Digital Signal Processing Lab

Name: Deep C. Patel

Roll No: 1401010

Lab Report

Lab Work:-

Lab – 2

1).

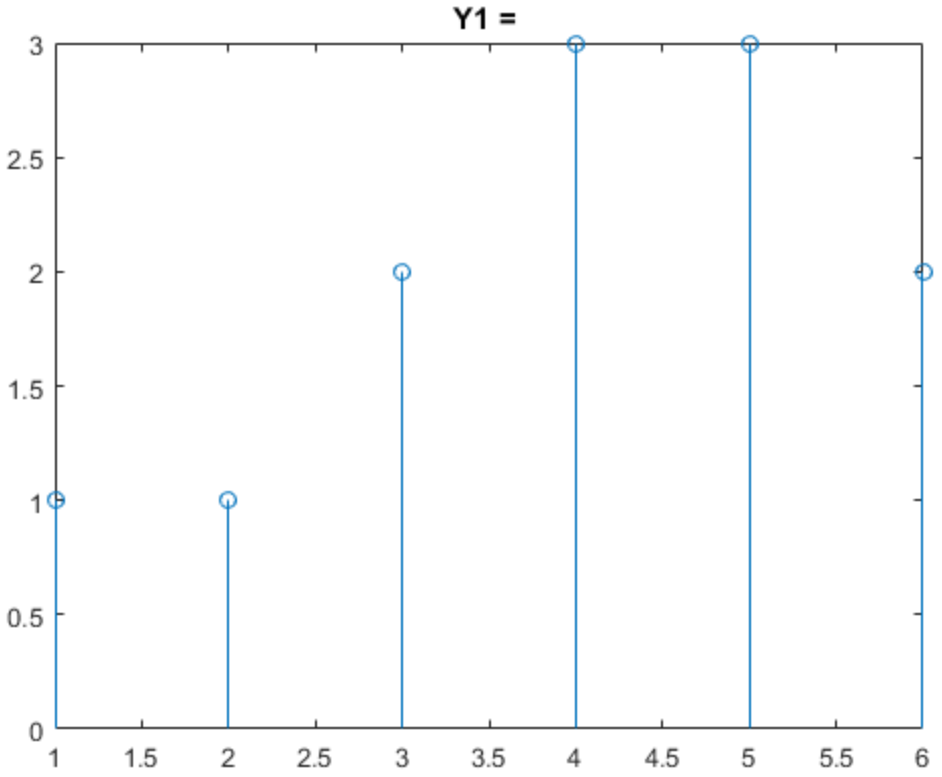
A: Inbuilt Convolution

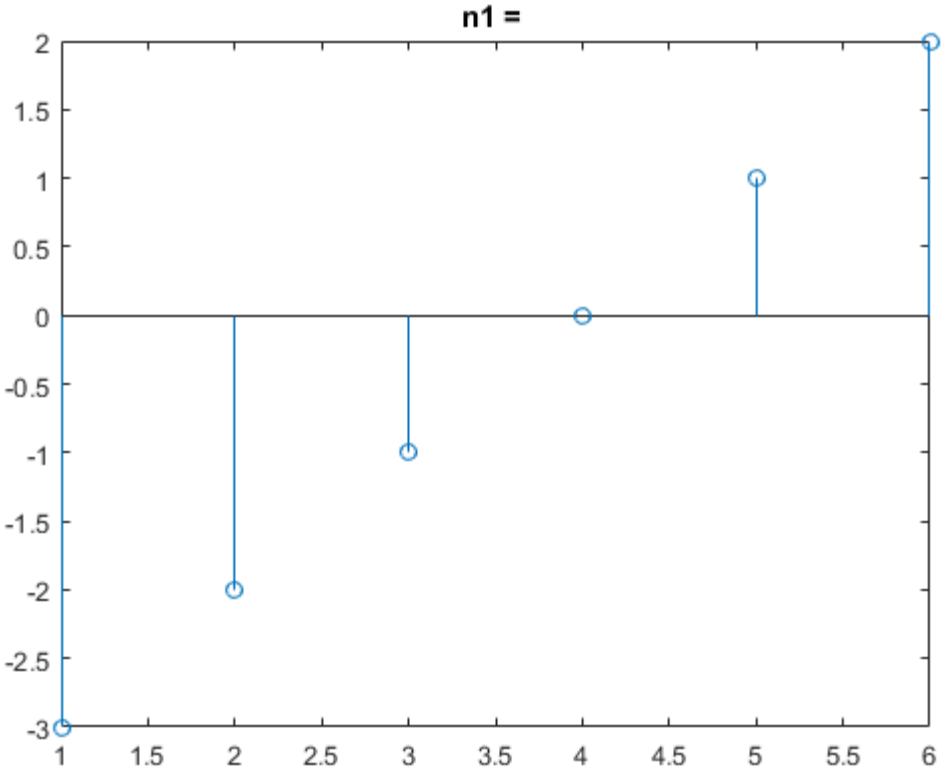
```
% Write Convolution Function using built in convolution

clc;
clear;

x1 = [1,2,2,1];
h1 = [1,-1,2];

n1_1 = [-1 0 1 2];
n2_1 = [-2 -1 0];

[y1, n1] = convn(x1,h1,n1_1,n2_1);

figure(1);
stem(y1);
title('Y1 = ');
figure(2);
stem(n1);
title('n1 = ');

x2 = [-2,0,1,-1,3];
h2 = [1,2,0,-1];

n1_2 = [-3 -2 -1 0 1];
n2_2 = [-1 0 1 2];

[y2, n2] = convn(x2,h2,n1_2,n2_2);
```

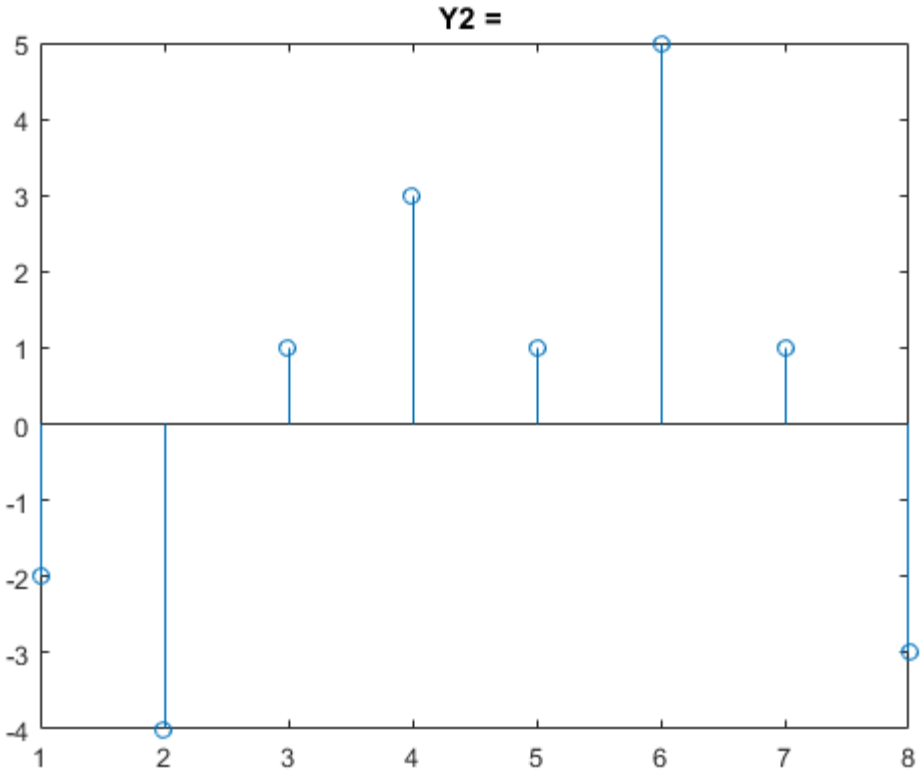
```
figure(3);
stem(y2);
title('Y2 = ');
figure(4);
stem(n2);
title('n2 = ');

x3 = [1,2,3,1];
h3 = [1,2,1,-1];

n1_3 = [-3 -2 -1 0];
n2_3 = [0 1 2 3];

[y3, n3] = convn(x3,h3,n1_3,n2_3);

figure(5);
stem(y3);
title('Y3 = ');
figure(6);
stem(n3);
title('n3 = ');

x4 = [9,1,5,4];
h4 = [0,2,2];

n1_4 = [0 1 2 3];
n2_4 = [-1 0 1];

[y4, n4] = convn(x4,h4,n1_4,n2_4);

figure(7);
stem(y4);
title('Y4 = ');
figure(8);
stem(n4);
title('n4 = ');
```

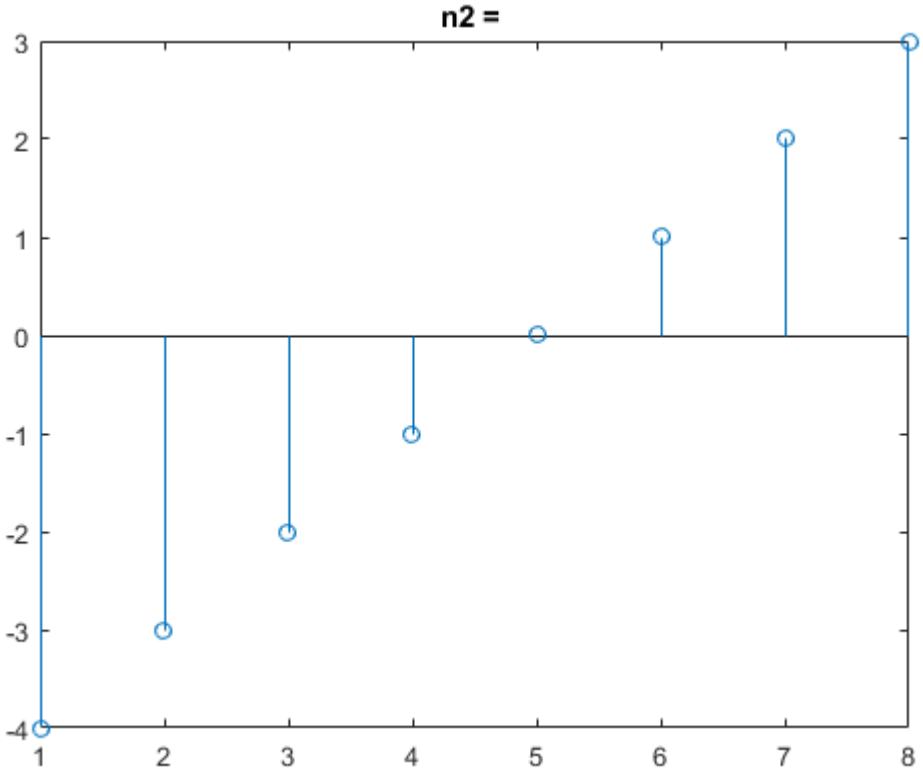
CONVN.M

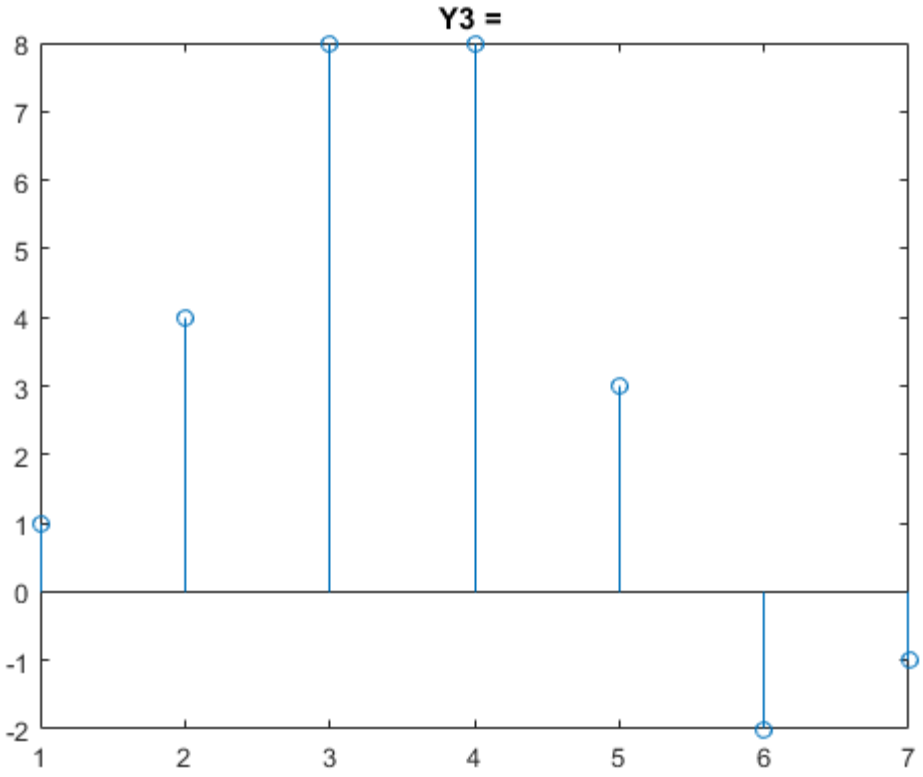
```
% Convolution function returning Index of Y, using inbuilt convolution
% function

function [y, n] = convn(x, h, n1, n2)

    y = conv(x,h,'full');
    c = min(n2) + min(n1);
    n = c:1:c+length(y)-1;
end
```

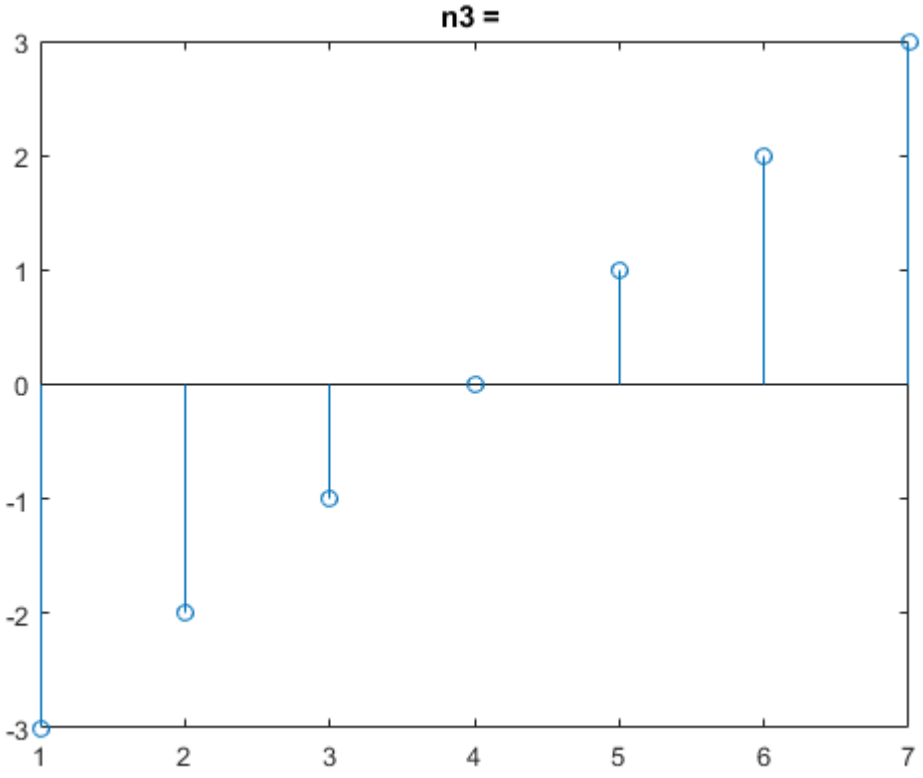

Y1 =
3
2.5
2
1.5
1
0.5
0
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6

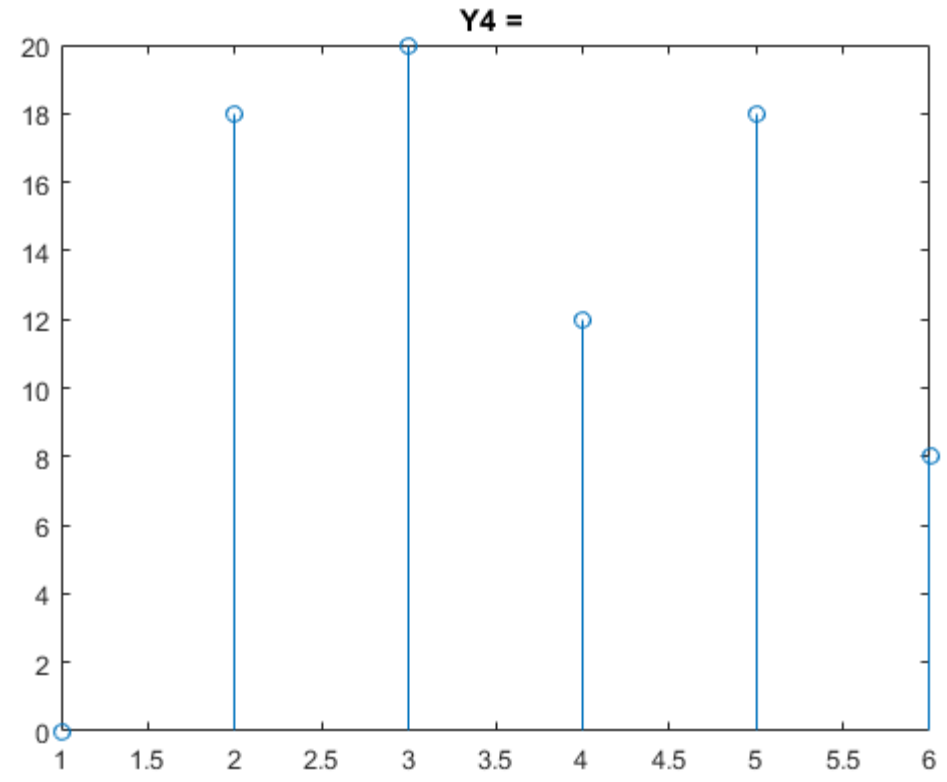

n1 =
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6

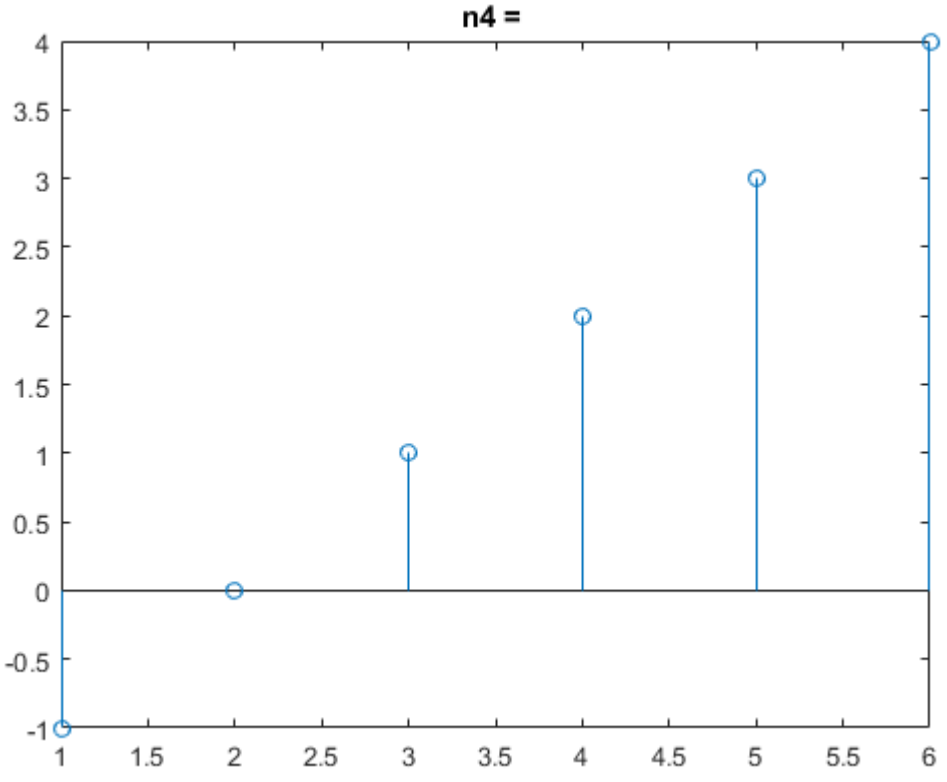

Y2 =
5
4
3
2
1
0
-1
-2
-3
-4
1
2
3
4
5
6
7
8


n2 =
3
2
1
0
-1
-2
-3
-4
1
2
3
4
5
6
7
8


Y3 =
8
7
6
5
4
3
2
1
0
-1
-2
1
2
3
4
5
6
7


n3 =
3
2
1
0
-1
-2
-3
1
2
3
4
5
6
7


Y4 =
20
18
16
14
12
10
8
6
4
2
0
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6


n4 =
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6

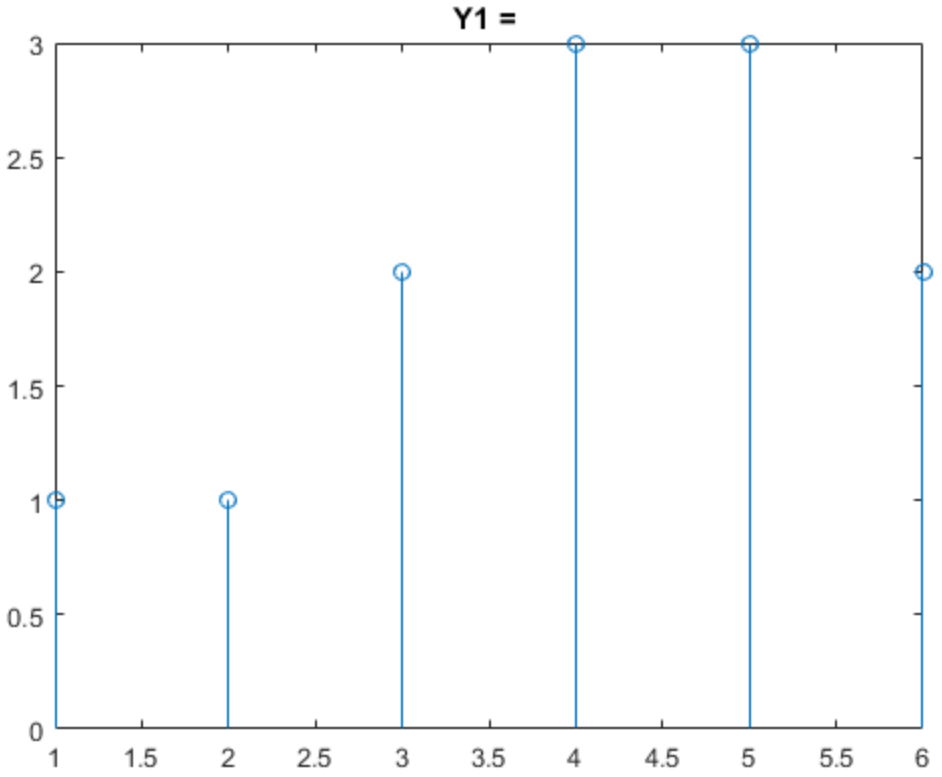
B: Self Made Convolution

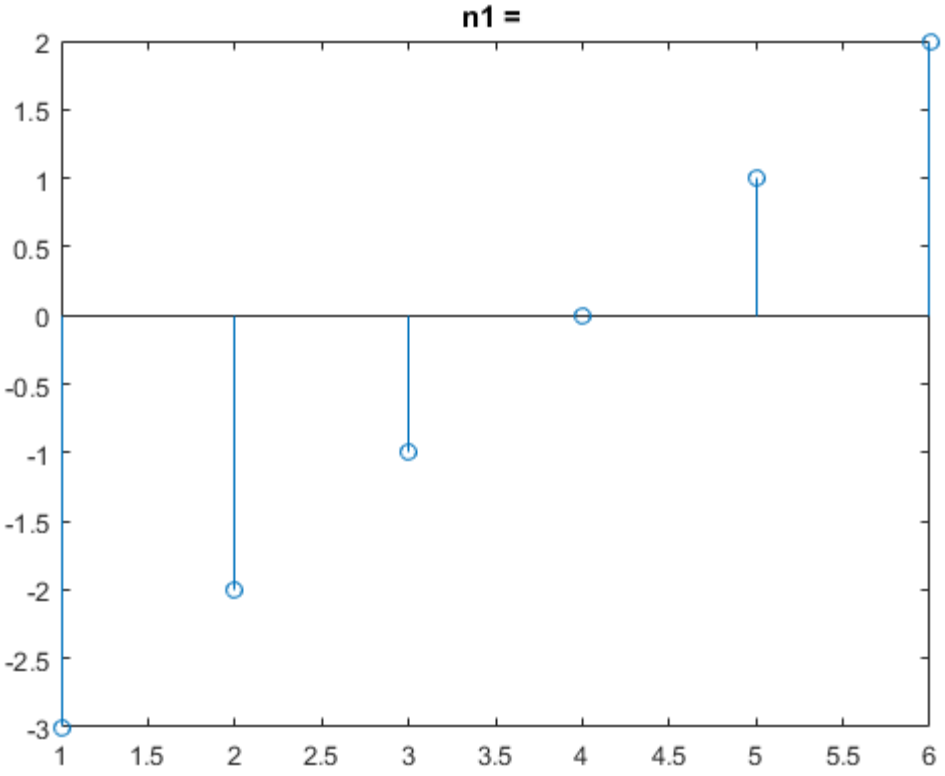
```
% Write Convolution Function using self made convolution

clc;
clear;

x1 = [1,2,2,1];
h1 = [1,-1,2];

n1_1 = [-1 0 1 2];
n2_1 = [-2 -1 0];

[y1, n1] = convi(x1,h1,n1_1,n2_1);

figure(1);
stem(y1);
title('Y1 = ');
figure(2);
stem(n1);
title('n1 = ');

x2 = [-2,0,1,-1,3];
h2 = [1,2,0,-1];

n1_2 = [-3 -2 -1 0 1];
n2_2 = [-1 0 1 2];

[y2, n2] = convi(x2,h2,n1_2,n2_2);

figure(3);
stem(y2);
title('Y2 = ');
figure(4);
stem(n2);
title('n2 = ');

x3 = [1,2,3,1];
h3 = [1,2,1,-1];

n1_3 = [-3 -2 -1 0];
n2_3 = [0 1 2 3];

[y3, n3] = convi(x3,h3,n1_3,n2_3);

figure(5);
stem(y3);
title('Y3 = ');
figure(6);
stem(n3);
```

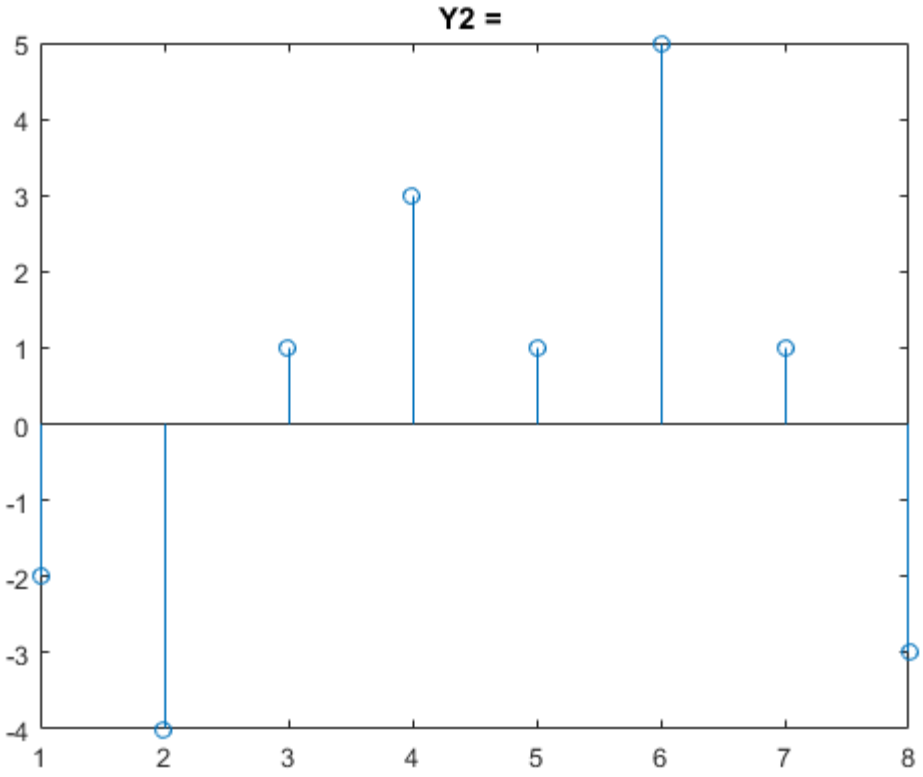
```
title('n3 = ');

x4 = [9,1,5,4];
h4 = [0,2,2];

n1_4 = [0 1 2 3];
n2_4 = [-1 0 1];

[y4, n4] = convi(x4,h4,n1_4,n2_4);

figure(7);
stem(y4);
title('Y4 = ');
figure(8);
stem(n4);
title('n4 = ');
```

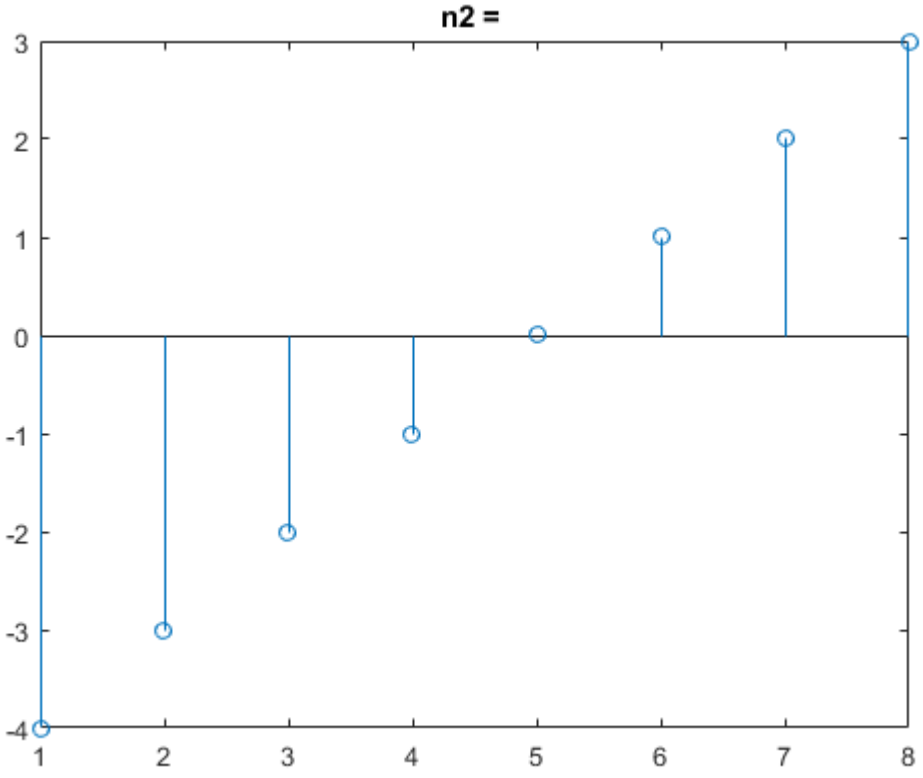
CONVI.M

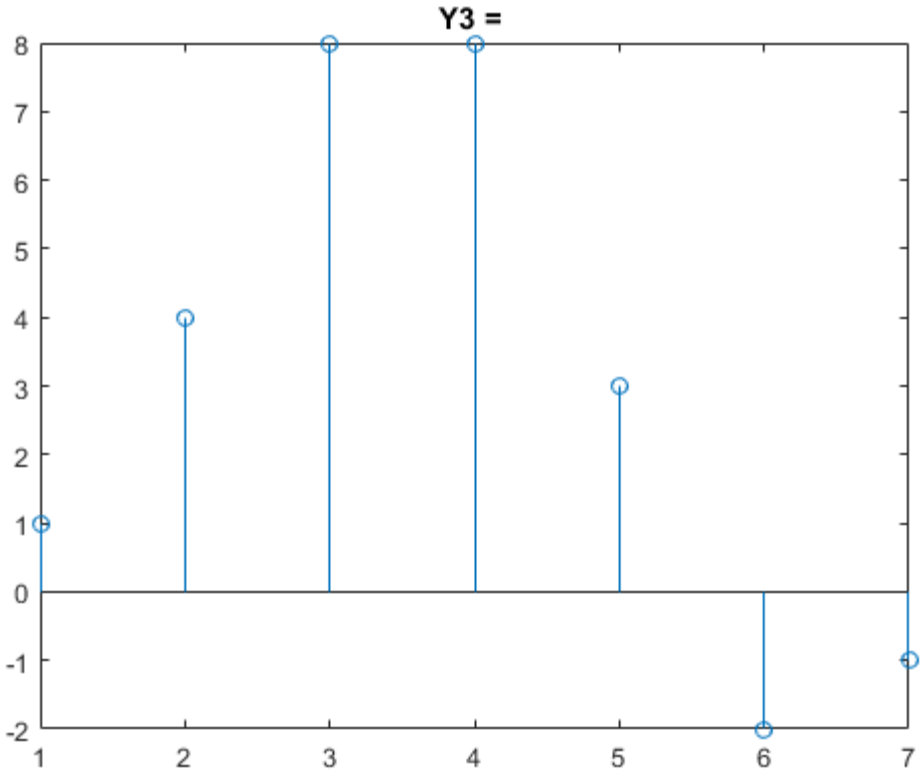
```
% Convolution function returning Index of Y, using custom convolution
% function

function [y, n] = convi(x, h, n1, n2)

    lenX = length(x);
    lenH = length(h);
    h = flip(h);
    y = zeros(1,lenH+lenX-1);

    h = [zeros(1,lenX) h zeros(1,lenX-1)];
    newlenH = length(h);

    m = lenH+lenX;
    i=1;

    while m ~= 1
        y(1,i) = sum(x.*h(1,m:newlenH-(i-1)));
        m = m-1;
        i = i+1;
    end

    c = min(n2) + min(n1);

    n = c:1:c+length(y)-1;

end
```

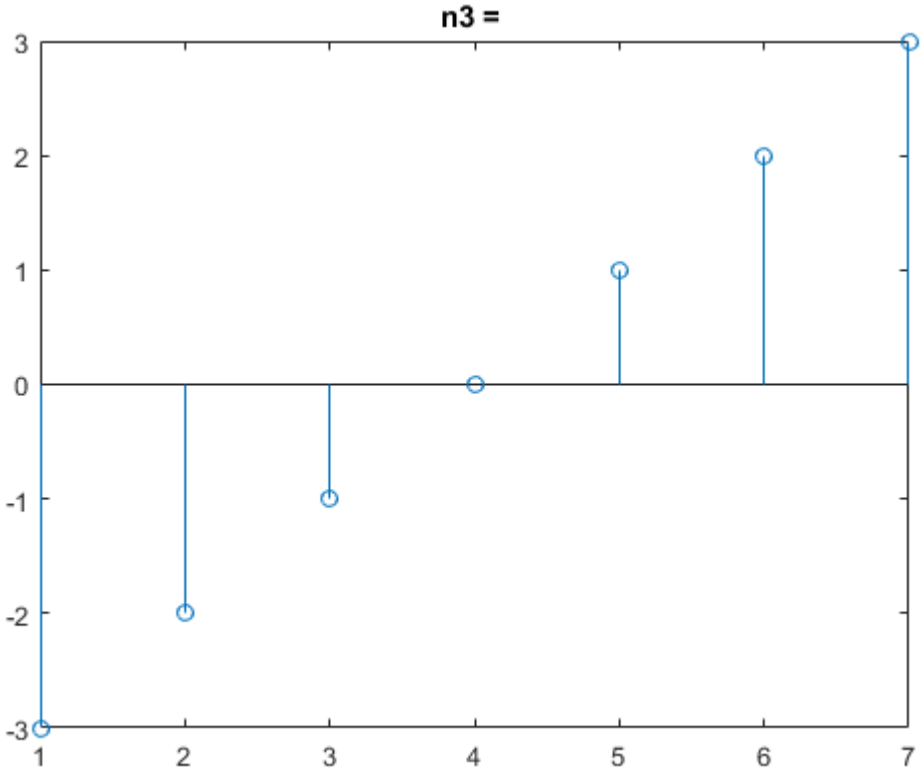

Y1 =
3
2.5
2
1.5
1
0.5
0
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6

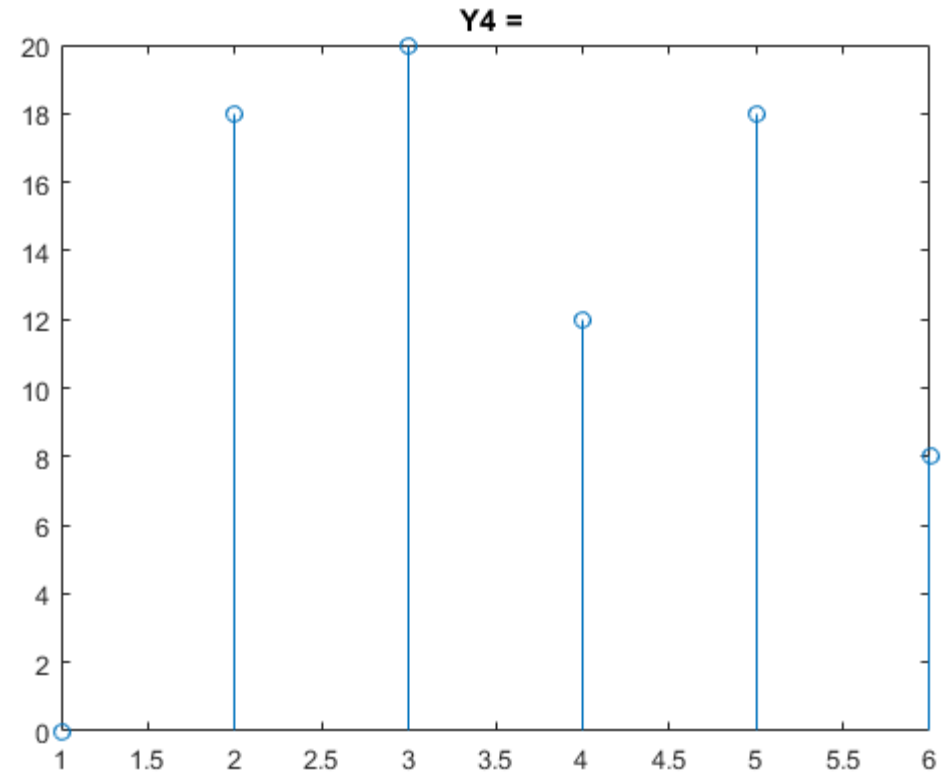

n1 =
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6

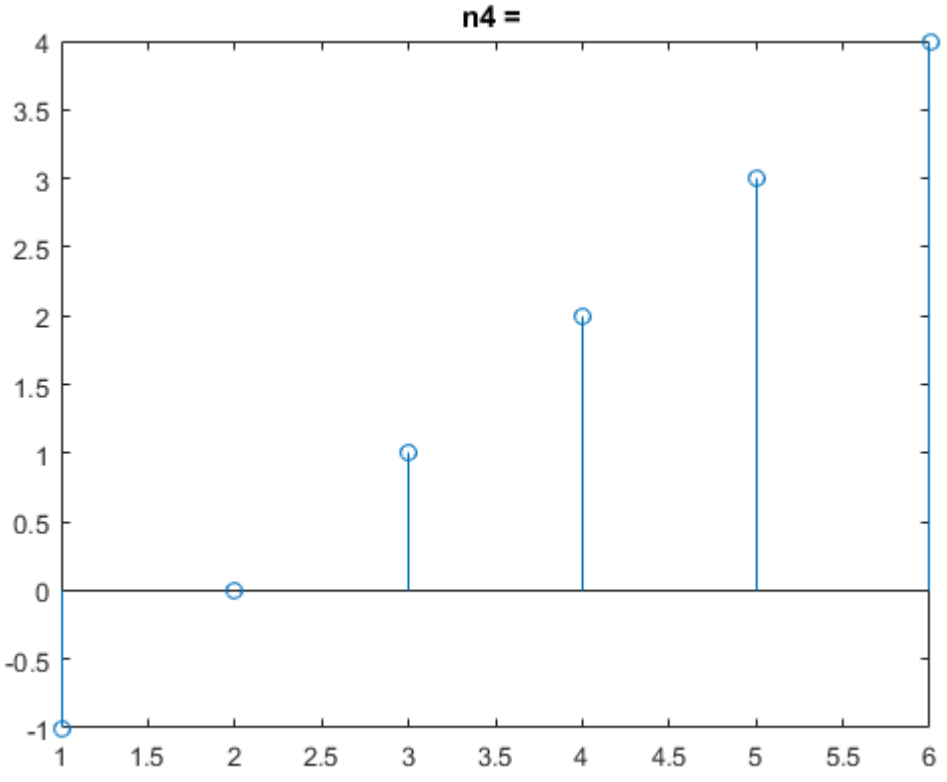

Y2 =
5
4
3
2
1
0
-1
-2
-3
-4
1
2
3
4
5
6
7
8


n2 =
3
2
1
0
-1
-2
-3
-4
1
2
3
4
5
6
7
8


Y3 =
8
7
6
5
4
3
2
1
0
-1
-2
1
2
3
4
5
6
7


n3 =
3
2
1
0
-1
-2
-3
1
2
3
4
5
6
7


Y4 =
20
18
16
14
12
10
8
6
4
2
0
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6


n4 =
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6

2).

```
% Find the linear convolution for following infinite length sequences and Plot
required outputs.

clc;
clear;

xs = input('Start Index of X:');
hs = input('Last Index of X:');
xl = input('Start Index of h:');
hl = input('Last Index of h:');

n1 = xs:1:1;

n2 = xl:1:hl;

lengthn1 = length(n1);

lengthn2 = length(n2);

ones1 = ones(1,lengthn1);

ones2 = ones(1,lengthn2);

for i=1:lengthn1
    if(n1(i)<0)
        ones1(i) = 0;
    end
end

for i=1:lengthn2
    if(n2(i)<0)
        ones2(i) = 0;
    end
end

[y,n] = convi(ones1,ones2,n1,n2);
subplot(3,1,1);
stem(n,y);
title('u(n)*u(n)');

for i=1:lengthn1
    if(n1(i)<0)
        ones1(i) = 0;
    end
    ones1(i) = ones1(i)*cos(n1(i)*pi);
end

for i=1:lengthn2
```

```
    if(n2(i)<0)
        ones2(i) = 0;
    end
end

[y,n] = convi(ones1,ones2,n1,n2);
subplot(3,1,2);
stem(n,y);
title('cos(n*pi).u(n) * u(n)');

for i=1:lengthn1
    if(n1(i)<0)
        ones1(i) = 0;
    end
    ones1(i) = ones1(i)*n1(i);
end

for i=1:lengthn2
    if(n2(i)<0)
        ones2(i) = 0;
    end
    ones2(i) = (6+n(i))*ones2(i);
end

[y,n] = convi(ones1,ones2,n1,n2);
subplot(3,1,3);
stem(n,y);
title('n.u(n) * (6+n).u(n)');
```

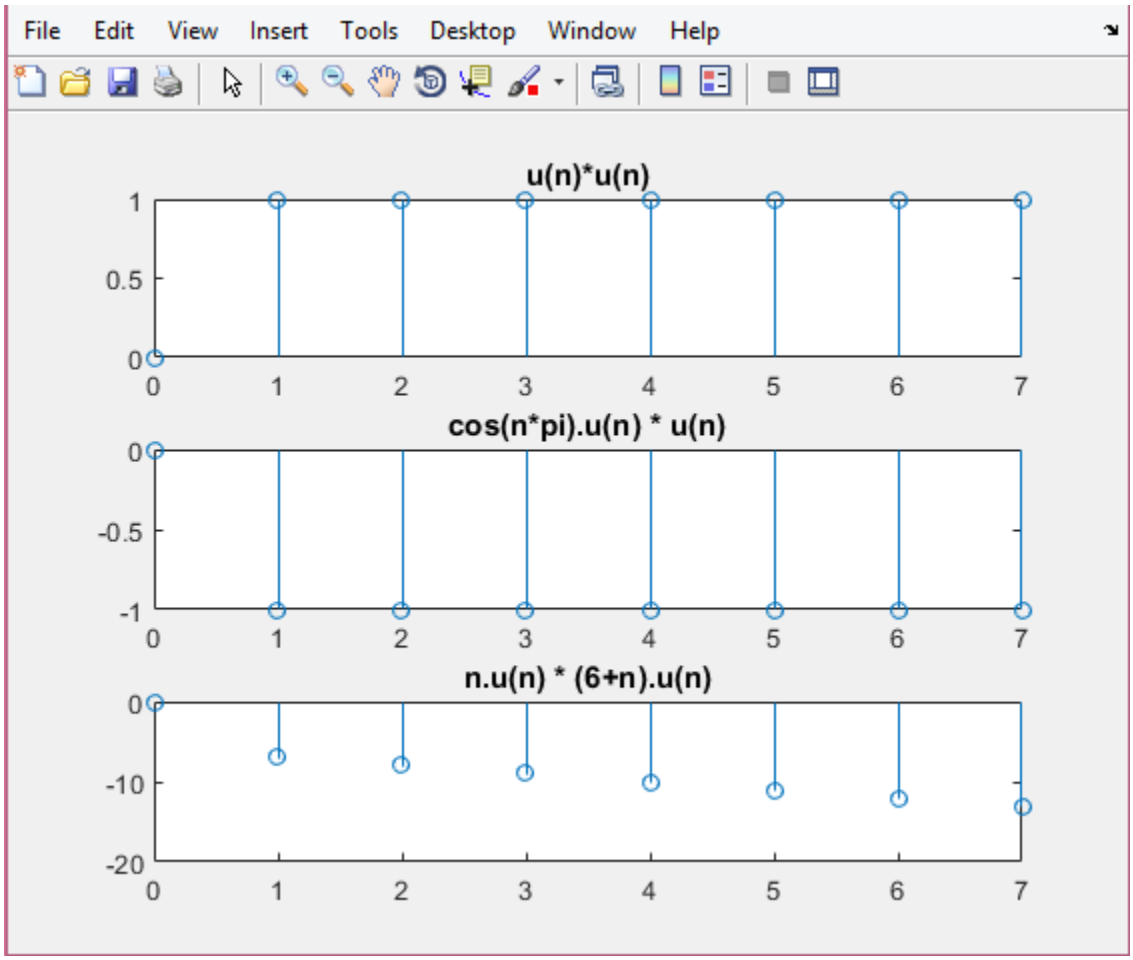
CONVI.M

```
% Convolution function returning Index of Y, using custom convolution
% function

function [y, n] = convi(x, h, n1, n2)

    lenX = length(x);
    lenH = length(h);
    h = flip(h);
    y = zeros(1,lenH+lenX-1);

    h = [zeros(1,lenX) h zeros(1,lenX-1)];
    newlenH = length(h);

    m = lenH+lenX;
    i=1;

    while m ~= 1
        y(1,i) = sum(x.*h(1,m:newlenH-(i-1)));
        m = m-1;
```

```
        i = i+1;
    end

    c = min(n2) + min(n1);

    n = c:1:c+length(y)-1;

end
```

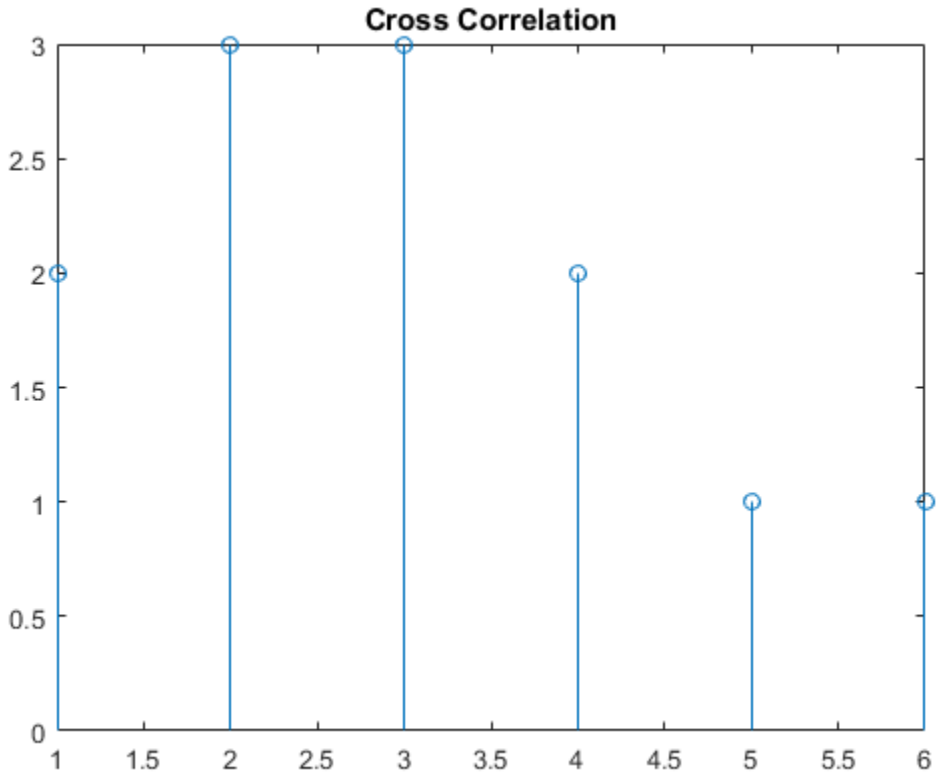
Input:-

```
Start Index of X:1
Last Index of X:5
Start Index of h:-1
Last Index of h:6
```



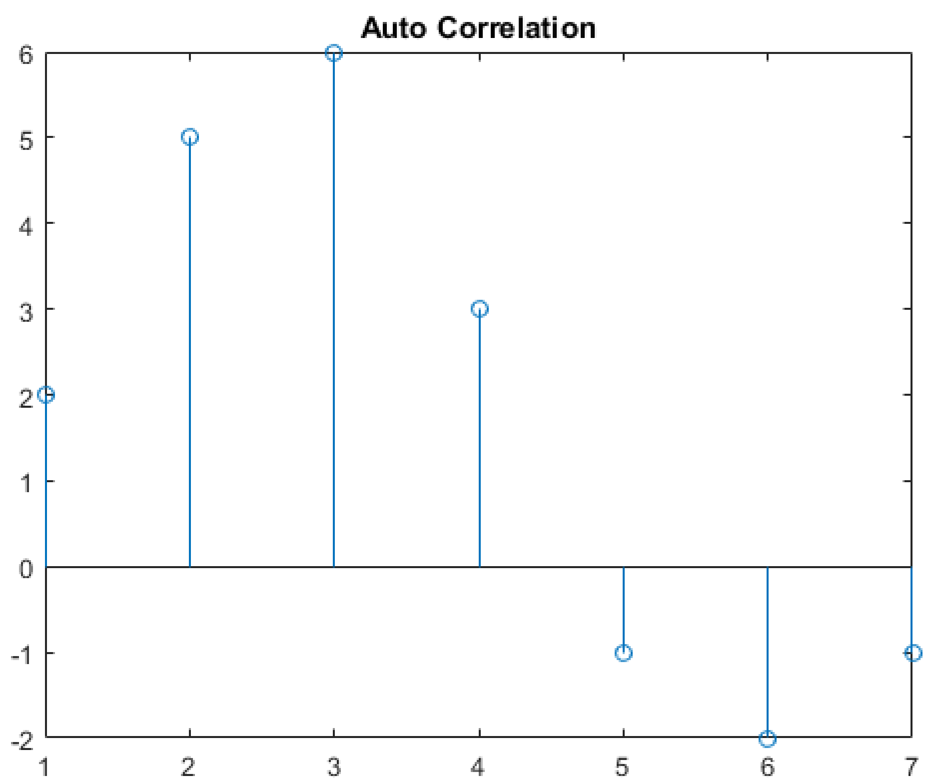
3).

```
% Find cross-correlation between two sequences by developing your own function.
% Verify your program for following sequence and Plot required outputs.

clc;
clear;

x1 = [1,2,2,1];
h1 = [1,-1,2];

n1_1 = [-1 0 1 2];
n2_1 = [-2 -1 0];

y = cross_corr(x1,h1,n1_1,n2_1);

figure(1)
stem(y);
title('Cross Correlation');
```



4).

```
% Develop function for Auto-correlation and find out auto-correlation for any sequence
and
% plot required outputs.

clc;
clear;

x1 = [1,2,2,1];

n1_1 = [-1 0 1 2];

y = cross_corr(x1,n1_1);

figure(1)
stem(y);
title('Auto Correlation');
```



-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-x-