Title: Linear convolution and circular convolution

AIM: To perform linear and circular convolution operations on a given pair of sequences .

Objective: To perform linear and circular convolution operations on a given pair of sequences using MATlab.

Convolution is the process by which one may compute the overlap of two graphs.
Convolution is also interpreted as the area shared by the two graphs over time.
It is a blend between the two functions as one passes over the other.
So, given two functions F and G,
The convolution of the two is expressed and given by the following mathematical expression:

For continuous time,

$$f(t) * g(t) = \int_{-\infty}^{\infty} f(u).g(t-u)du$$

For discrete time,

$$f(n) * g(n) = \sum_{k=-\infty}^{\infty} f(k).g(n-k)$$

Write function in MATLAB for linear convolution of two sequences given below without using built-in MATLAB function.

Description

Linear convolution is a mathematical operation done to calculate the output of any **Linear-Time Invariant (LTI)** system given its input and impulse response.
It is applicable for both continuous and discrete-time signals.
We can represent Linear Convolution as
y(n)=x(n)*h(n)

Here, y(n) is the output (also known as convolution sum). x(n) is the input signal, and h(n) is the impulse response of the LTI system.

Graphically, when we perform linear convolution, there is a linear shift taking place.The formula for a linear convolution is given by.

$$\sum_{-\infty}^{\infty} x(k)h(n-k)$$

Matlab function

```
%{
The Function to find the Linear Convolution of two sequence
author :Sudip Biswas
%}
function c = convolution(a, b)
u = length(a);
v = length(b)
m =u +v -1;
a=[a,zeros(1,v-1)];
b=[b,zeros(1,u-1)];
for i = 1:1:m
   c(i)=0;
      for j = 1:1:i
              c(i) = c(i)+ a(j)*b((i-j+1));
      end
end
```

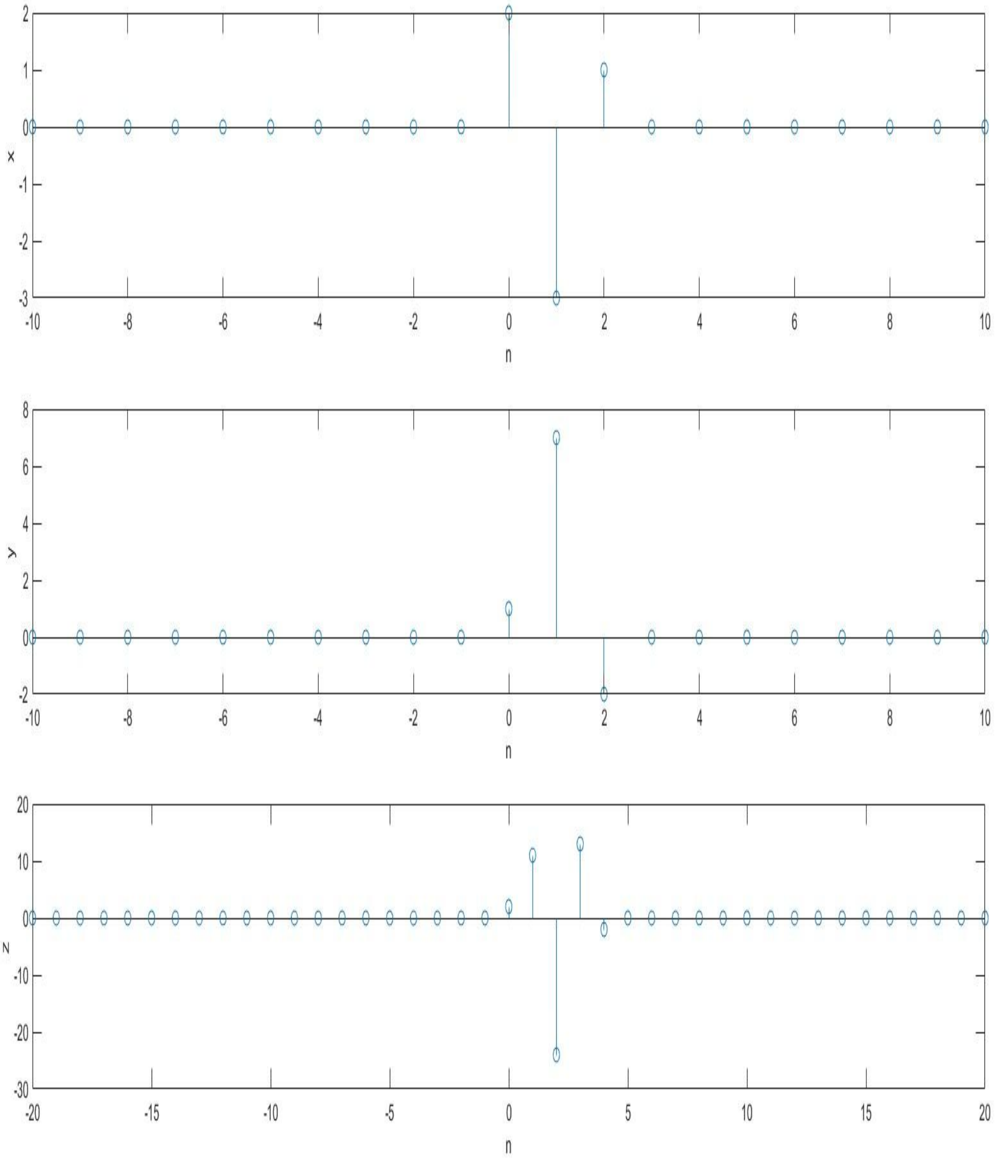
a.

$$x(n) = \begin{cases} 2 & n = 0 \\ -3 & n = 1 \\ 1 & n = 2 \\ 0 & \text{otherwise} \end{cases} \qquad y(n) = \begin{cases} 1 & n = 0 \\ 7 & n = 1 \\ -2 & n = 2 \\ 0 & \text{otherwise} \end{cases}$$

Matlab Program

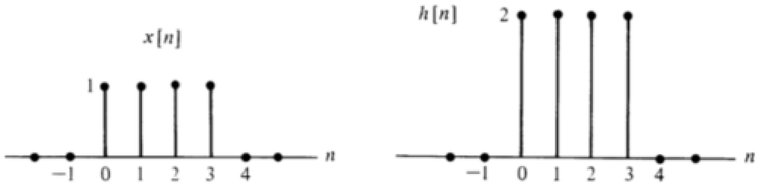
```
n = -10:10;
n1 = -20:20;
x = [0 0 0 0 0 0 0 0 0 0 2 -3 1 0 0 0 0 0 0 0 0 ];
y=  [0 0 0 0 0 0 0 0 0 0 1 7 -2 0 0 0 0 0 0 0 0 ];
subplot(3,1,1);
stem(n,x);
xlabel('n');
ylabel('x');
subplot(3,1,2);
stem(n,y);
xlabel('n');
ylabel('y');
subplot(3,1,3);
z = convolution(x,y);
stem(n1,z);
xlabel('n');
ylabel('z');
```

Output of Program



Result:
[0 0 <u>**2**</u> 11 -24 13 -2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0**]**

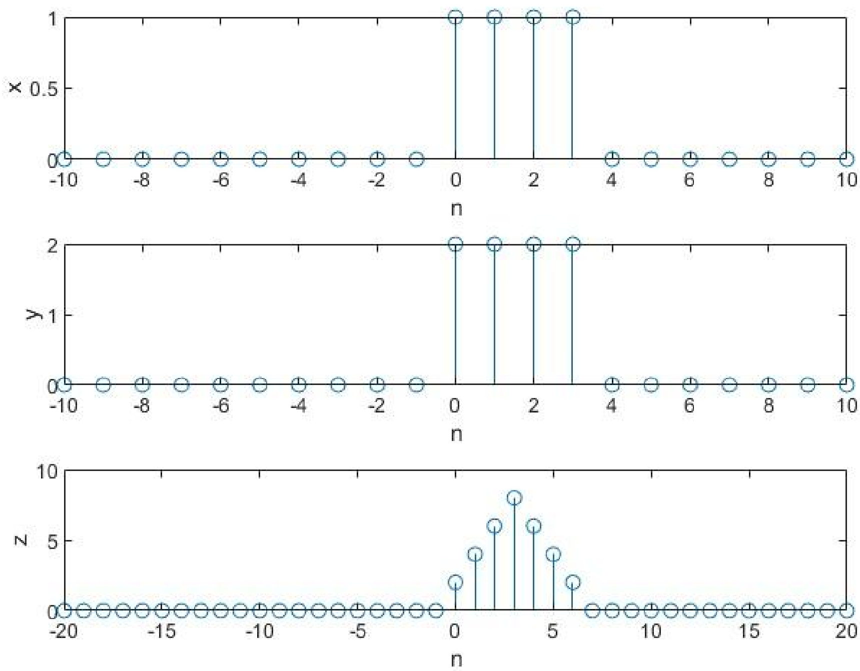
b.



Matlab Program

```
n = -10:10;
n1 = -20:20;
x = [0  0  0  0  0  0  0  0  0  0  1  1  1  1  0  0  0  0  0  0  0 ];
y=  [0  0  0  0  0  0  0  0  0  0  2  2  2  2  0  0  0  0  0  0  0 ];
subplot(3,1,1);
stem(n,x);
xlabel('n');
ylabel('x');
subplot(3,1,2);
stem(n,y);
xlabel('n');
ylabel('y');
subplot(3,1,3);
z = convolution(x,y);
stem(n1,z);
xlabel('n');
ylabel('z');
```

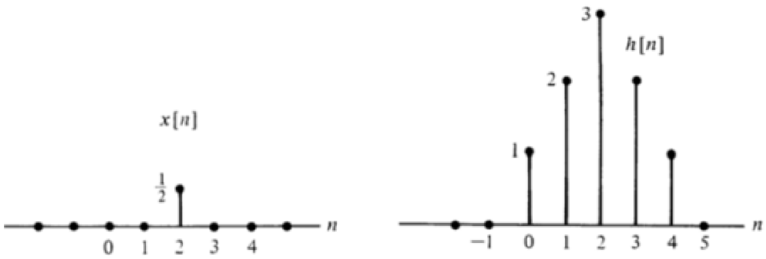
Output of Program



Result:

[0 0 **<u>2</u>** 4 6 8 6 4 2 0 0 0**]**

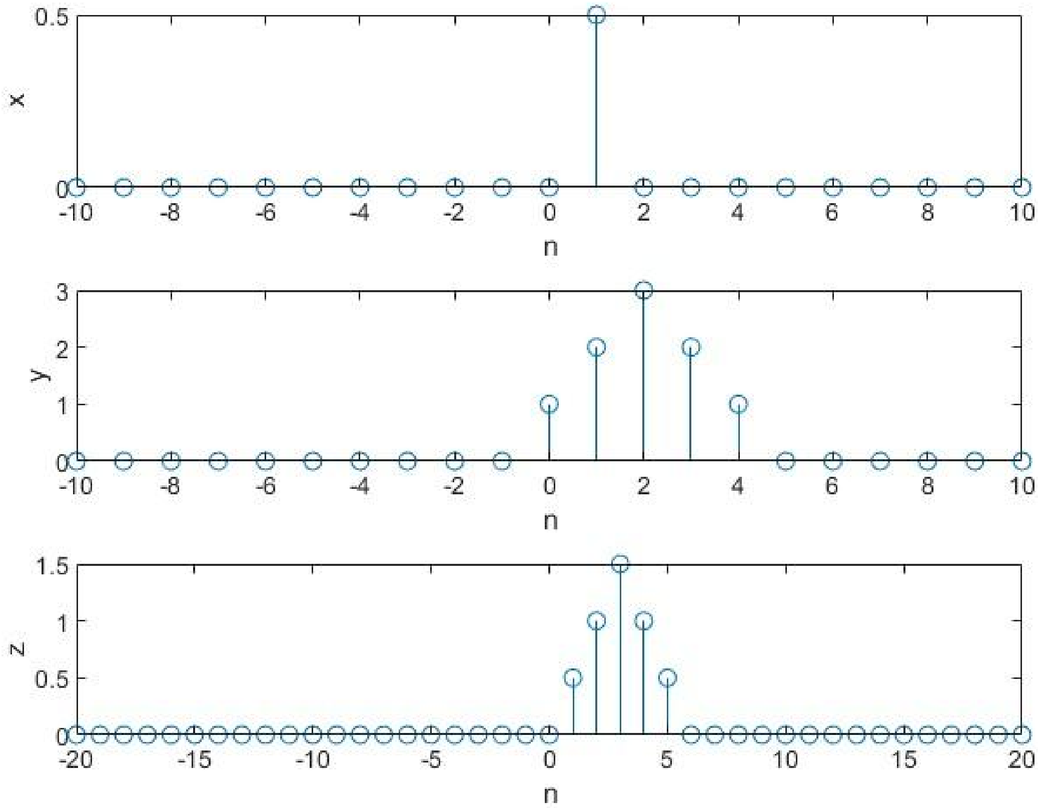
c.



Matlab Program

```
n = -10:10;
n1 = -20:20;
x = [0 0 0 0 0 0 0 0 0 0 0 .5 0 0 0 0 0 0 0 0 0 ];
y=  [0 0 0 0 0 0 0 0 0 0 1  2 3 2 1 0 0 0 0 0 0 ];
subplot(3,1,1);
stem(n,x);
xlabel('n');
ylabel('x');
subplot(3,1,2);
stem(n,y);
xlabel('n');
ylabel('y');
subplot(3,1,3);
z = convolution(x,y);
stem(n1,z);
xlabel('n');
ylabel('z');
```

Output of Program



Result:

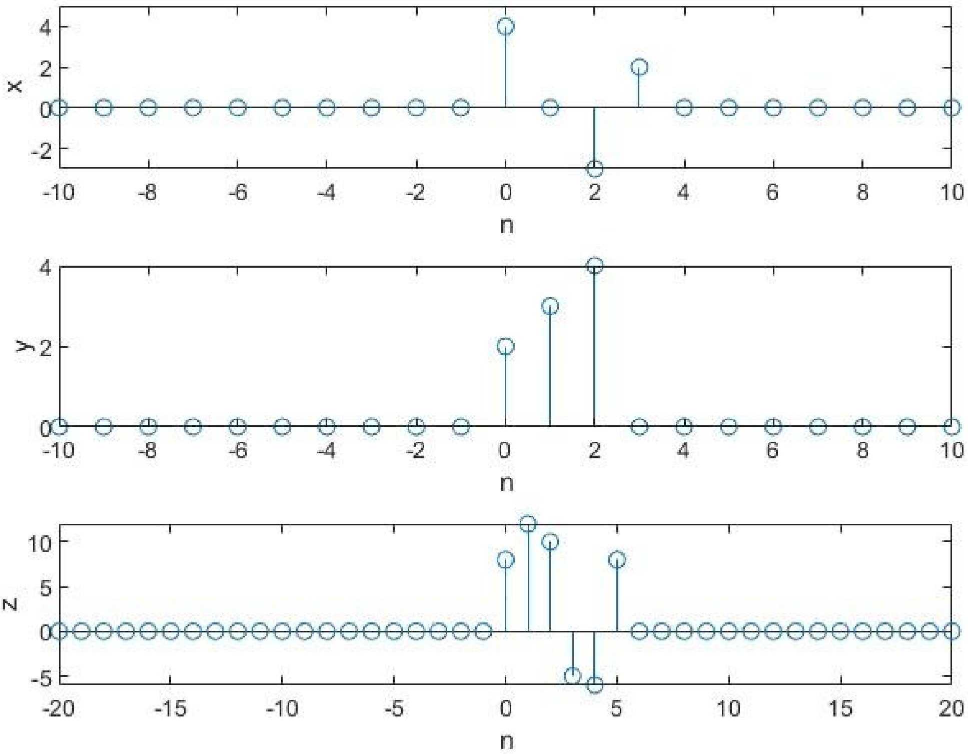
[0 <u>0</u> 0.5 1 1.5 1 0.5 0 0 **0]**

d. X[n]=4δ[n]-3δ[n-2]+2δ[n-3]; h[n]={2,4,3}

Matlab Program

```
n = -10:10;
n1 = -20:20;
x = [0  0  0  0  0  0  0  0  0  0  4  0  -3  2  0  0  0  0  0  0  0 ];
y=  [0  0  0  0  0  0  0  0  0  0  2  3   4  0  0  0  0  0  0  0  0 ];
subplot(3,1,1);
stem(n,x);
xlabel('n');
ylabel('x');
subplot(3,1,2);
stem(n,y);
xlabel('n');
ylabel('y');
subplot(3,1,3);
z = convolution(x,y);
stem(n1,z);
xlabel('n');
ylabel('z');
```

Output of Program



Result:
[0 **8** 12 10 -5 -6 8 0 0 0**]**

Write a function in MATLAB for circular convolution of two sequences given below without using built–in function.

Description
Circular convolution is essentially the same process as linear convolution. Just like linear convolution, it involves the operation of folding a sequence, shifting it, multiplying it with another sequence, and summing the resulting products. However, in circular convolution, the signals are all periodic. Thus the shifting can be thought of as actually being a rotation. Since the values keep repeating because of the periodicity. Hence, it is known as circular convolution.
Circular convolution is also applicable for both continuous and discrete-time signals.
We can represent Circular Convolution as
y(n)=x(n)⊕h(n)
Here y(n) is a periodic output, x(n) is a periodic input, and h(n) is the periodic impulse response of the LTI system.

Matlab Function

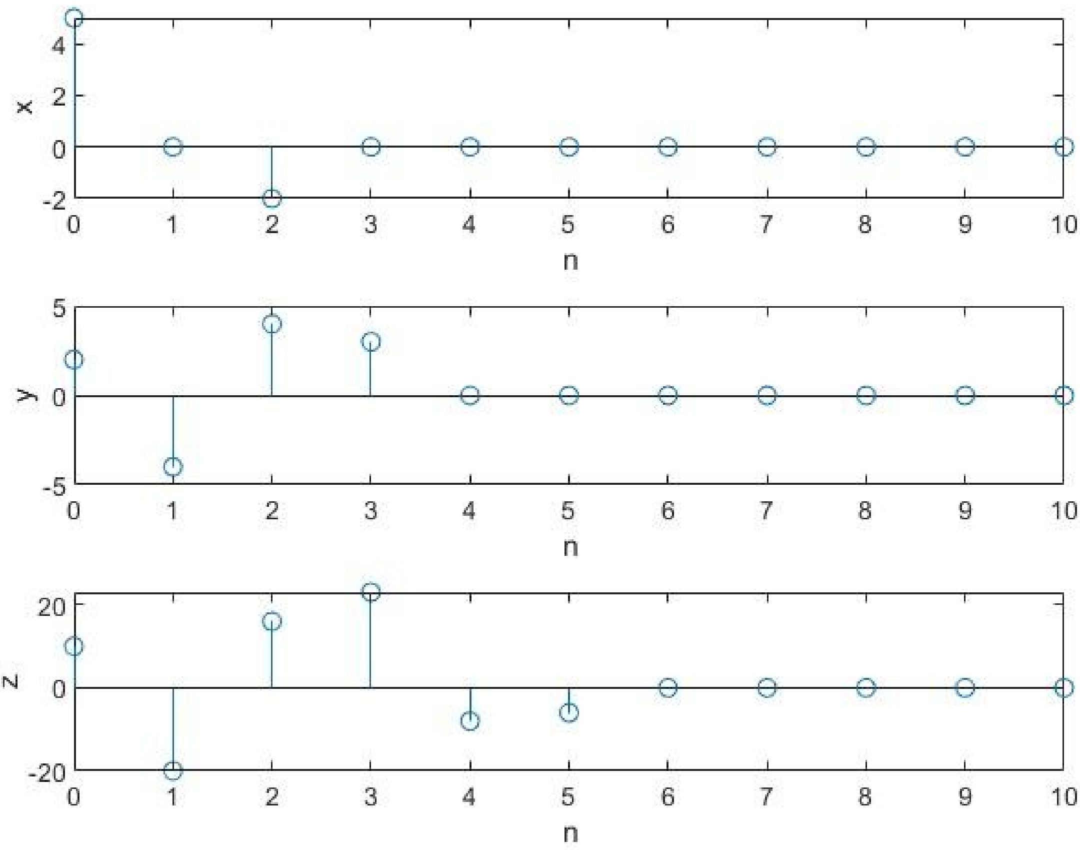
```
%{
The Function to find the Circular Convolution of two sequence
author :Sudip Biswas
%}
function c = circonvolution(a, b)
u = length(a);
v = length(b)
n =max(u,v);
if(u>v)
b=[b,zeros(1,u-v)];
else
a=[a,zeros(1,v-u)];
end
for i =0:n-1
   c(i+1)=0;
   for j = 0:n-1
       k = mod((i-j),n);
       c(i+1) = c(i+1) + a(j+1)*b(k+1);
   end
end
```

(a) X1[n]={5,0,-2}; X2[n]={2,-4,4,3}

Matlab Program

```
n = 0:10;
n1 = 0:10;
x = [5  0 -2 0 0 0 0 0 0 0 0 ];
y=  [2 -4  4 3 0 0 0 0 0 0 0 ];
subplot(3,1,1);
stem(n,x);
xlabel('n');
ylabel('x');
subplot(3,1,2);
stem(n,y);
xlabel('n');
ylabel('y');
subplot(3,1,3);
z = circonvolution(x,y);
stem(n1,z);
xlabel('n');
ylabel('z');
```

Output of program



Result:

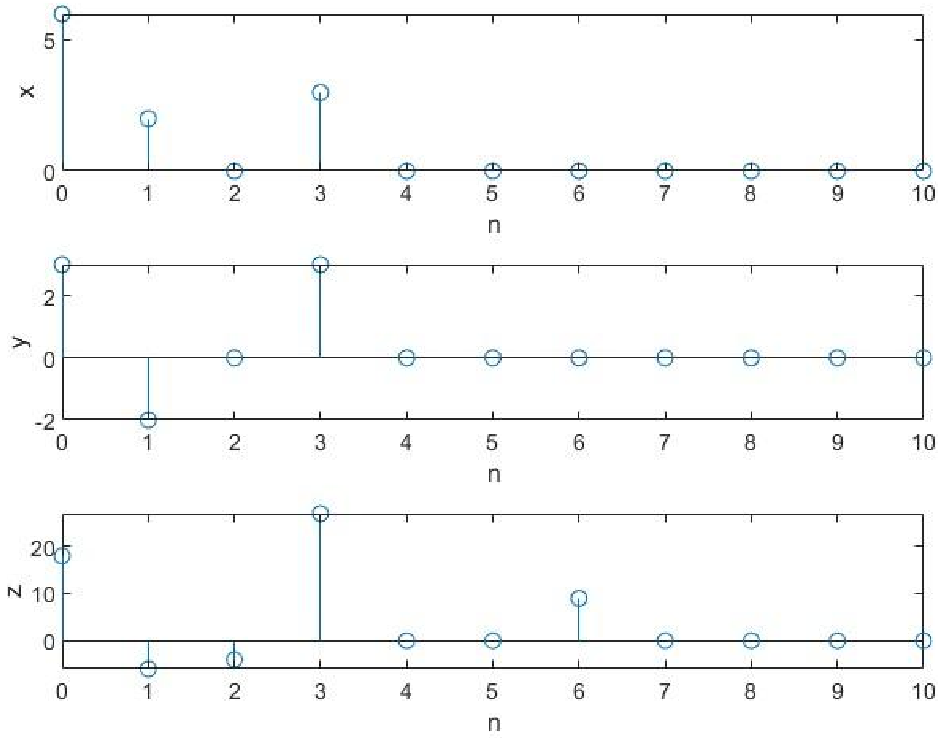
[10 -20 16 23 -8 -6 0 0 0 0 0]

(b) X1[n]={6,2,0,3}; X2[n]=3δ[n]-2δ[n-1]+3δ[n-3]

Matlab Program

```
n = 0:10;
n1 = 0:10;
x = [6  2  0 3 0 0 0 0 0 0 0 ];
y=  [2 -2  0 3 0 0 0 0 0 0 0 ];
subplot(3,1,1);
stem(n,x);
xlabel('n');
ylabel('x');
subplot(3,1,2);
stem(n,y);
xlabel('n');
ylabel('y');
subplot(3,1,3);
z = circonvolution(x,y);
stem(n1,z);
xlabel('n');
ylabel('z');
```

Output of program



Result:

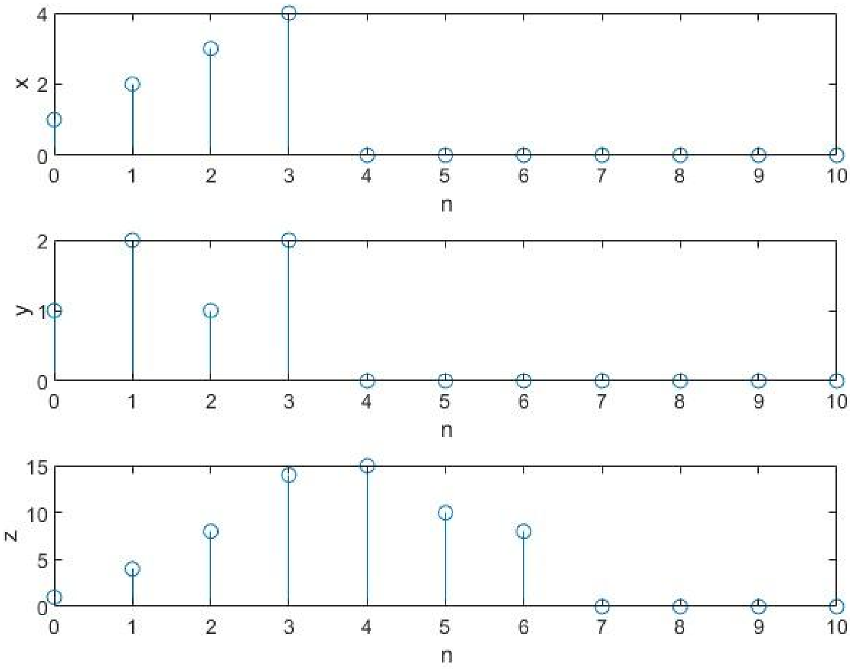
[18 -6 -4 27 0 0 9 0 0 0 0]

(c) X1[n] = {1, 2, 3, 4, 0, 0, 0}; X2[n] = {1, 2, 1, 2, 0, 0, 0}

Matlab Program

```
n = 0:10;
n1 =0:10;
x = [1 2  3 4 0 0 0 0 0 0 0 ];
y=  [1 2  1 2 0 0 0 0 0 0 0 ];
subplot(3,1,1);
stem(n,x);
xlabel('n');
ylabel('x');
subplot(3,1,2);
stem(n,y);
xlabel('n');
ylabel('y');
subplot(3,1,3);
z = circonvolution(x,y);
stem(n1,z);
xlabel('n');
ylabel('z');
```

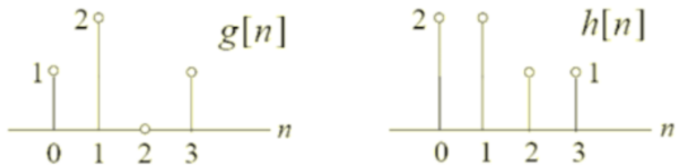
Output of program



Result:

[1 4 8 14 15 10 8 0 0 0 0**]**

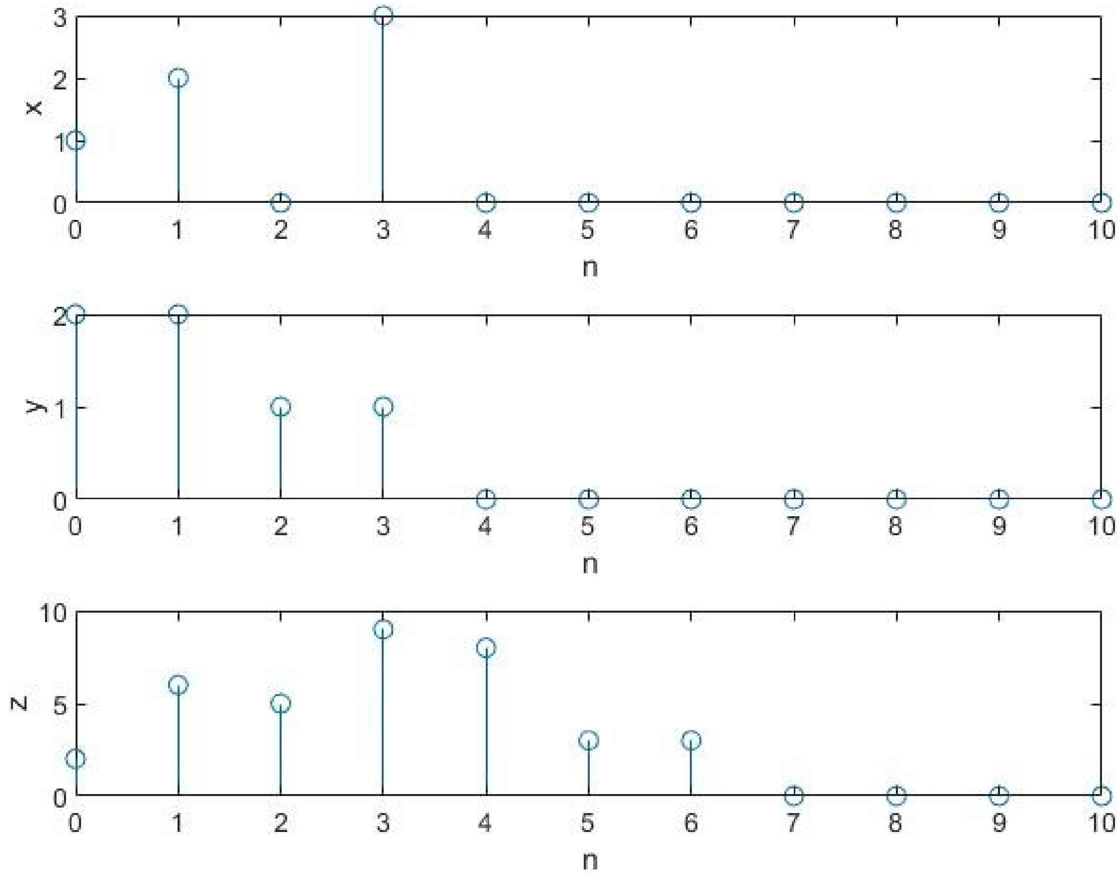
d.



Matlab Program

```
n = 0:10;
n1 = 0:10;
x = [1 2  0 3 0 0 0 0 0 0 0 ];
y=  [2 2  1 1 0 0 0 0 0 0 0 ];
subplot(3,1,1);
stem(n,x);
xlabel('n');
ylabel('x');
subplot(3,1,2);
stem(n,y);
xlabel('n');
ylabel('y');
subplot(3,1,3);
z = circonvolution(x,y);
stem(n1,z);
xlabel('n');
ylabel('z');
```

Output of program



Result:

[<u>**2**</u> 6 5 9 8 3 3 0 0 0 0**]**