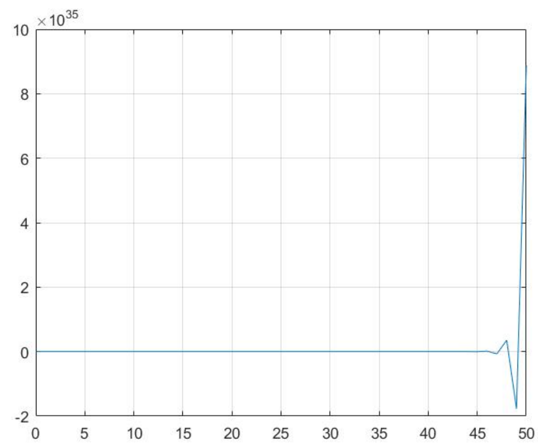
Signal Processing

1. Understanding properties of Discrete Time Sinusoidal signals
   (a) Plot the discrete time real sinusoidal signal xn=10β! for positive C when,
   (i) β<−1

%code

```
syms B;
B = -5;
% B = -0.5;
% B = 0.5;
% B = 5;
n = 0:1:10;

x = 10*B.^n;
stem(n, x);
grid;
```

%output

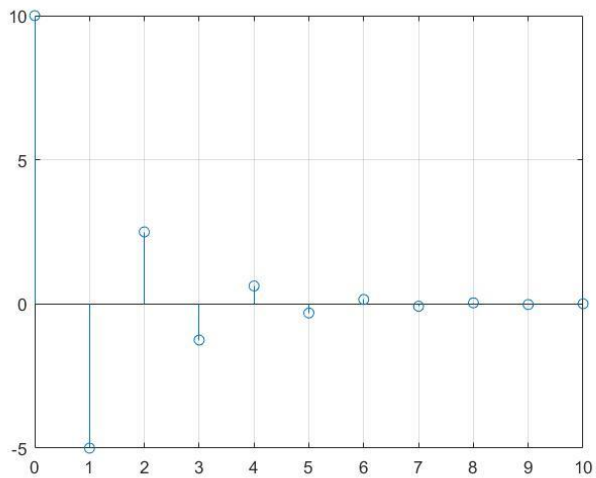


   (ii) −1<β<0

%code

```
syms B;
% B = -5;
B = -0.5;
% B = 0.5;
% B = 5;
n = 0:1:10;

x = 10*B.^n;
stem(n, x);
grid;
```

%output

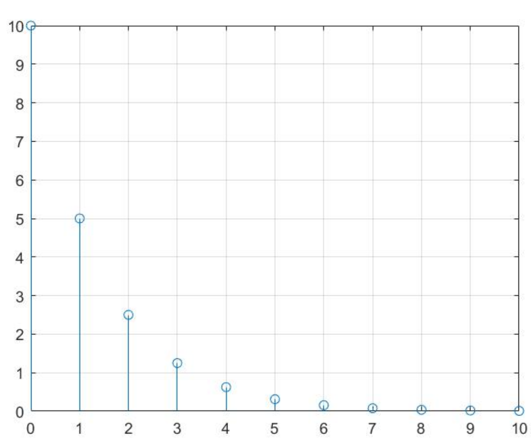


(iii) 0<β<1

%code

```
syms B;
% B = -5;
% B = -0.5;
B = 0.5;
% B = 5;
n = 0:1:10;

x = 10*B.^n;
stem(n, x);
grid;
```

%output



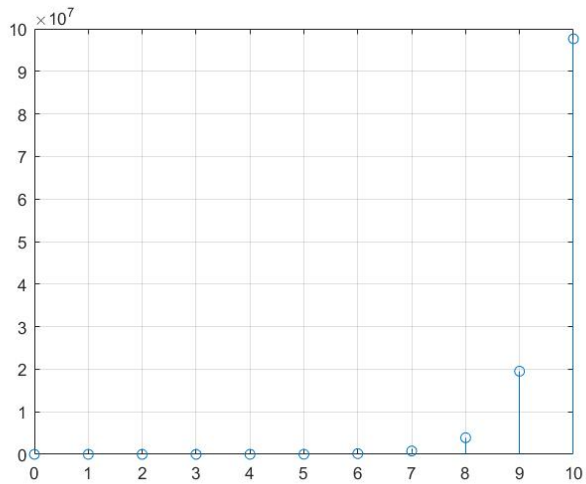
(iv) β>1

%code

```
syms B;
```

```
% B = -5;
% B = -0.5;
% B = 0.5;
B = 5;
n = 0:1:10;

x = 10*B.^n;
stem(n, x);
grid;
```

%output



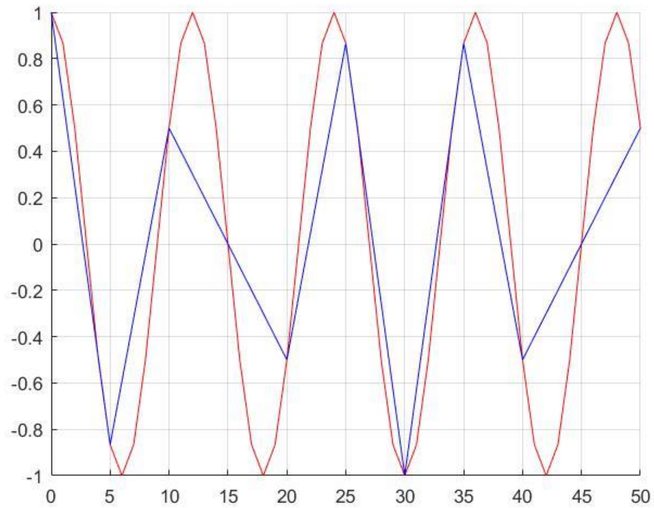
(b) Plot xn and $x(t)$ in the same plot for the following sinusoidal signals.Let n=kT where T=5s and k∈Z. That is $x[n]$ is obtained by sampling $x[t]$ at every 5 seconds. Determine the theoretical fundamental period of each signal

(i) $\mathbf{X[n] = \cos\left(\frac{2\pi n}{12}\right)}$, $\boldsymbol{x[t] = \cos(\frac{2\pi t}{12})}$

%code

```
syms n t1 t2;

t1 = 0:1:50;
n = 0:5:50;


xn = cos((2*pi*n)/12);
xt = cos((2*pi*t1)/12);

% xn = cos((8*pi*n)/31);
% xt = cos((8*pi*t1)/31);

hold on;
plot(t1, xt, 'r');
plot (n, xn, 'b');
hold off
grid on;

[~, locs] = findpeaks(xt);
mean(diff(locs)*0.1)

[~, locs] = findpeaks(xn);
mean(diff(locs)*0.1)
```

%output



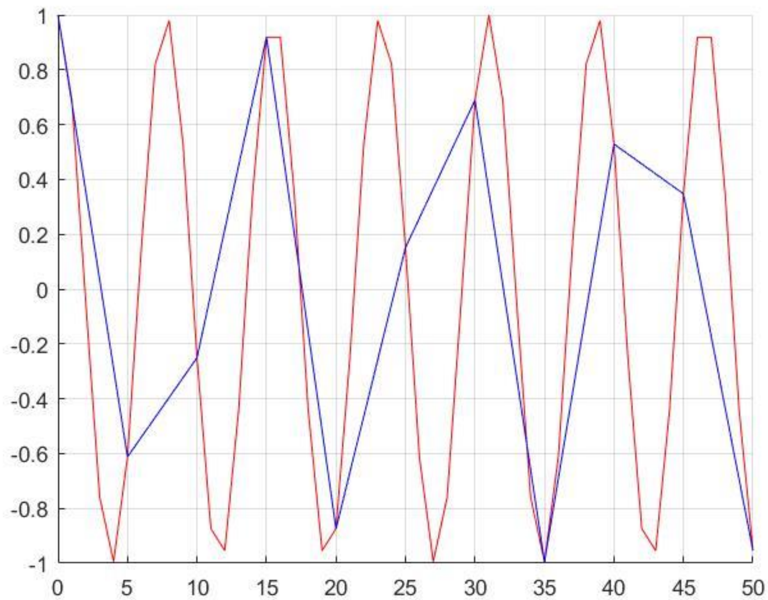
Theoretical fundamental period of xn = 0.2500
Theoretical fundamental period of xt = 1.2000

(ii) $\mathbf{X[n] = \cos\left(\frac{8\pi n}{31}\right),\ \ x[t] = \cos\left(\frac{8\pi t}{31}\right)}$

%code

```
syms n t1 t2;

t1 = 0:1:50;
n = 0:5:50;


% xn = cos((2*pi*n)/12);
% xt = cos((2*pi*t1)/12);

xn = cos((8*pi*n)/31);
xt = cos((8*pi*t1)/31);

hold on;
plot(t1, xt, 'r');
plot (n, xn, 'b');
hold off
grid on;

[~, locs] = findpeaks(xt);
mean(diff(locs)*0.1)

[~, locs] = findpeaks(xn);
mean(diff(locs)*0.1)
```

%output



Theoretical fundamental period of xn = 0.2500
Theoretical fundamental period of xt = 0.7800

Is the observed period of the signal from the plot always equal to the theoretical period?

No it doesn't, it varies with the time

(c) Plot the following nine discrete time signals in the same graph

(i) $\mathbf{X[n] = \cos(0.\boldsymbol{n})}$

(ii) $\mathbf{X[n] = \cos\left(\frac{\pi n}{8}\right)}$

(iii) $\mathbf{X[n] = \cos\left(\frac{\pi n}{4}\right)}$

(iv) $\mathbf{X[n] = \cos\left(\frac{\pi n}{2}\right)}$

(v) $\mathbf{X[n] = \cos(\pi n)}$

(vi) $\mathbf{X[n] = \cos\left(\frac{3\pi n}{2}\right)}$

(vii) $\mathbf{X[n] = \cos\left(\frac{7\pi n}{4}\right)}$

(viii) $\mathbf{X[n] = \cos\left(\frac{15\pi n}{8}\right)}$

(ix) $\mathbf{X[n] = \cos(2\pi n)}$

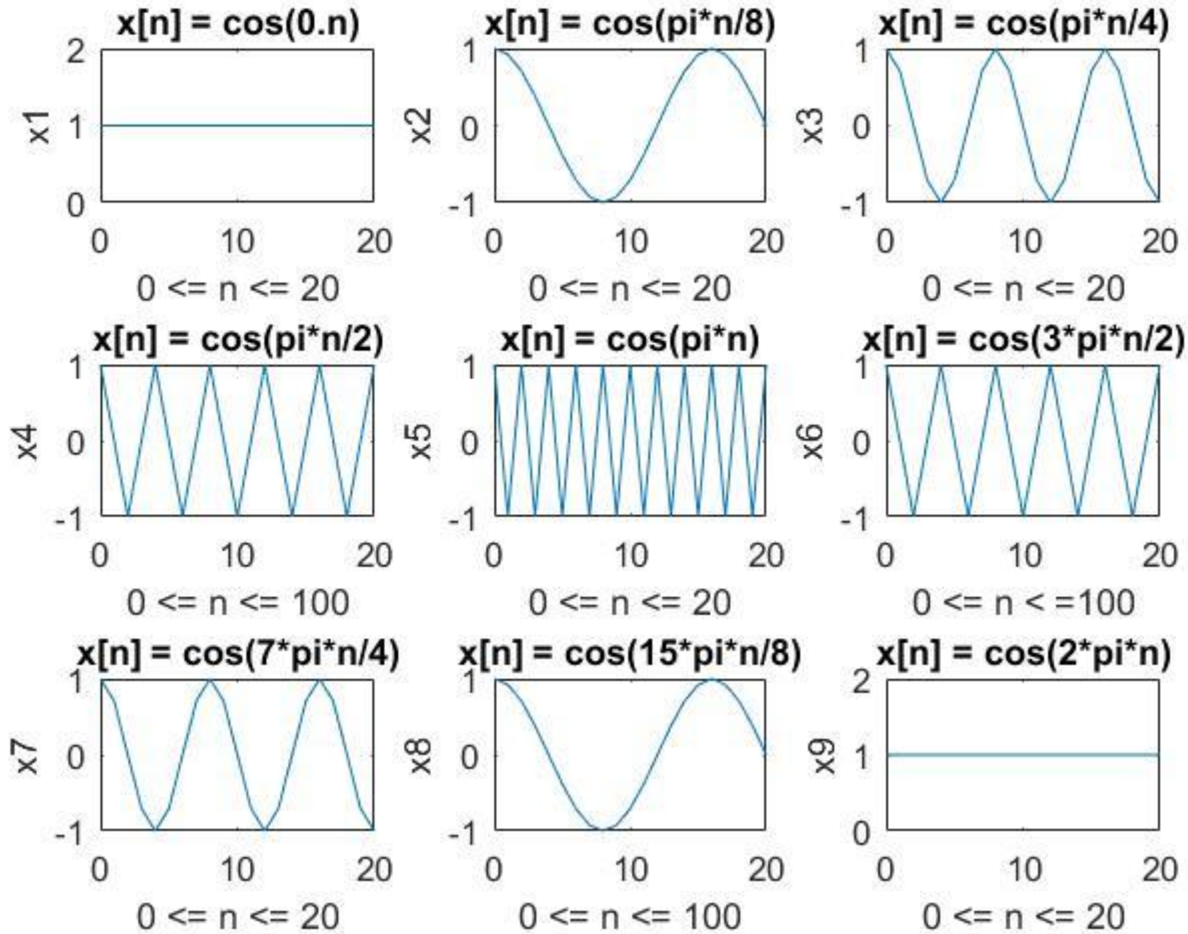
%code

```
syms n;

n = 0:1:20;
```

```
x1 = cos(0*n);
x2 = cos(pi*n/8);
x3 = cos(pi*n/4);
x4 = cos(pi*n/2);
x5 = cos(pi*n);
x6 = cos(3*pi*n/2);
x7 = cos(7*pi*n/4);
x8 = cos(15*pi*n/8);
x9 = cos(2*pi*n);

subplot(3,3,1);
plot(n, x1);
title('x[n] = cos(0.n)');
xlabel('0 <= n <= 20');
ylabel('x1');

subplot(3,3,2);
plot(n, x2);
title('x[n] = cos(pi*n/8)');
xlabel('0 <= n <= 20');
ylabel('x2');

subplot(3,3,3);
plot(n, x3);
title('x[n] = cos(pi*n/4)');
xlabel('0 <= n <= 20');
ylabel('x3');

subplot(3,3,4);
plot(n, x4);
title('x[n] = cos(pi*n/2)');
xlabel('0 <= n <= 100');
ylabel('x4');

subplot(3,3,5);
plot(n, x5);
title('x[n] = cos(pi*n)');
xlabel('0 <= n <= 20');
ylabel('x5');

subplot(3,3,6);
plot(n, x6);
title('x[n] = cos(3*pi*n/2)');
xlabel('0 <= n < =100');
ylabel('x6');

subplot(3,3,7);
plot(n, x7);
title('x[n] = cos(7*pi*n/4)');
xlabel('0 <= n <= 20');
ylabel('x7');

subplot(3,3,8);
plot(n, x8);
title('x[n] = cos(15*pi*n/8)');
xlabel('0 <= n <= 100');
ylabel('x8');
```

```
subplot(3,3,9);
plot(n, x9);
title('x[n] = cos(2*pi*n)');
xlabel('0 <= n <= 21');
ylabel('x9');
```

%output



(d) By observing the plots, you have obtained in question 1.c, what can you tell about the shape of the signal as discrete frequency is varied?

As the frequency increased the circularity of the graph also increased, more like becoming a continuous signal.

2) Discrete convolution

a) write a matlab function to implement discrete convolution for n>0. Note that yn=xn∗hn is given by the convolution summation yn= xkh[n−k]

%code

```
function y = convolution(x,h)
    m=length(x);
    n=length(h);
    X=[x,zeros(1,n)];
    H=[h,zeros(1,m)];
    for i=1:n+m-1
        y(i)=0;
        for j=1:m
            if(i-j+1>0)
                y(i)=y(i)+X(j)*H(i-j+1);
            else
            end
        end
    end
end
```

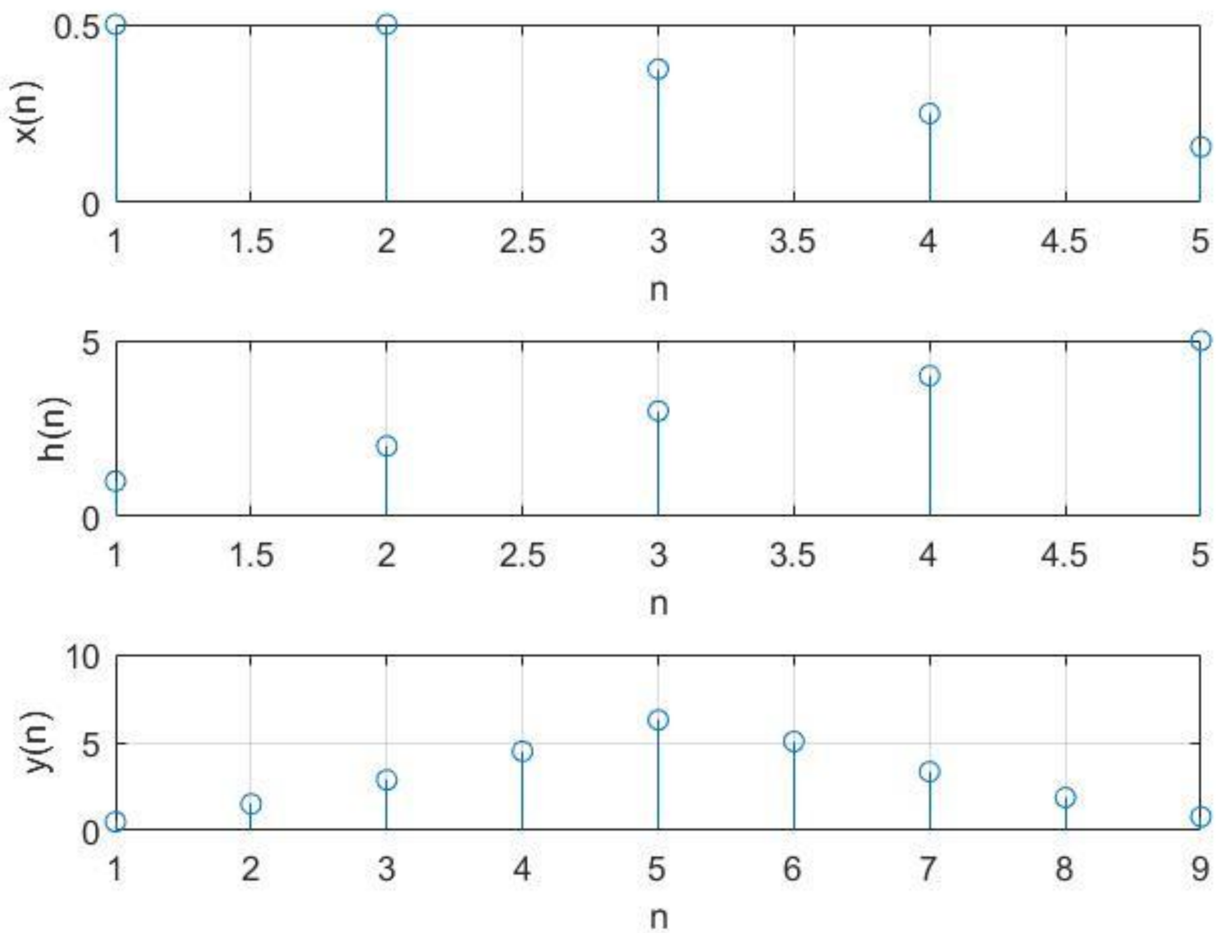
b) Using the function written in section a, convolue x[n] = 0.5^{n}u(n) with h[n] = u[n]. Plot the output signal along with the two input signals.

%code

```
n = [1,2,3,4,5];
un = [1,2,3,4,5];
hn = un;
xn = 0.5 .^n .* un;

y = convolution(x,h);

subplot(3,1,1);
stem(xn);
grid
xlabel( 'n' ) ;
ylabel( 'x(n)' ) ;

subplot(3,1,2);
stem(hn);
grid
xlabel( 'n' ) ;
ylabel( 'h(n)' ) ;

subplot(3,1,3);
stem(y);
grid
xlabel( 'n' ) ;
ylabel( 'y(n)' ) ;
```

```
function y = convolution(xn,hn)
    m=length(x);
    n=length(h);
    X=[x,zeros(1,n)];
    H=[h,zeros(1,m)];
    for i=1:n+m-1
        y(i)=0;
        for j=1:m
            if(i-j+1>0)
                y(i)=y(i)+X(j)*H(i-j+1);
            else
            end
        end
    end
end
```

%output



Consider the following two signals x[n] = [1 1 1 1 1 0 0 0 0 0 0 0 0 0 0], h[n] = [2 4 8 16 32 64 0 0 0 0 0 0 0 0 0]

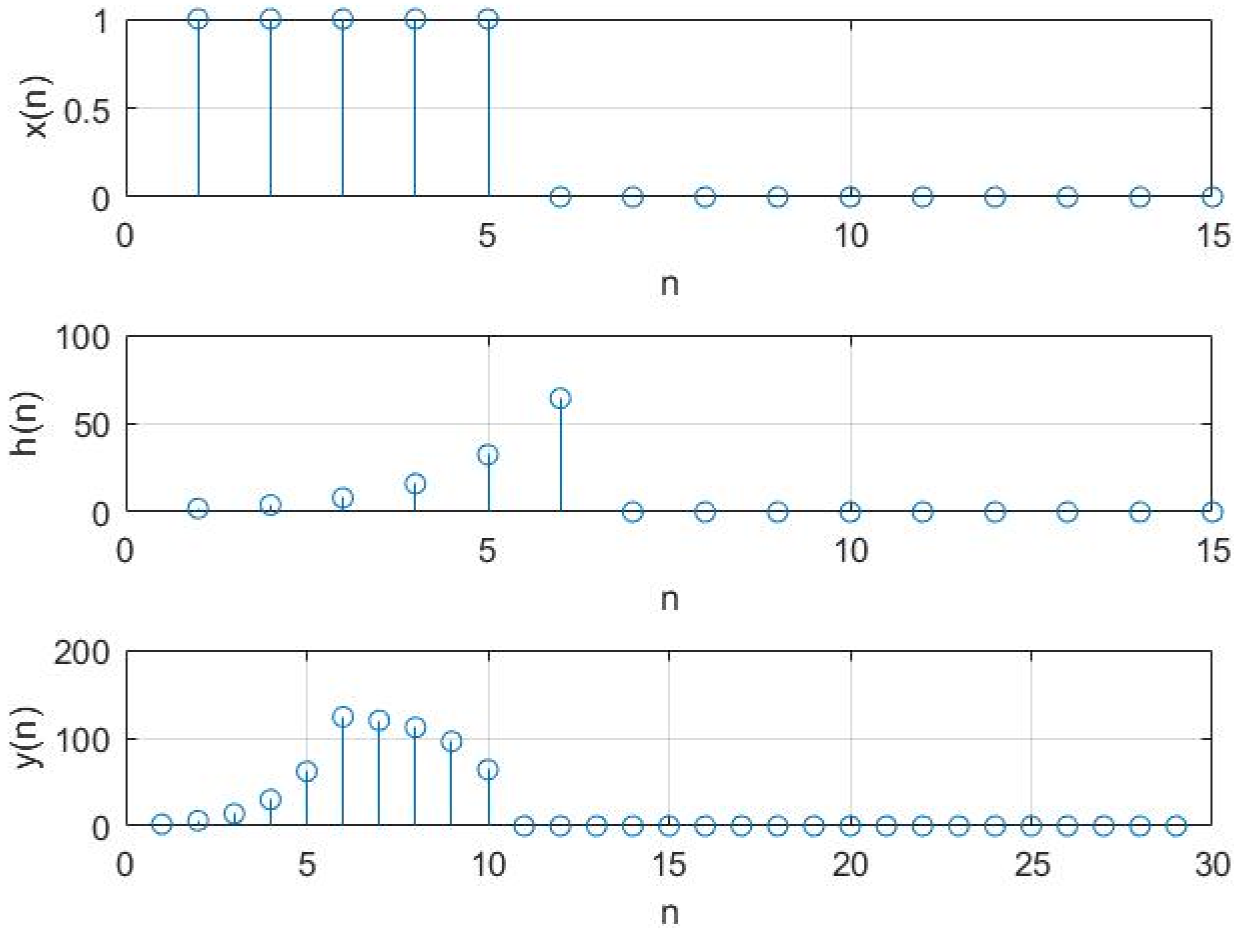
i) Convolue the two signals using the function written in part a. Use matlab conv command to verify your answer

%code

```
xn = [1,1,1,1,1,0,0,0,0,0,0,0,0,0,0];
hn = [2,4,8,16,32,64,0,0,0,0,0,0,0,0,0];

y = convolution(xn,hn)

y_1 = conv(xn,hn)

function y = convolution(x,h)
    m=length(x);
    n=length(h);
    X=[x,zeros(1,n)];
    H=[h,zeros(1,m)];
    for i=1:n+m-1
        y(i)=0;
        for j=1:m
            if(i-j+1>0)
                y(i)=y(i)+X(j)*H(i-j+1);
            else
            end
        end
    end
end
```

```
%output
y =

  Columns 1 through 10

     2     6    14    30    62   124   120   112    96    64

  Columns 11 through 20

     0     0     0     0     0     0     0     0     0     0

  Columns 21 through 29

     0     0     0     0     0     0     0     0     0


y_1 =

  Columns 1 through 10

     2     6    14    30    62   124   120   112    96    64

  Columns 11 through 20

     0     0     0     0     0     0     0     0     0     0

  Columns 21 through 29

     0     0     0     0     0     0     0     0     0
```

ii) **Consider the shape of the signal h[n] and the output signal, what sort of a transformation has been applied through the convolution operation?**



3) LTI Systems

a) Consider the following processes. Identify input x[n] and output y[n] for each case. Implement a matlab function to implement the given system.

i) An investor is maintaining a bank account. The bank pays him a monthly interest of 1%. It is given that the net savings he makes is P. Write a function to calculate his current bank balance B in terms of B and P.

```
function   B = interest(P)
    B = P + P /100;
end
```

ii) A merchant earns M amount of money monthly. He spends half of it and retains the rest of its as savings. Write a function to calculate the amount of money he has as savings

```
function   S = saving_balance(M)

    S = M / 2;

end
```

b) Find the impulse response of the above two LTI systems.

i)

```
P = 100000;
B = interest(P)
x = conv(B,P)
stem(x)

function   B = interest(P)

    B = P + P /100;

End
```

ii)

```
M = 100000;
S = saving_balance(M)
x = conv(S,M)
stem(x)

function   S = saving_balance(M)

    S = M / 2;

end
```

c) Based on the results obtains at part b, classify two LTI systems into IIR or FIR

The two LTI systems have finite impulse response. If the impulse response of the system is finite, it’s a FIR system (Finite Impulse Response). But if the impulse response is infinite it’s a IIR system (Infinite Impulse Response). Therefore both the LTI systems are FIR systems.