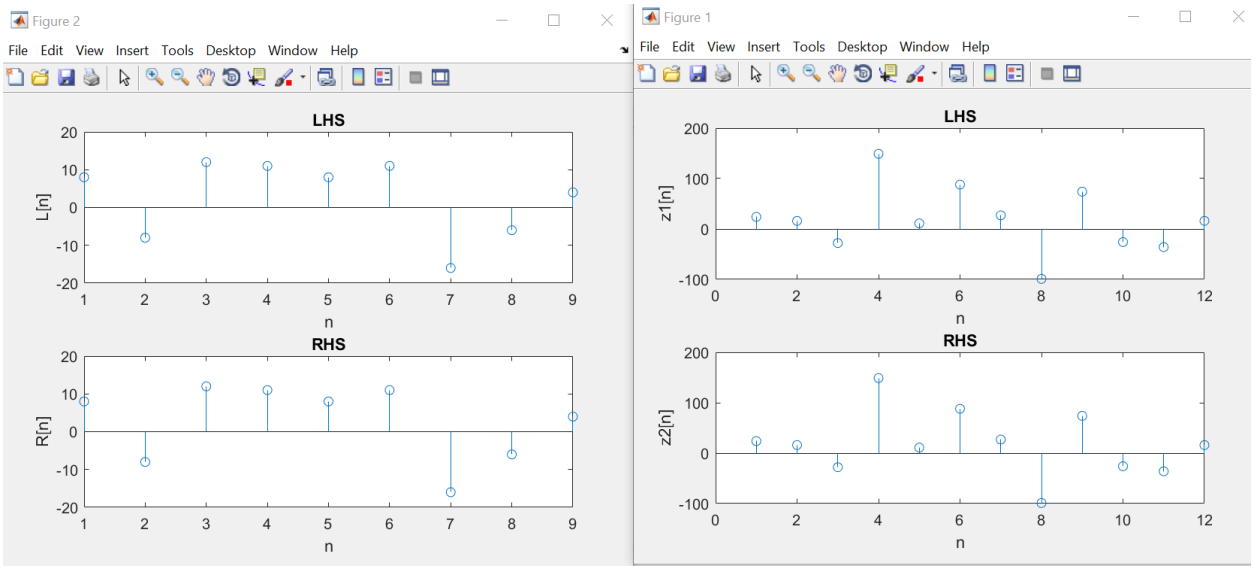
Algorithm:
- Write the code
- Execute it
- Observe the output

Code:





Output / Graphs / Plots / Results:



Discussion and Conclusion:

We can verify different convolution properties in MATLAB.