**Batch: \_\_B3\_\_\_Roll No.:\_\_1611124\_\_Exp NO:\_3\_**

|  |
| --- |
| **Title:** To compute linear and circular convolution of two discrete time sequences using Matlab. |

**Objective:** To familiarize the beginner to MATLAB by introducing the basic features and commands of the program.

**Expected Outcome of Experiment:**

|  |  |
| --- | --- |
| **CO** | **Outcome** |
| **CO3** | To understand the concept of convolution and perform different convolution operations on the given input signals. |

**Books/ Journals/ Websites referred:**

1. http://www.mathworks.com/support/
2. www.math.mtu.edu/~msgocken/intro/intro.html
3. www.mccormick.northwestern.edu/docs/efirst/matlab.pdf
4. A.Nagoor Kani “Digital Signal Processing”, 2nd Edition, TMH Education.

**Pre Lab/ Prior Concepts:**

**Convolution**

Discrete time convolution is a method of finding response of linear time invariant system. It is based on the concepts of linearity and time invariance and assumes that the system information

is known in terms of its impulse response h[n].

Convolution is defined as

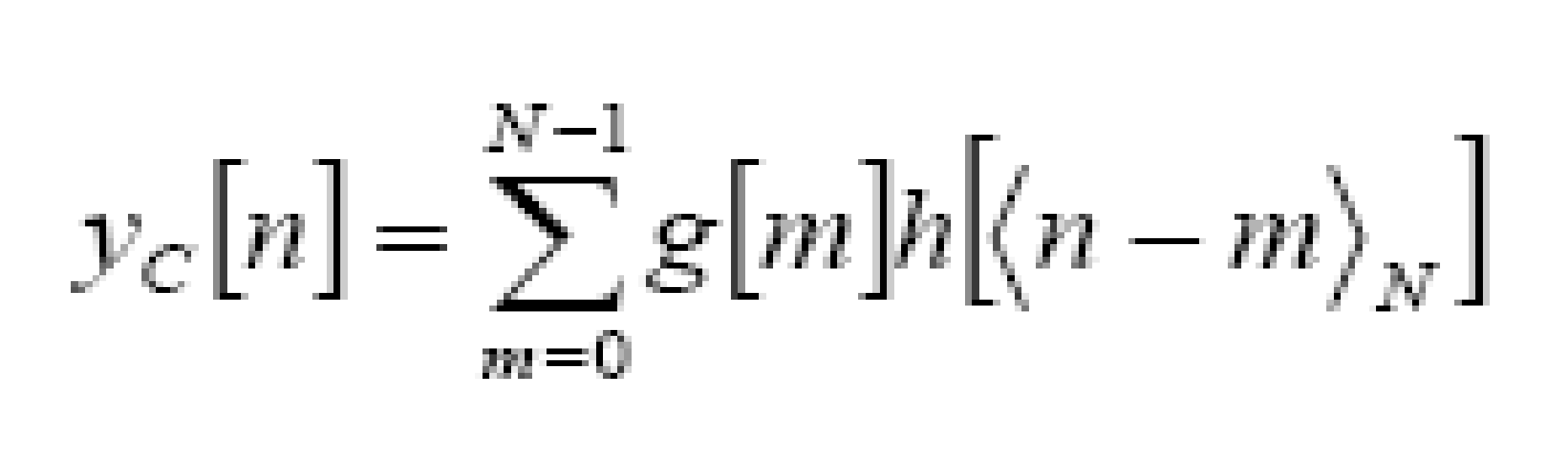
∞

Y[n] = Σ h[k]x [n-k] =h[n]\*x[n] k=-∞

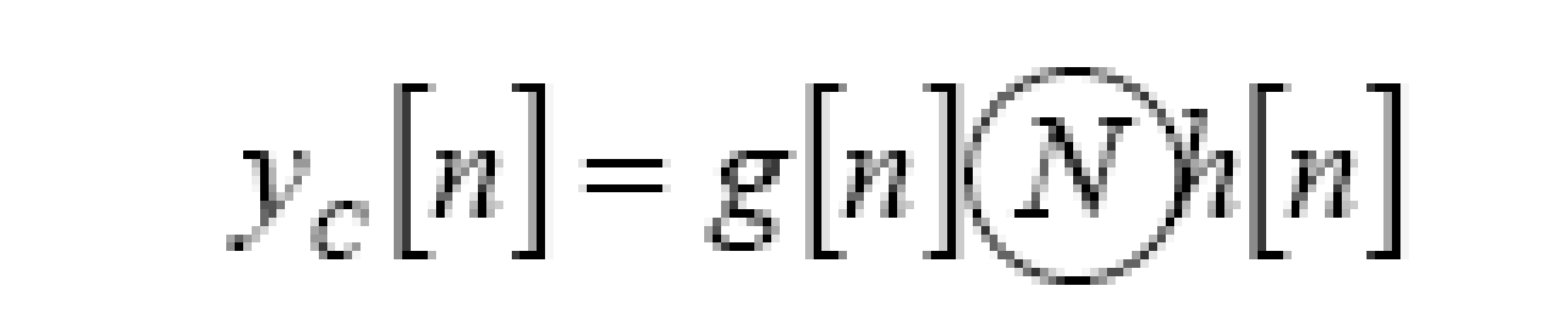
Convolution consists of folding, shifting, Multiplication and summation operations.

**Circular Convolution**

Circular convolution between two length N sequences can be carried out as shown by the expression below:

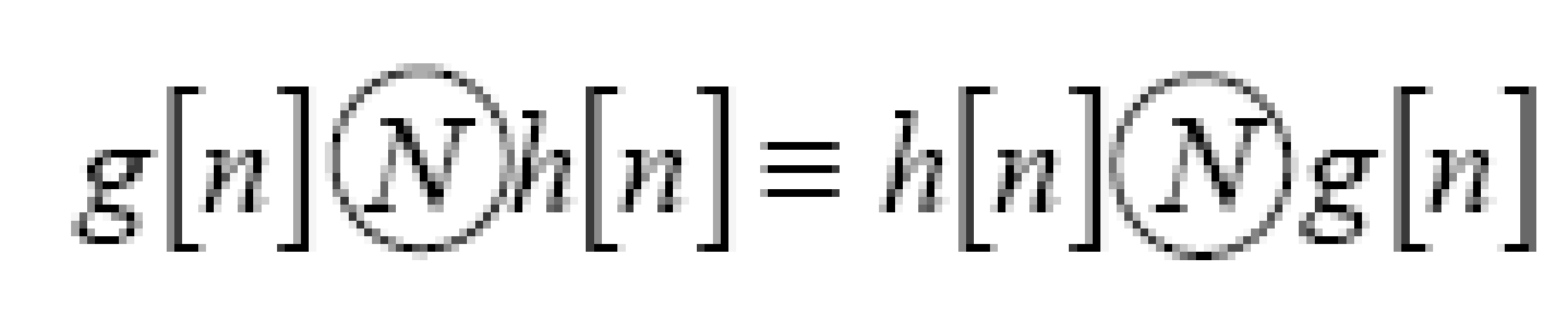


Since the above operation involves two length-N sequences it is referred to as the N-point circular convolution and denoted by:

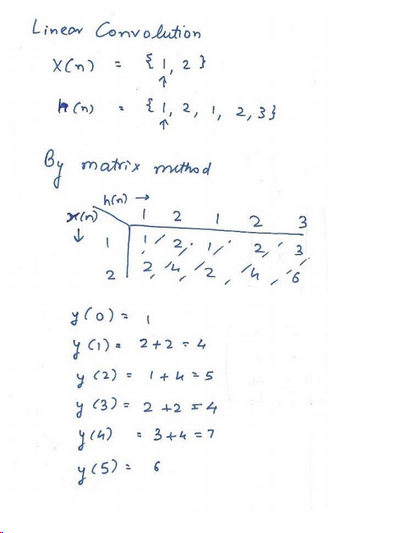


As in linear convolution circular convolution is commutative.

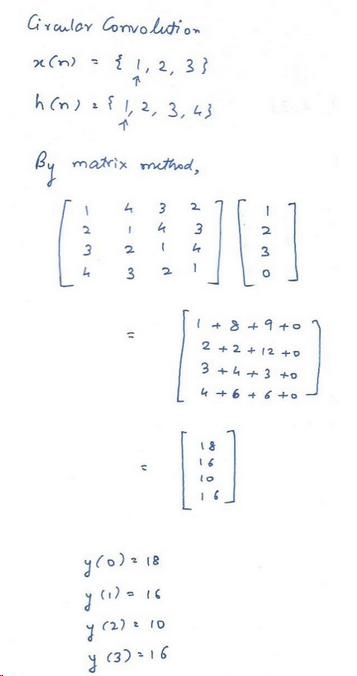
i.e.



**Example of Linear Convolution:**

****

**Example of Circular Convolution:**

****

**Implementation details along with screenshots:**

**Linear Convolution:**

x=[];

h=[];

in=[];

num1 = input('Enter the no of elements in x : ');

index1 = input('Enter the starting index of x : ');

last1=index1+num1-1;

fprintf('Enter the first signal x\n');

for i=index1:last1

value=input('Enter the value : ');

x = [x value];

in = [in i];

end

num2 = input('Enter the no of elements in h : ');

index2 = input('Enter the starting index of h : ');

last2=index2+num2-1;

fprintf('Enter the second signal h \n');

for i=index2:last2

value=input('Enter the value : ');

h = [h value];

end

length = num1+num2-1;

index3 = index1+index2;

last3 = last1+last2;

y =[];

sum =0;

offset1 = -index1;

offset2 = -index2;

index2

last2

for n=index3:last3

for k=index1:last1

a = k+offset1+1

b = n - k + offset2 + 1

if n-k < index2 || n-k > last2

value = 0;

sum = sum + value;

else

value = x(a).\*h(b);

sum = sum + value;

end

end

sum

y = [y sum];

sum = 0;

end

subplot(3, 1, 1);

stem(x);

title("x(n)");

subplot(3, 1, 2);

stem(h);

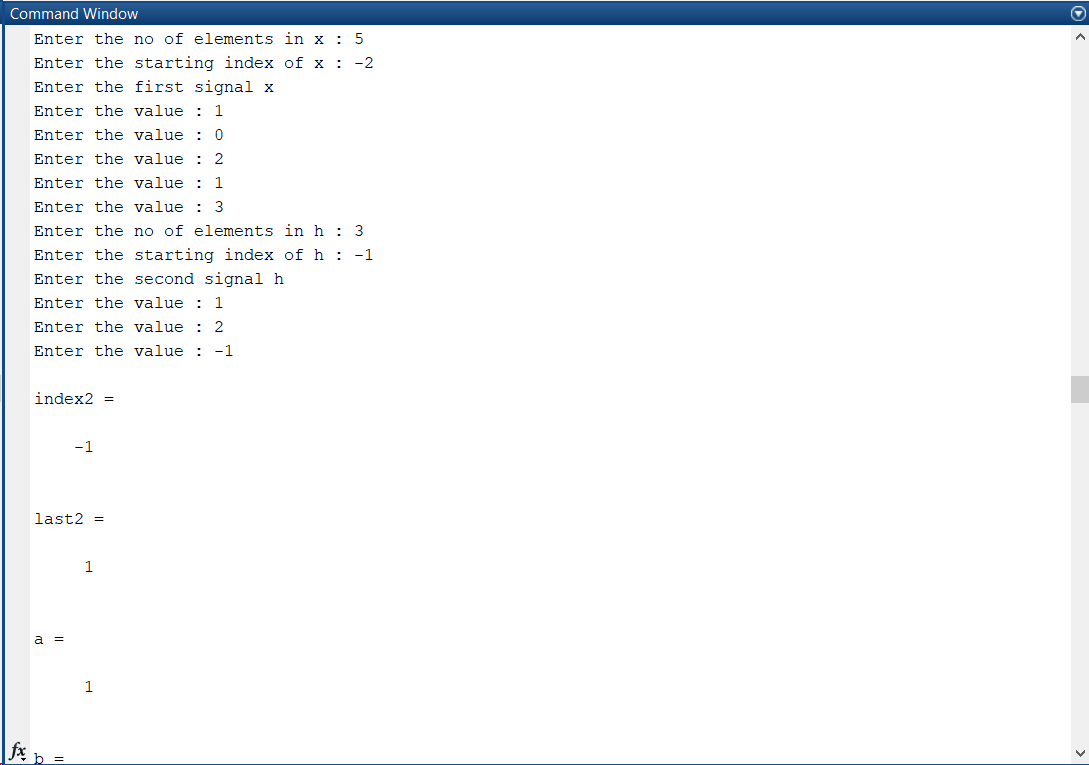
title("h(n)");

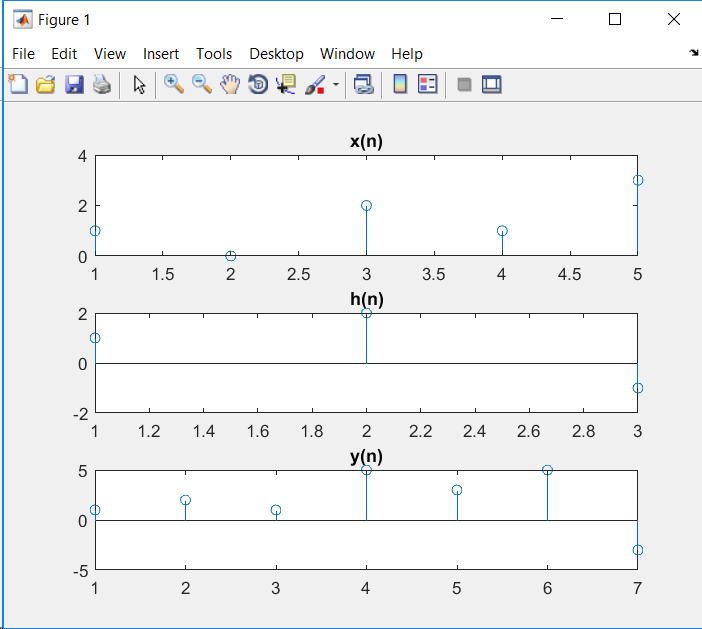
subplot(3, 1, 3);

stem(y);

title("y(n)");

OUTPUT





**Circular Convolution:**

X=input('Enter input signal array: ');

H=input('Enter impulse response array: ');

N1=length(X);

N2=length(H);

subplot(2,2,1);

stem(0:N1-1,X)

title('x(n)')

subplot(2,2,2);

stem(0:N2-1,H)

title('h(n)')

disp('Circular Convolution:')

if(N1>N2)

tempH=[H,zeros(1,N1-N2)]';

tempX=X

else

tempH=H;

tempX=[X,zeros(1,N2-N1)]';

end

while(rem(length(tempH),4)~=0)

tempH=[tempH,0];

end

while(rem(length(tempX),4)~=0)

tempX=[tempX,0];

end

tempMat=[];

tempN=N2;

for i=1:tempN

tempMat=[tempMat;circshift(tempH,[0,i-1])];

end

tempMat=tempMat';

% disp(tempMat)

Y=(tempMat\*tempX)';

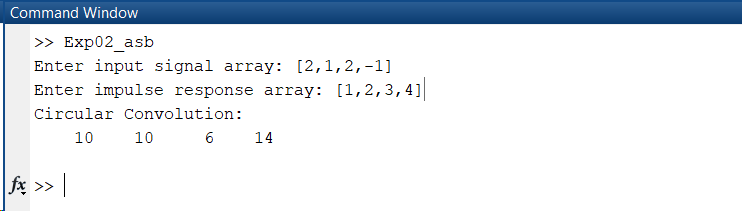
disp(Y)

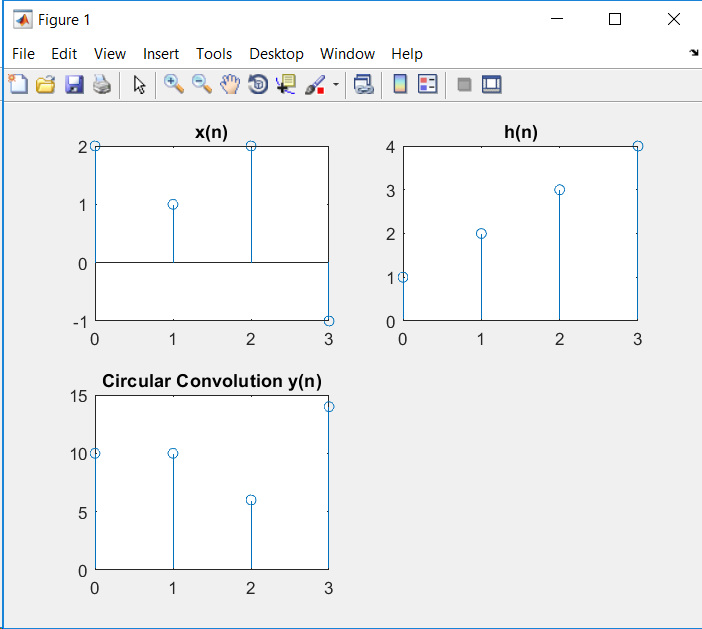
subplot(2,2,3);

stem(0:length(Y)-1,Y)

title('Circular Convolution y(n)')

OUTPUT





**Conclusion:-** Thus we have successfully computed linear and circular convolution of two discrete time sequences using Matlab.

**Date: \_25/02/2019\_ Signature of faculty in-charge**

**Post Lab Descriptive Questions**

* 1. Explain the role of convolution in signal processing.

Ans.

Convolution of signals occurs in several contexts in signal processing. The input-

output characteristic of linear time-invariant (LTI) systems, the most widely used type

of system in signal processing, is described entirely in terms of the impulse response of

the system. The impulse response is the output of the system due to an impulse input

signal. Given the input signal, the output of an LTI system is the convolution of the

input signal with the impulse response of the system. Hence, convolution plays a key

role in relating the input and output signals of an LTI system.

Convolution also arises when we analyze the effect of multiplying two signals in the

frequency domain. If we multiply two signals in time, the Fourier representation for

the product is the convolution of the Fourier representations of the individual signals.

This type of analysis occurs when the discrete Fourier transform is applied to truncated

(finite length) signals, and in the design of finite impulse response (FIR) filters.

* 1. Explain the difference between linear and circular convolution?

Ans.

Linear convolution is the basic operation to calculate the output for any linear time invariant system given its input and its impulse response.

Circular convolution is the same thing but considering that the support of the signal is periodic (as in a circle, hence the name).

Most often it is considered because it is a mathematical consequence of the discrete

Fourier transform (or discrete Fourier series to be precise):

One of the most efficient ways to implement convolution is by doing multiplication in

the frequency.

Sampling in the frequency requires periodicity in the time domain.

However, due to the mathematical properties of the FFT this results in circular

convolution.

The method needs to be properly modified so that linear convolution can be done (e.g.

overlap-add method).

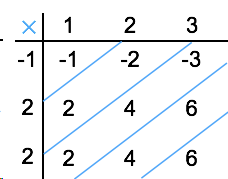
Therefore , If input signal is not periodic Linear convolution else Circular convolution.

* 1. Explain with the help of an example the steps required to transform linear convolution with circular convolution and vice-versa.

Ans.

**Linear to Circular Convolution**

convolute two sequences x[n] = {1,2,3} & h[n] = {-1,2,2} using circular convolution



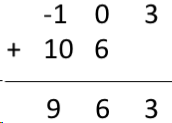
**Linear Convolution**

Normal Convoluted output y[n] = [ -1, -2+2, -3+4+2, 6+4, 6].

= [-1, 0, 3, 10, 6]

Here x[n] contains 3 samples and h[n] also has 3 samples. Hence the resulting sequence obtained by circular convolution must have max[3,3]= 3 samples.

Now to get periodic convolution result, 1st 3 samples [as the period is 3] of normal convolution is same next two samples are added to 1st samples as shown below:



∴ **Circular convolution result *y*[*n*]=[9 6 3]**

**Circular to linear Convolution**

x1(n) = {1, 2, 3, 4}

and x2 (n) = {1, 2, 1, 2}

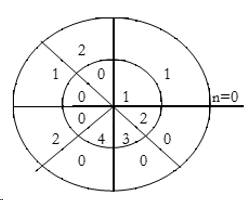
L=4, M=4

Length of y(n) = L+M-1=4+4-1=7

∴,x1(n) = {1, 2, 3, 4, 0, 0, 0}

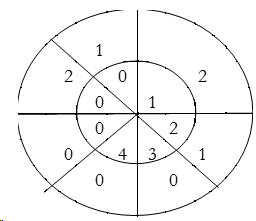
& x2(n) = {1, 2, 1, 2, 0, 0, 0}

For y(0),



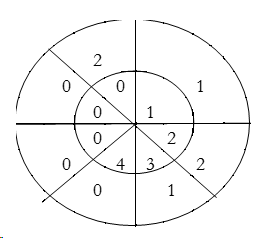
∴∴, y(0)= 1×1=1

For y(1),



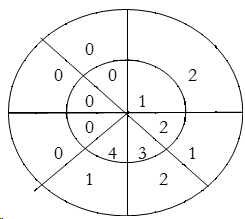
∴∴, y(1)= 2×1+1×2=4

For y(2),



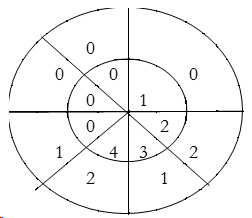
∴∴ , y(2)= 1×1+2×2+3×1=8

For y(3),



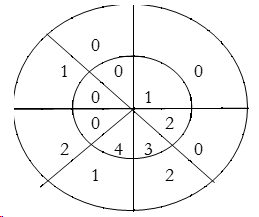
y(3)=1×2+2×1+3×2+4×1=14

For y(4),



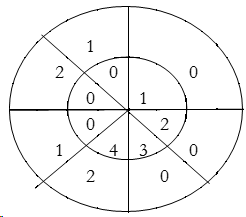
∴∴, y(4)= 4×2+3×1+2×2=15

For y(5),



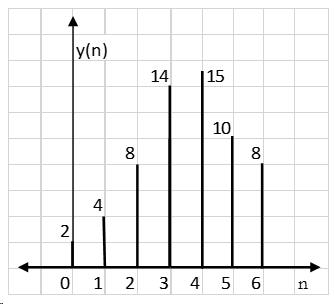
∴∴, y(5) = 4×1+3×2=10

For y(6),



∴∴, y(6) = 4×2=8

∴∴ ,y(n) = {1, 4, 8, 14, 15, 10, 8}



Result: y(n) = {2, 4, 8, 14, 15, 10, 8}