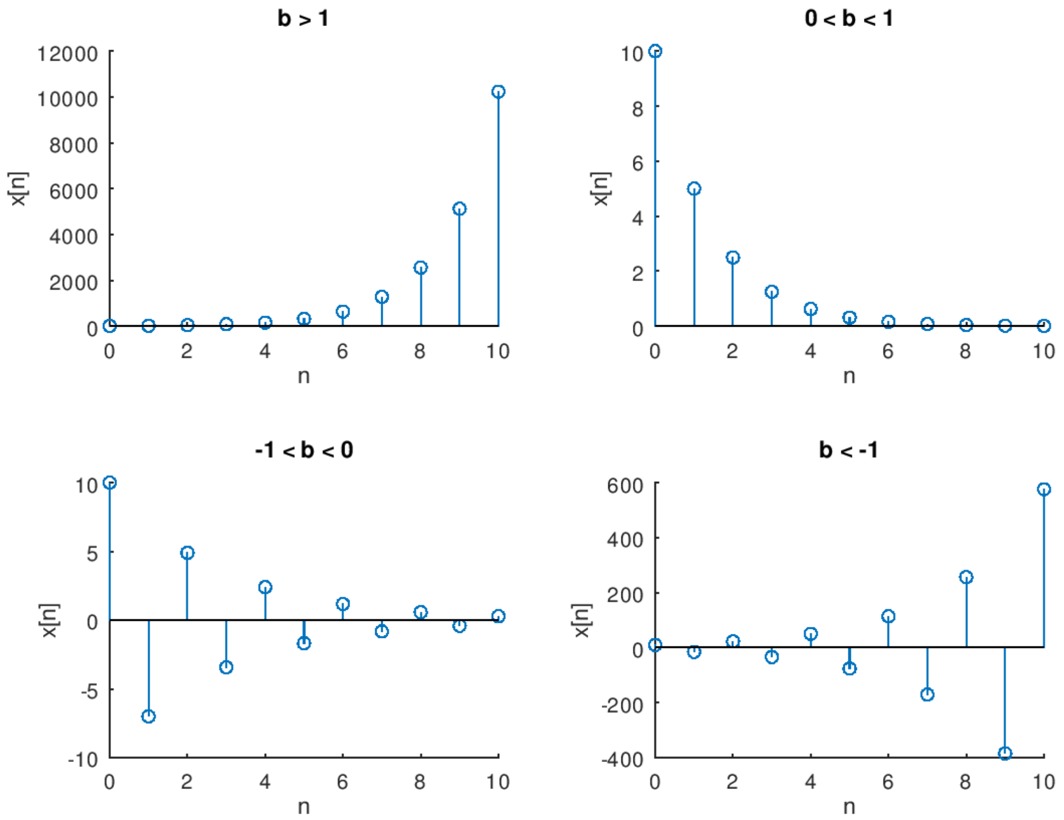
Signal Processing

Laboratory on Discrete Time Signals

1) **Understanding properties of Discrete Time Sinusoidal signals**
   a) **. Plot the discrete time real sinusoidal signal x[n] = 10β^n for positive C when,**
      i) **β <-1**
      ii) **-1< β<0**
      iii) **0< β <1**
      iv) **β >1**

```
n=[0:10];

% β <-1

b=2

x1=10.*(b.^n);

subplot(2,2,1);

stem(n,x1);xlabel('n');ylabel('x[n]');title('b > 1');

%-1< β<0

b=0.5;

x2=10*(b.^n);

subplot(2,2,2);

stem(n,x2);xlabel('n');ylabel('x[n]');title('0 < b < 1');

%0< β <1

b=-0.7;

x3=10*(b.^n);

subplot(2,2,3);

stem(n,x3);xlabel('n');ylabel('x[n]');title('-1 < b < 0');
```

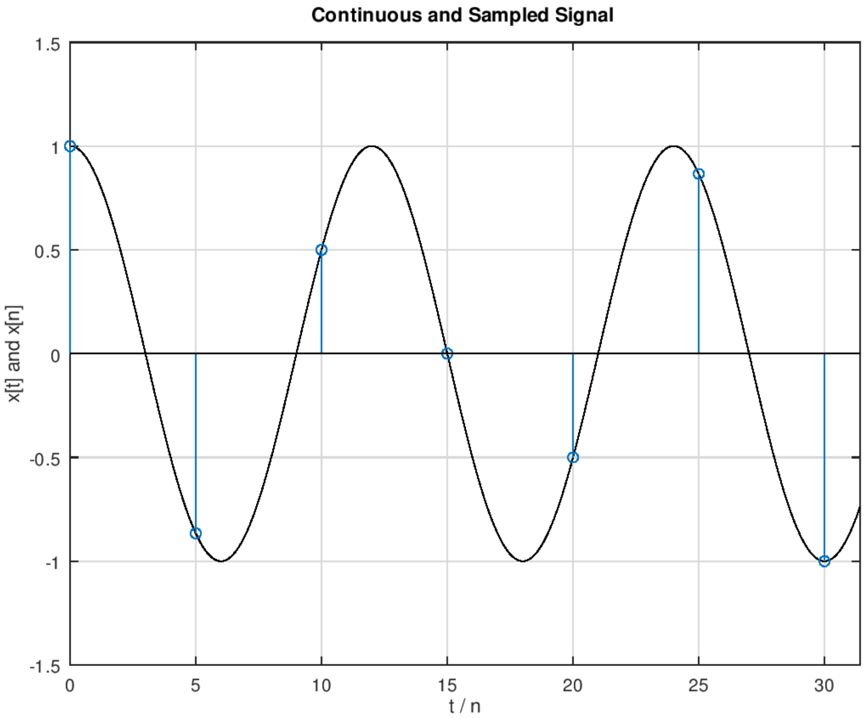
```
b=-1.5;

x4=10*(b.^n);

subplot(2,2,4);

stem(n,x4);xlabel('n');ylabel('x[n]');title('b < -1');
```



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

i)

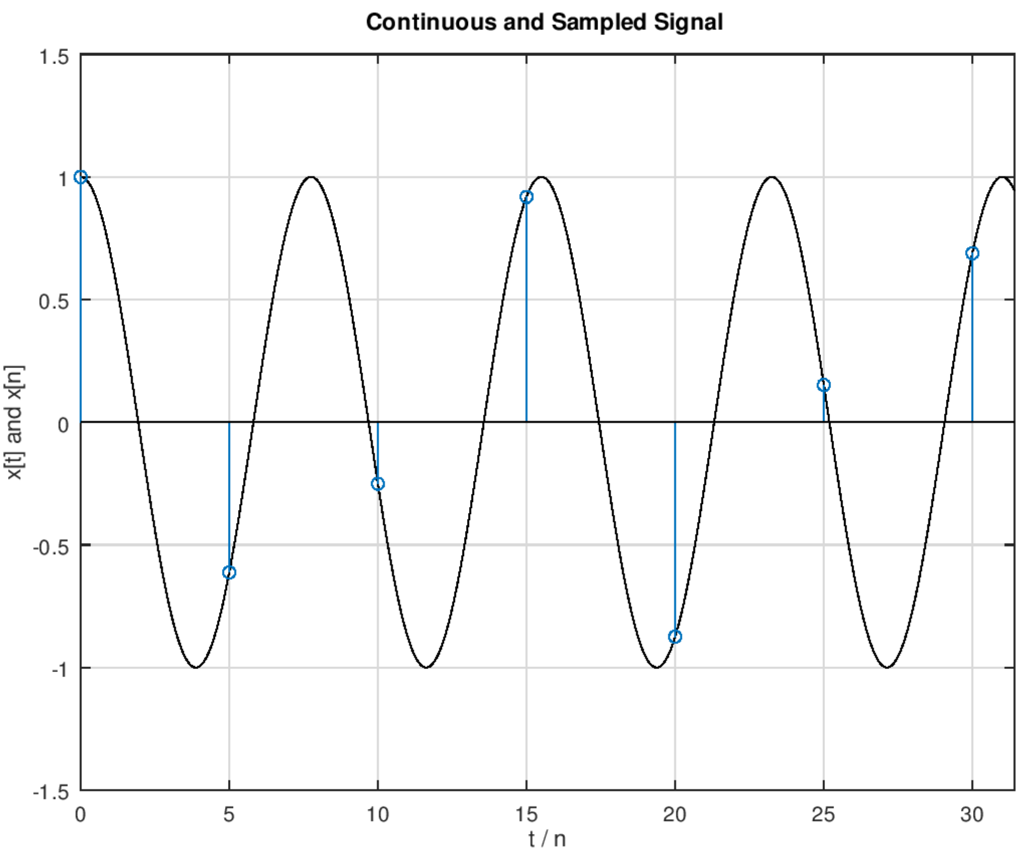
```
T=5;

t=[0:0.01:10*pi];

x_t = cos((2*pi.*t)/12);


%continuous signal

plot(t,x_t,'k');

axis([0 10*pi -1.5 1.5]);


%sample the signal

nmin = ceil(min(t)/ T);

nmax = floor(max(t) / T);

n =[nmin:nmax];

grid

hold on


x_n=cos((2*pi.*n*T)/12);

stem(n*T,x_n)

axis([0 10*pi -1.5 1.5]);

xlabel('t / n');ylabel('x[t] and x[n]');title('Continuous and Sampled Signal ');
```



ii)

```
T=5;
t=[0:0.01:10*pi];
x_t = cos((8*pi.*t)/31);
plot(t,x_t,'k');
axis([0 10*pi -1.5 1.5]);

%sample the signal
nmin = ceil(min(t)/ T);
nmax = floor(max(t) / T);
n =[nmin:nmax];
grid
hold on
```

```
x_n=cos((8*pi.*n*T)/31);

stem(n*T,x_n)

axis([0 10*pi -1.5 1.5]);

xlabel('t / n');ylabel('x[t] and x[n]');title('Continuous and Sampled Signal ');
```



If the signal is periodic $x(n)=x(n+N)$

Then, $x[n] = \cos(\Omega_0 n)$ → eq1

$X[n+N] = \cos(\Omega_0(n+N))$

$= \cos(\Omega_0 n+ \Omega_0 N)$ → eq2

To satisfy above condition,

$\Omega_0 N = 2\pi m$ where m = 1,2,3.....

N must be an integer

i) General equation for Continuous time sinusoidal is

$x(t)=A\cos(w_0 t)$ where $w_0 = \frac{2\pi}{T}$

Here $x(t) = \cos(\frac{2\pi t}{12})$

Therefore, $w_0 = \frac{2\pi}{12}$

So the theoretical fundamental period (T) = 12 s

$\frac{2\pi N}{12} = 2\pi$

Therefore N=12

So this is a periodic signal.

And the observed period of the signal from the plot always equal to the theoretical period.

ii) General equation for Continuous time sinusoidal is

$x(t)=A\cos(w_0 t)$ where $w_0 = \frac{2\pi}{T}$

Here $x(t) = \cos(\frac{8\pi t}{31})$

Therefore, $w_0 = \frac{8\pi}{31}$

So the theoretical fundamental period (T) = $\frac{31}{4}$ s

$\frac{8\pi N}{31} = 2\pi$

Therefore $N=\frac{31}{4}$.

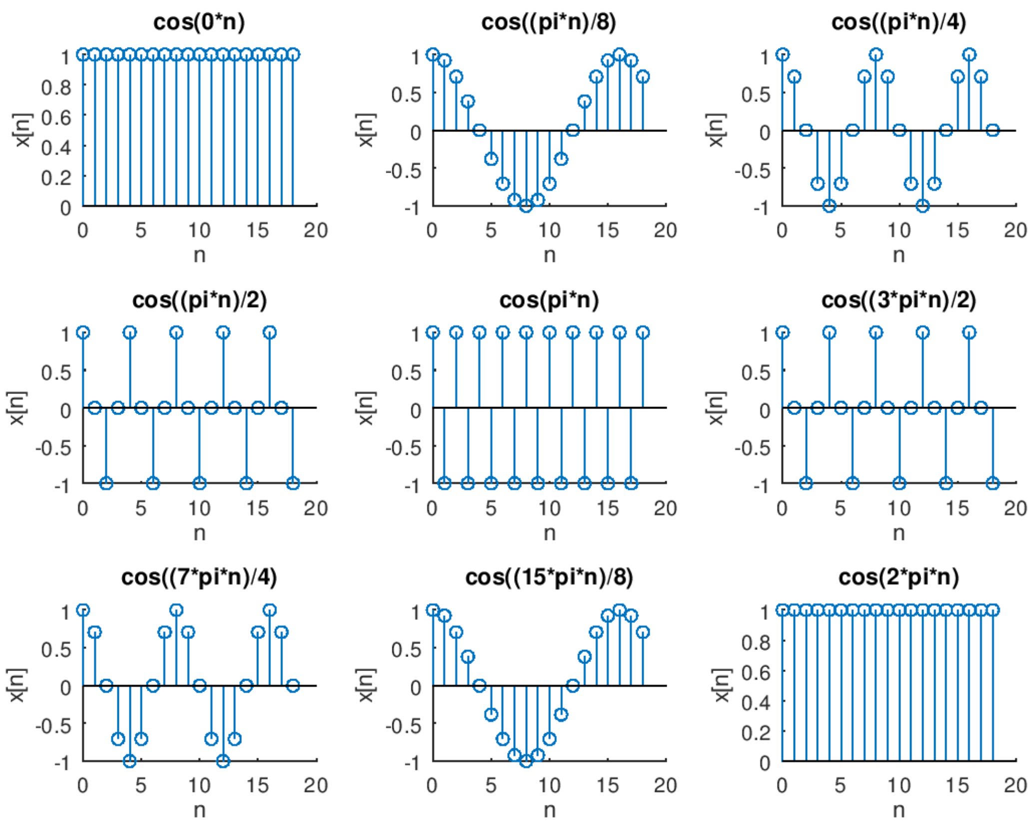
So this is not a periodic signal.

And the observed period of the signal from the plot always not equal to the theoretical period.

c) Plot the following nine discrete time signals in the same graph (use subplot command).

```
n=0:6*pi
x_1=cos(0.*n);
x_2=cos((pi.*n)/8);
x_3=cos((pi.*n)/4);
x_4=cos((pi.*n)/2);
x_5=cos(pi.*n);
x_6=cos((3*pi.*n)/2);
x_7=cos((7*pi.*n)/4);
x_8=cos((15*pi.*n)/8);
x_9=cos(2*pi.*n);
%plotting
subplot(3,3,1);
stem(n,x_1);xlabel('n');ylabel('x[n]');title('cos(0*n)');
subplot(3,3,2);
stem(n,x_2);xlabel('n');ylabel('x[n]');title('cos((pi*n)/8)');
subplot(3,3,3);
stem(n,x_3);xlabel('n');ylabel('x[n]');title('cos((pi*n)/4)');
subplot(3,3,4);
stem(n,x_4);xlabel('n');ylabel('x[n]');title('cos((pi*n)/2)');
subplot(3,3,5);
stem(n,x_5);xlabel('n');ylabel('x[n]');title('cos(pi*n)');
subplot(3,3,6);
stem(n,x_6);xlabel('n');ylabel('x[n]');title('cos((3*pi*n)/2)');
subplot(3,3,7);
stem(n,x_7);xlabel('n');ylabel('x[n]');title('cos((7*pi*n)/4)');
```

```
subplot(3,3,8);

stem(n,x_8);xlabel('n');ylabel('x[n]');title('cos((15*pi*n)/8)');

subplot(3,3,9);

stem(n,x_9);xlabel('n');ylabel('x[n]');title('cos(2*pi*n)');
```



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?

When discrete frequency is varied, the shape of the signal not always different. If the frequency is multiple of 2π the shape of the signal is same as the original signal.

$x[n] = A\cos(\Omega n)$

$x\_1[n] = A\cos((\Omega+2\pi)n) = A\cos(\Omega n + 2\pi n)$

Here shape of the x[n] and the x_1[n] signals are same since they satisfy the above condition.
Same shape signals are cos(0*n) and cos(2πn),
cos((pi*n)/8) and cos((15*pi*n)/8),
cos((pi*n)/4) and cos((7*pi*n)/4),
cos((pi*n)/2) and cos((3*pi*n)/2)

In these pairs they do not satisfy the condition (multiple of 2π) exactly. But the shapes are nearly same.(nearly satisfy the condition)

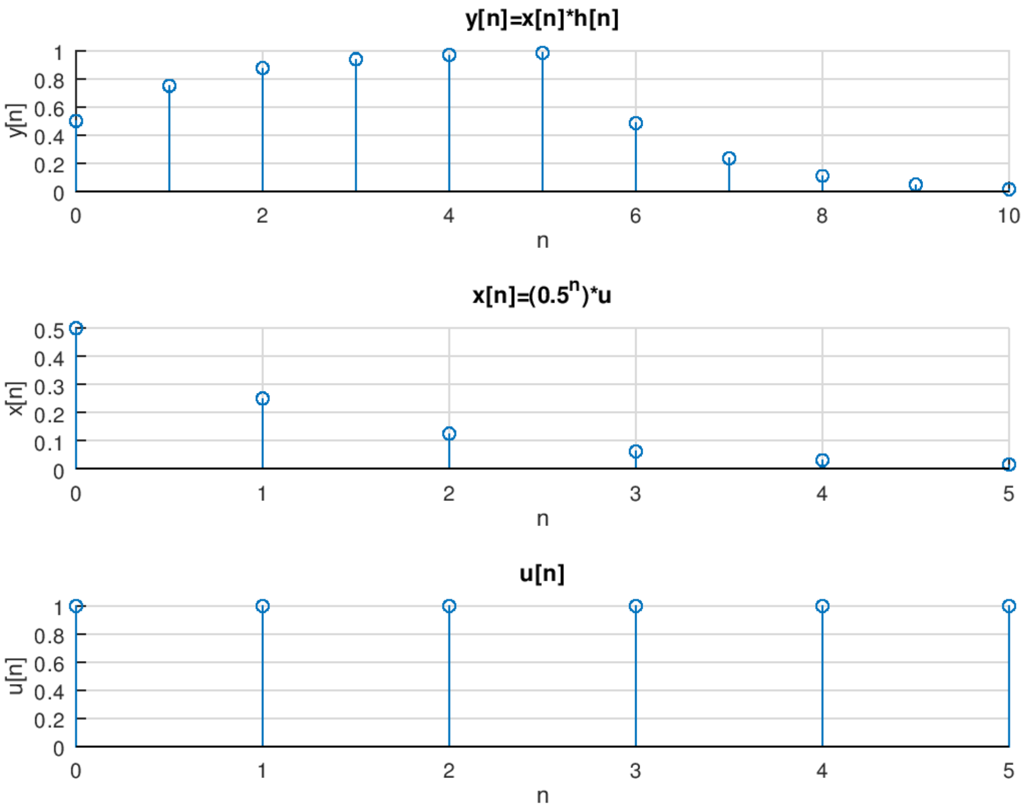
2) Discrete convolution

a) Write a matlab function to implement discrete convolution for n > 0.

```
function Y = Dconvolution(x,h)    %convolution function
 m=length(x);
 n=length(h);

 %fill with zeros
 X=[x,zeros(1,n)];
 H=[h,zeros(1,m)];

 %calculate convolution
 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
```

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

```
u= [1 1 1 1 1 1];
n=1:length(u);
x=(0.5.^n).*u;

%convolution
y=Dconvolution(x,u);

%plotting
subplot(3,1,1)
stem((0:length(y)-1),y);xlabel('n');ylabel('y[n]');title('y[n]=x[n]*h[n]');
grid

subplot(3,1,2)
stem((0:length(x)-1),x);xlabel('n');ylabel('x[n]');title('x[n]=(0.5^n)*u');
grid

subplot(3,1,3)
stem((0:length(u)-1),u);xlabel('n');ylabel('u[n]');title('u[n]');
grid
```



c)

```
x = [1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 ];

h = [2 4 8 16 32 64 0 0 0 0 0 0 0 0 0 0];

y = Dconvolution(x,h);
```

```
%>> y

%Columns 1 through 18:

 %  2    6   14   30   62  124  120  112   96   64    0    0    0    0    0    0    0    0

 % Columns 19 through 29:

  % 0    0    0    0    0    0    0    0    0    0    0
```

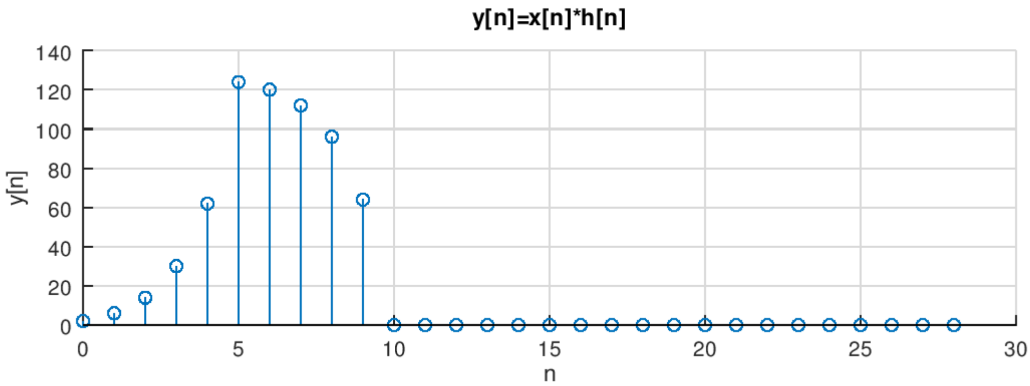
```
y1 = conv(x,h);
```

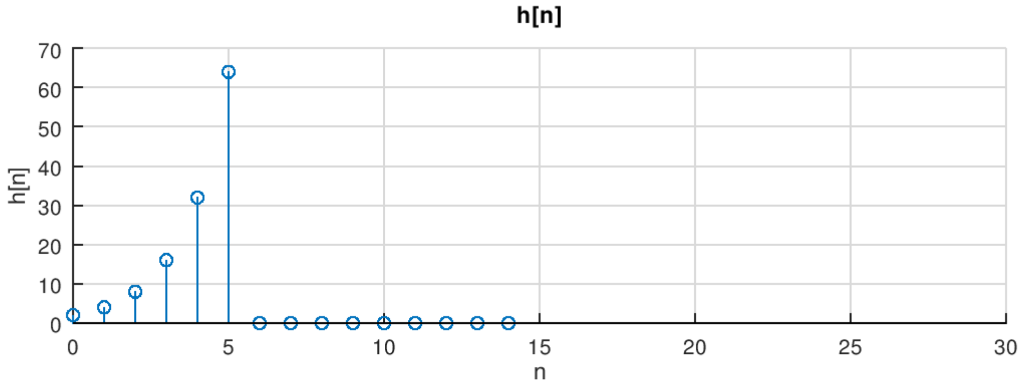
```
%>> y1
 %Columns 1 through 18:
 %   2    6   14   30   62  124  120  112   96   64    0    0    0    0    0    0    0    0
 % Columns 19 through 29:
 % 0    0    0    0    0    0    0    0    0    0    0
```

iii) Here the convolution result from the created function and the result which is obtained from matlab built-in function are same.

iv)

```
%plotting
subplot(2,1,1);
stem((0:length(y)-1),y);xlabel('n');ylabel('y[n]');title('y[n]=x[n]*h[n]');
grid

subplot(2,1,2);
stem((0:length(h)-1),h);xlabel('n');ylabel('h[n]');title('h[n]');
axis([0,30,0 70]);
grid
```





iv) A linear transformation has been applied through the convolution operation.

3) LTI systems

a)

i) Here ‘p’ is the net savings which the investor makes and assume that he save the money at the end of the month. ‘b’ is the balance at the end of the month.

In this case,

input (x[n]) is the net savings investor makes (p).

output (y[n]) is the bank balance at the end of the month n

Here the current bank balance is the addition of interest paid by the bank to the previous month balance amount, previous balance amount and net saving for the current month.

Y[n] = 1.01*y[n-1] + x[n]

```
function b = calBalance(p)

  y = [zeros(1,15)]%initialize the array

  y(1)=p(1);

  %calculate balance

  for i=2:length(p)

   y(i) = 1.01*y(i-1)+p(i); %calculate balanace

  endfor

  b=y(i);


  h=deconv(y,p) % finding impulse response

  y1=conv(p,h)

endfunction
```

ii) Here ‘m’ is the amount of money he earns and b is his savings at the end of the month**.**

In this case,

Input (x[n]) is amount of money he earns.

Output (y[n]) is amount of money he has saved at the end of the month n.

Output is calculated as the addition of 0.5*amount of he saves and the previous month blanace.

Y[n] = y[n-1] + x[n]*0.5

```
function b = calSaving(m)
  y = [zeros(1,20)] %initialize the array
  y(1)=m(1)/2;
  %calculate balance
  for i=2:length(m)
    y(i)=(m(i)*0.5) + y(i-1);
  endfor
  b=y(i);


  h=deconv(y,m)
  y1=conv(m,h)
endfunction
```

b) To find the impulse response h[n], when we know the y[n] and x[n], we use deconvolution. Impulse response of the first problem is

h[n] = [1.0000 1.0100 1.0201 1.0303 1.0406 1.0510….]

Impulse response of the second problem is

h[n] = [0.50000 0.50000 0.50000 0.50000 0.50000 0.50000..]

c) If the impulse response of a system is of finite length,then the system is said to be a Finite Impulse Response (FIR) system. The term finite impulse response arises because the output is computed as a weighted, finite term sum, of past, present, and perhaps future values of the filter input.
If the impulse response is not of finite length then such a system is called an Infinite Impulse Response (IIR) system.

In both systems the output depends on the inputs and the past outputs, therefore both systems are IIR systems.

1st system → IIR
2nd system → IIR