DSP LAB: EXPERIMENT-1

QUESTION A]:

Write a sample script to convolve following continuous time signals;
Plot x(t), h(t) and the convolved output y(t).

1) x(t) = u(t) - u(t-5) with h(t) = (sin(2.pi.t))/(2.pi.t)
Repeat the processing without flipping (in the four step process) by means of convolution to obtain g(t).
What is the effect of 'b' when we consider h(bt) in this process? WHEN;
i) 0<b<1; and
ii) 1<b<finite positive constant
Compare g(t) with y(t).

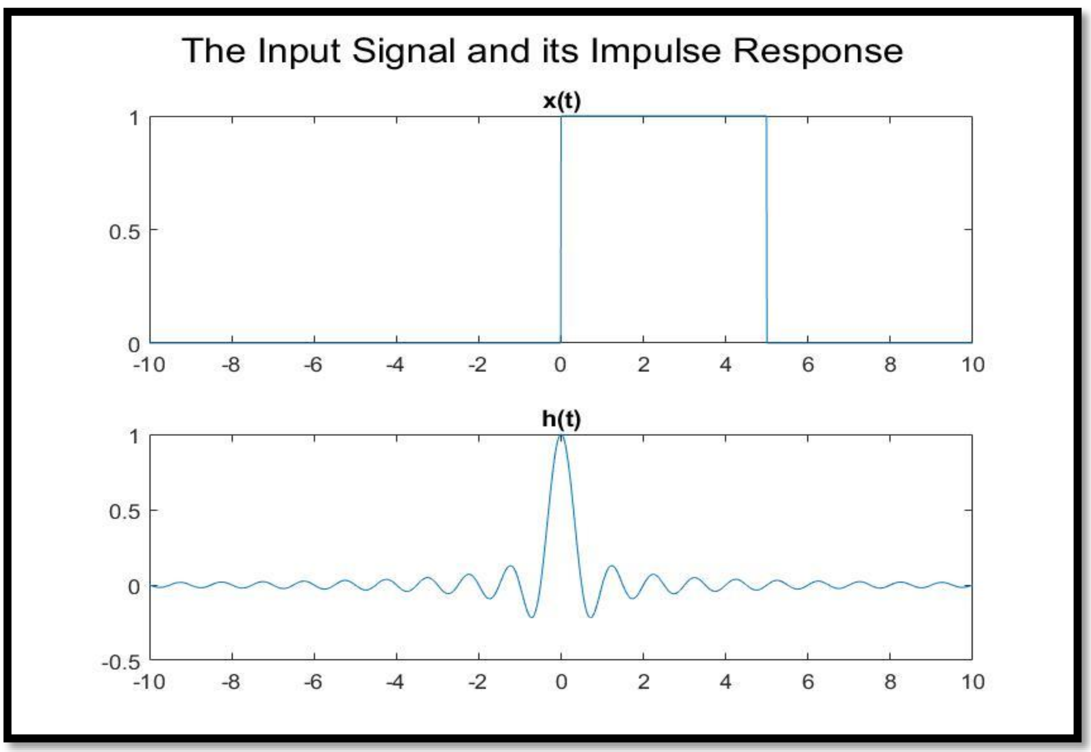
MATLAB CODE:

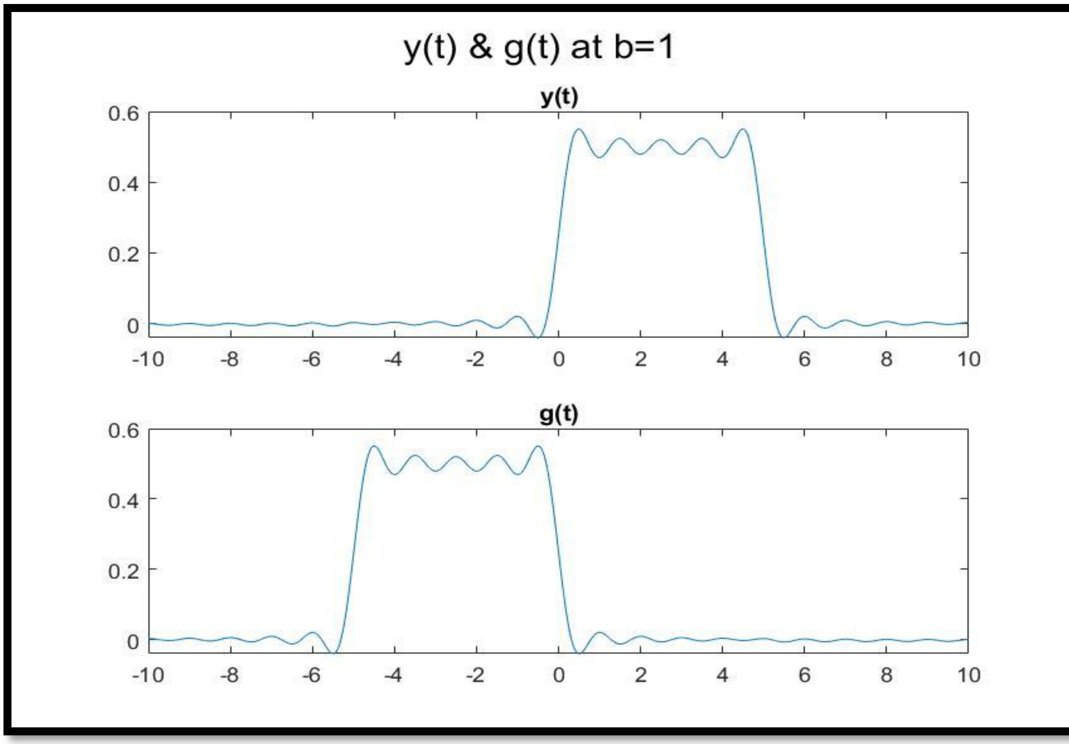
```
%[PART A]
%WE ARE SUPPOSED TO CONVOLVE
%x(t) = u(t) - u(t-5) WITH  h(t) = (sin(2.pi.t))/(2.pi.t)

clc
t=-10:0.01:10;

%Working on function h; sinc(x) returns (sin(pi.x))/(pi.x);
therefore;
h=sinc(2*t);
i=1;

%Working on function x, where u(t) is denoted with inbuilt
heaviside(t)
%In MATLAB heaviside(0) = 0.5

for w=-10:0.01:10
 if(w==0)
 x(i)=heaviside(w)-heaviside(w-5)+0.5;
 i=i+1;
 elseif(w==5)
 x(i)=heaviside(w)-heaviside(w-5)-0.5;
 i=i+1;
 else
 x(i)=heaviside(w)-heaviside(w-5);
 i=i+1;
 end
end
```

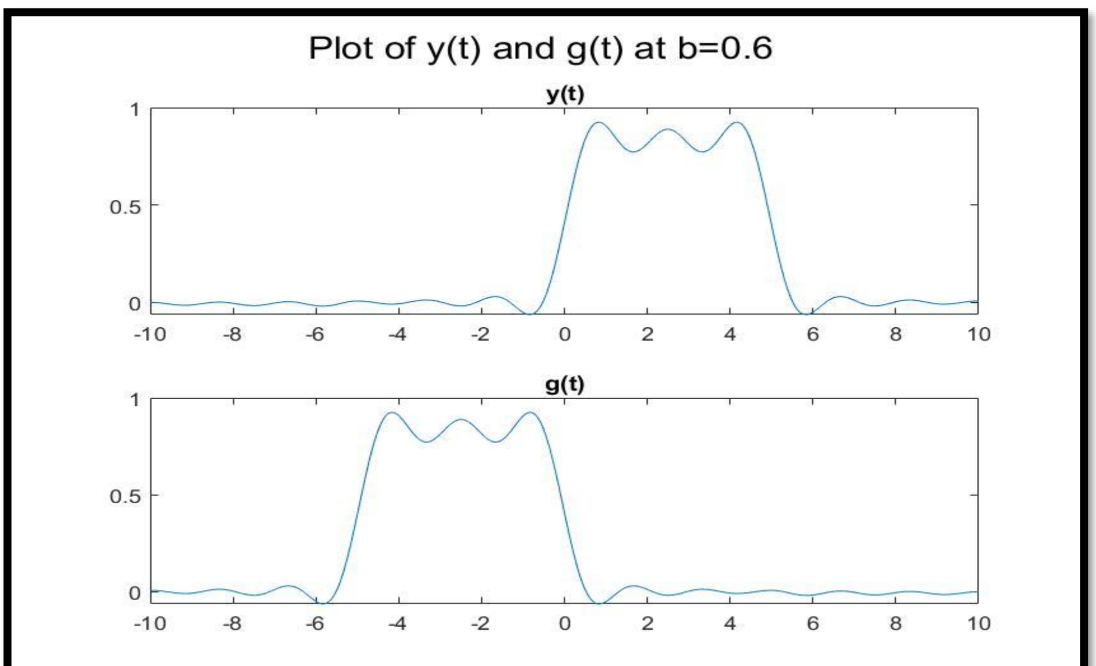
```
%conv() flips the signal, so what we do is that we give a flipped
signal input to the conv() function of MATLAB.

  % For plotting the figure of x(t) and h(t)
  sgtitle("The Input Signal and its Impulse Response")
  subplot(2,1,1)
  plot(t,x)
  title("x(t)")
  subplot(2,1,2)
  plot(t,h)
  title("h(t)")
  figure;


  % Convolution of x(t) with h(t) and its plot code is as follows:
  y=conv(x,h,'same')*0.01;  %Simple Convolution
  sgtitle("y(t) & g(t) at b=1")
  subplot(2,1,1)
  plot(t,y)
  title("y(t)")
  xflip=fliplr(x); %For flipping signal with respect to y-axis
  g=conv(xflip,h,'same')*0.01; %Convolution without flipping
  subplot(2,1,2)
  plot(t,g)
  title("g(t)")
  figure;


  % Case 1: 0<b<1 at b=0.6
  t=-10:0.01:10;
  h=sinc(0.6*2*t); %since b=0.6
  y=conv(x,h,'same')*0.01;  %Simple Convolution
  sgtitle("Plot of y(t) and g(t) at b=0.6 ")
  subplot(2,1,1)
  plot(t,y)
  title("y(t)")
  xflip=fliplr(x); %For flipping signal with respect to y-axis
  g=conv(xflip,h,'same')*0.01; %Convolution without flipping
  subplot(2,1,2)
  plot(t,g)
  title("g(t)")
  figure;
```

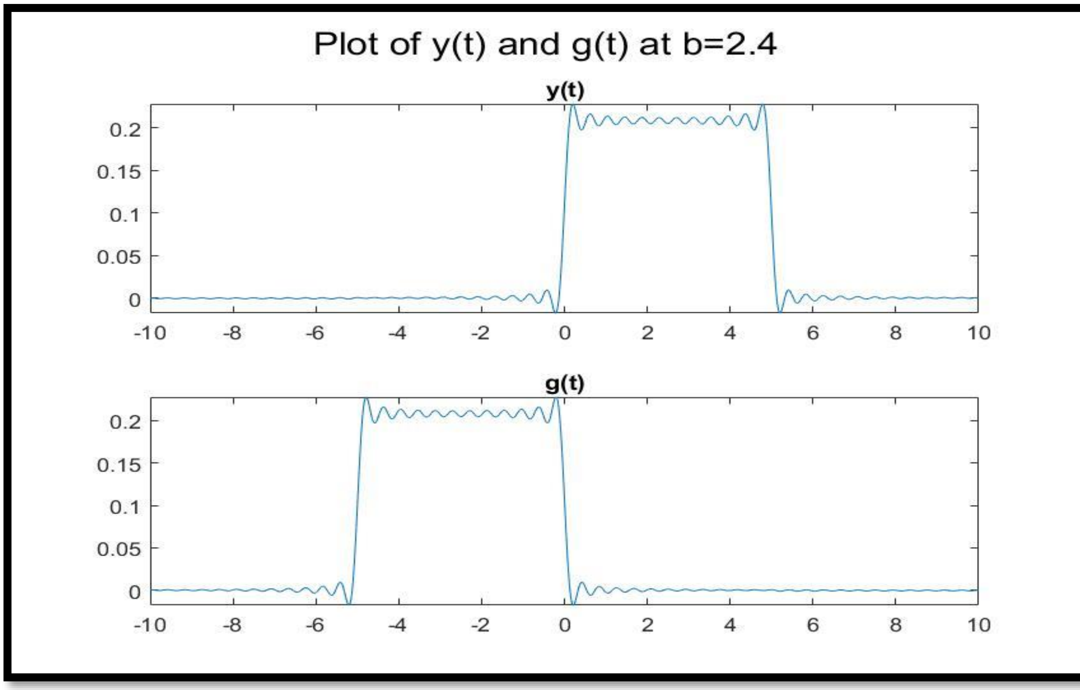
```
% Case 2: 1<b<finite positive value at b=2.4
t=-10:0.01:10;
h=sinc(2.4*2*t); %since b=2.4
y=conv(x,h,'same')*0.01;
sgtitle("Plot of y(t) and g(t) at b=2.4")
subplot(2,1,1)
plot(t,y)
title("y(t)")
xflip=fliplr(x);
g=conv(xflip,h,'same')*0.01;
subplot(2,1,2)
plot(t,g)
title("g(t)")
```

INPUT/OUTPUT PLOTS FOR QUESTION A.]




y(t) & g(t) at b=1
y(t)
g(t)


Plot of y(t) and g(t) at b=0.6
y(t)
g(t)



RESULT FOR QUESTION A]:

After running the code, the plots of convoluted signals were obtained at different values of b in h(bt) (different examples of time scaling of h(t)) for two cases namely :

i.) Flipping in four-step process

ii.) Without flipping in four-step process.

COMPARATIVE CONCLUSION:

We conclude here that even if we miss the flipping step, we get the h(t) signal symmetric to the y-axis.

The only difference is that the starting points and indices are different in both the flip and non-flipping cases of the convoluted signal.

Since, x(t) is a Heaviside or unit step function, it possess same values at all the points; hence, the convoluted signal has same value too.

When h(t) is scaled to h(bt), the results change. When;

i.) $0<b<1$ (my case b=0.6) : The graph has a spike, i.e, there is a sudden rise in value of convolution indicating that the frequency has increased by a factor of 'b'.

QUESTION B.]:

Write a sample script to convolve following discrete time signals;

Plot x(n), h(n) and convolved output y(n).
x(n) = u(n) - u(n-5) with h(n) = n.x(n);

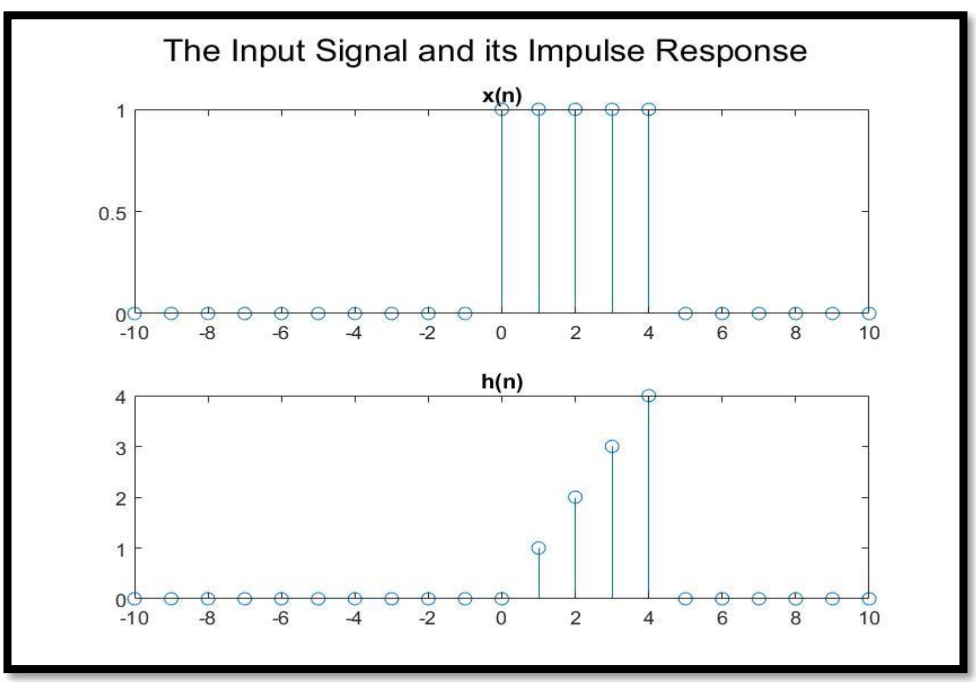
Repeat the processing without flipping (in the four step process) by means of convolution to obtain g(n). Compare g(n) with y(n).

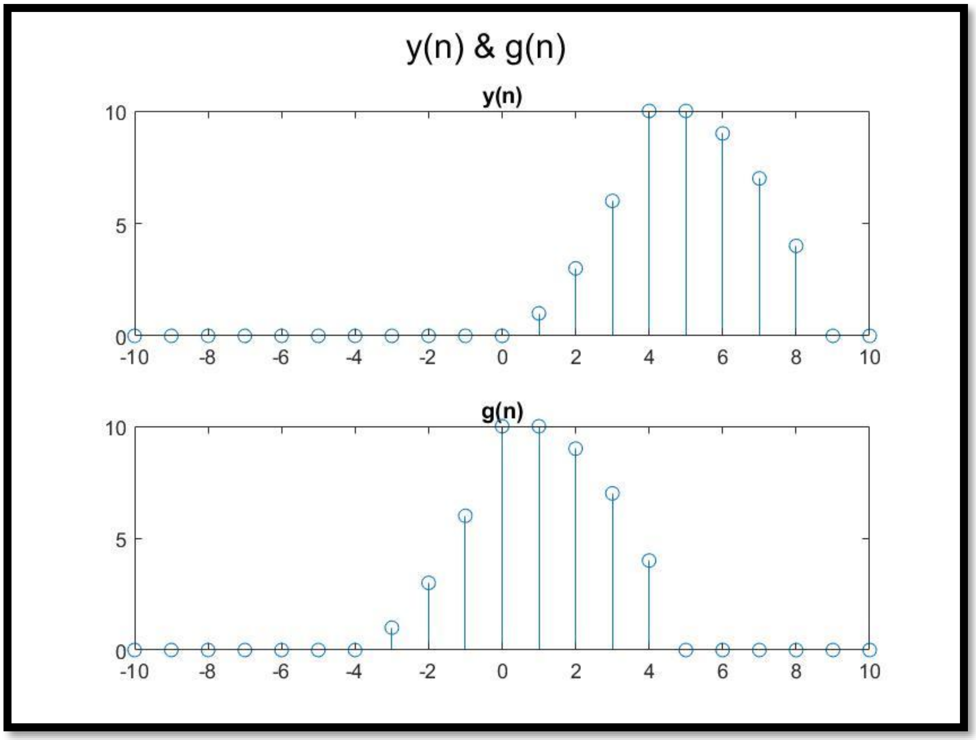
MATLAB CODE:

```
%PART B]
%WE ARE SUPPOSED TO
%Write a sample script to convolve following discrete time
signals;
%Plot x(n), h(n) and convolved output y(n).
%x(n) = u(n) - u(n-5)    with    h(n) = n.x(n);

clc
i=1;
x = 0:1:20;

%Working on function x, where u(t) is denoted with inbuilt
heaviside(t)
%In MATLAB heaviside(0) = 0.5
%For Function x[n]

for w=-10:1:10
 if(w==0)
 x(i)=heaviside(w)-heaviside(w-5)+0.5;
 i=i+1;
 elseif(w==5)
 x(i)=heaviside(w)-heaviside(w-5)-0.5;
 i=i+1;
 else
 x(i)=heaviside(w)-heaviside(w-5);
 i=i+1;
 end
end

n=-10:1:10;

%Function h[n]
h=n.*x;
```

```
sgtitle("The Input Signal and its Impulse Response")
subplot(2,1,1)
stem(n,x)
title("x(n)")
subplot(2,1,2)
stem(n,h)
title("h(n)")
figure;

%Convolution
y=conv(x,h,'same');


sgtitle("y(n) & g(n)")
subplot(2,1,1)
stem(n,y)
title("y(n)")

%Convolution without flip
xflip=fliplr(x); %To flip signal with respect to y-axis
g=conv(xflip,h,'same'); %Since the conv() flips the signal we
input a flipped signal.
subplot(2,1,2)
stem(n,g)
title("g(n)")
```

OUTPUT FOR QUESTION B.]





RESULT FOR QUESTION B] :

Here, the result we obtained after running the code were plots of discrete convoluted signals mainly for the two cases:

i.) Four step convolution with flipping ;

ii.) Four Step Convolution without flipping.

COMPARATIVE CONCLUSION:

Here, in the case of discrete signals, we observed that missing out the flipping operation changes just the indices of the convoluted signal; otherwise similar results and values of the convoluted signal are observed. Here, h[n] is not symmetric about the y axis and the signal x[n] being an unit step function for the fixed range. When the flip operation is missed, we observe that the final signal g[n] shifts towards the left.