"In The Name of God The compassionate and the merciful"

MATLAB Project &Mathematica

Design by : MohammadReza Arani

UserName:810196405

CA designers:

Miss Hodabarkhordarpoor

Mr Ali Ranjbar

Chief TA:Dr.Jamal Kazazi

Sources: My own hardwork , Internet sites such as codesara.ir & good friends

Parts:

1.part1:MATHEMATICA

Part1_1:

Part1_2:

Part1-3:

Part1-4:

Part1-5:

2.part2:MATLAB

Part2_1:

Part2_2:

Part2-3:

Part2-4:

Part2-5:

Part1:MATHEMATICA

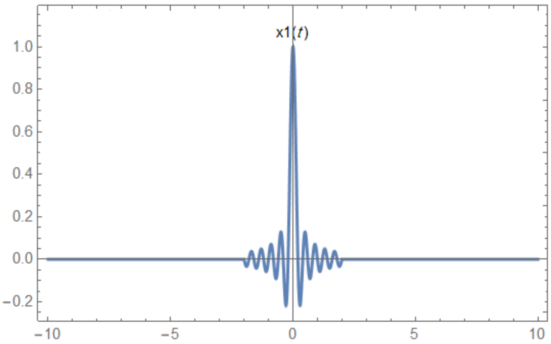
Part1-1:A) In this part we have to implement the functions in mathematica so to plot it properlyin each case.

```
x1[x_] = Sinc[5*(Pi)*t]*UnitStep[4-t^2]
plot1 = Plot[x1[t], {t, -10, 10}, PlotLabels → Placed[Automatic, Above], PlotStyle → Thick, Frame → True]
```

Out[ ]= $\begin{cases} 1 & -\frac{1}{2} \le x \le \frac{1}{2} \\ 0 & \text{True} \end{cases}$

Out[ ]= Sinc[5 π t] UnitStep[4 - t²]

Out[ ]=

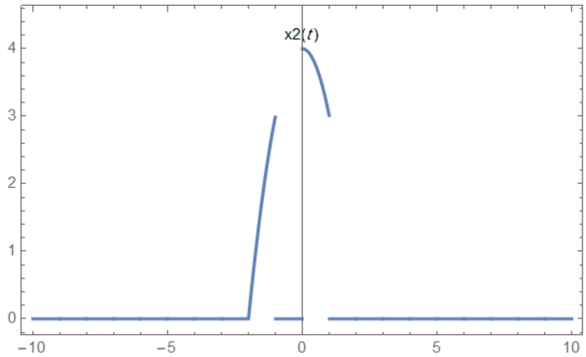


B)

```
In[ ]:= x2[x_] = UnitStep[Sin[(Pi)*t]]*Ramp[4-t^2]
plot2 = Plot[x2[t], {t, -10, 10}, PlotLabels → Placed[Automatic, Above], PlotStyle → Thick, Frame → True]
```

Out[ ]= Ramp[4 - t²] UnitStep[Sin[π t]]

Out[ ]=

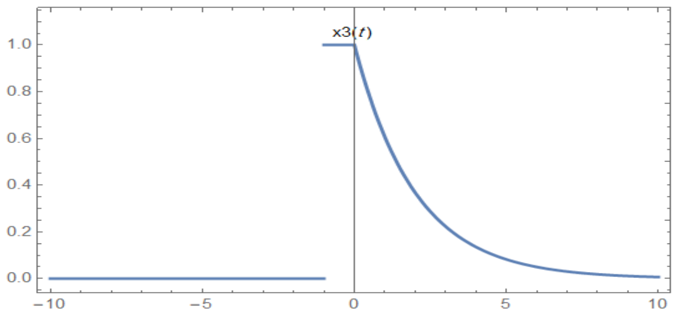


C)

```
In[ ]:= x3[x_] = Piecewise[{{0, t < -1}, {1, -1 < t < 0}, {Exp[-1/2 * t], 0 < t}}]
plot3 = Plot[x3[t], {t, -10, 10}, PlotLabels → Placed[Automatic, Above], PlotStyle → Thick,
  Frame → True, PlotRange → Full]
```

Out[ ]= $\begin{cases} 0 & t < -1 \\ 1 & -1 < t < 0 \\ e^{-t/2} & 0 < t \\ 0 & \text{True} \end{cases}$

Out[ ]=

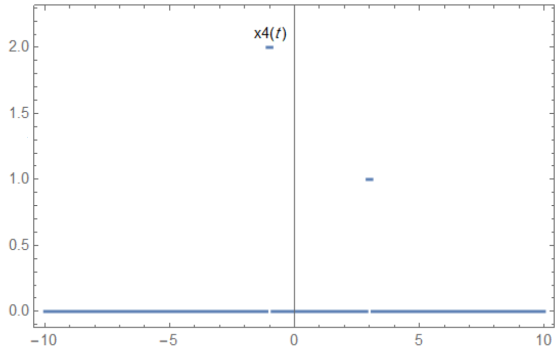


D)

```
In[•]:= x4[x_] = delta2[t - 3] + 2 delta2[t + 1]
        plot2 = Plot[x4[t], {t, -10, 10}, PlotLabels → Placed[Automatic, Above], PlotStyle → Thick,
          Frame → True, PlotRange → Full]
```

Out[•]= $\left(\begin{cases} 1 & -\frac{1}{10} \le -3 + t \le \frac{1}{10} \\ 0 & \text{True} \end{cases}\right) + 2\left(\begin{cases} 1 & -\frac{1}{10} \le 1 + t \le \frac{1}{10} \\ 0 & \text{True} \end{cases}\right)$

Out[•]=



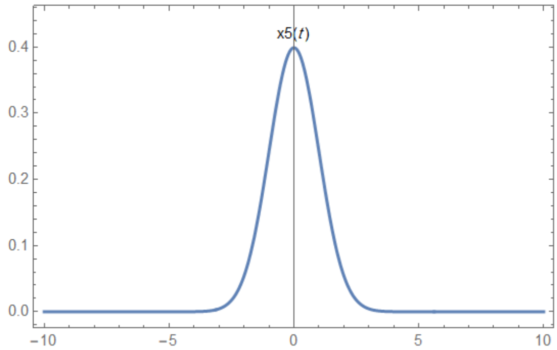
E)

In[•]:= x5[x_] = $\frac{1}{\sqrt{2\pi}}$ * Exp[$\frac{-1}{2}$ * t^2]

```
plot2 = Plot[x5[t], {t, -10, 10}, PlotLabels → Placed[Automatic, Above], PlotStyle → Thick,
  Frame → True, PlotRange → Full]
```

Out[•]= $\frac{e^{-\frac{t^2}{2}}}{\sqrt{2\pi}}$

Out[•]=



Part2:

Part1-2:In this very part we mean to calculate the energy asked in question:

```
In[◦]:= eng[x_] = Integrate[(x)^2, {t, -Infinity, +Infinity}]
        eng[x1[t]]
        eng[x2[t]]
        eng[x3[t]]
        eng[x4[t]]
        eng[x5[t]]
        Table[eng[x1[t]], eng[x2[t]], eng[x3[t]], eng[x4[t]], eng[x5[t]], {1, 5}]
```

Part1-3:In this section we need to accomplish the power of each signal:

```
pow[x_] = Limit[1/T Integral[(x)^2], T → +Infinity]
pow[x1[t]]
pow[x2[t]]
pow[x3[t]]
pow[x4[t]]
pow[x5[t]]
```

Part1-4:DC part and tables:

```
In[◦]:= Table[eng[i], {i, {x1[t], x2[t], x3[t], x4[t], x5[t]}}]
```

Out[◦]= $\left\{\frac{2\,\text{SinIntegral}[20\,\pi]}{5\,\pi}, \frac{256}{15}, 2, 1, \frac{1}{2\sqrt{\pi}}\right\}$

```
In[◦]:= Table[pow[i], {i, {x1[t], x2[t], x3[t], x4[t], x5[t]}}]

In[◦]:= {0, 0, 0, 0, 0}
        dcfunc[x_] = Integrate[(x), {t, -Infinity, +Infinity}]
        Table[dcfunc[i], {i, {x1[t], x2[t], x3[t], x4[t], x5[t]}}]

Out[◦]= {0, 0, 0, 0, 0}
```

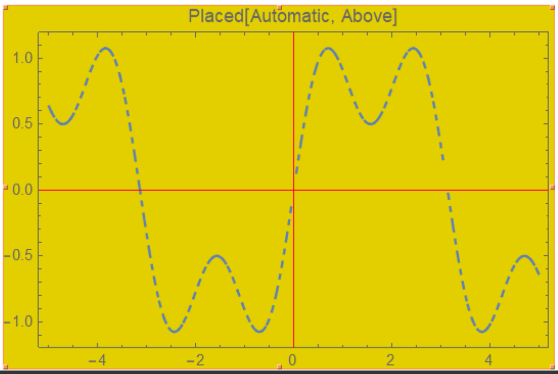
Part:1-5:Sampling method is done by defining a delta2 function:

```
x6[x_] = Sin[t] + 1/2 Sin[3 t];

Out[9]= {0, 0, 0, 0, 0}

In[27]:= xs[x_] = Sum[x6[0.1 n] * delta2[t - 0.1 n], {n, -50, +50}];
         plot2 = Plot[xs[t], {t, -5, +5}, Frame → True, PlotRange → Full, PlotLabel → Placed[Automatic, Above],
           PlotStyle → Dashed]
```

Out[30]=

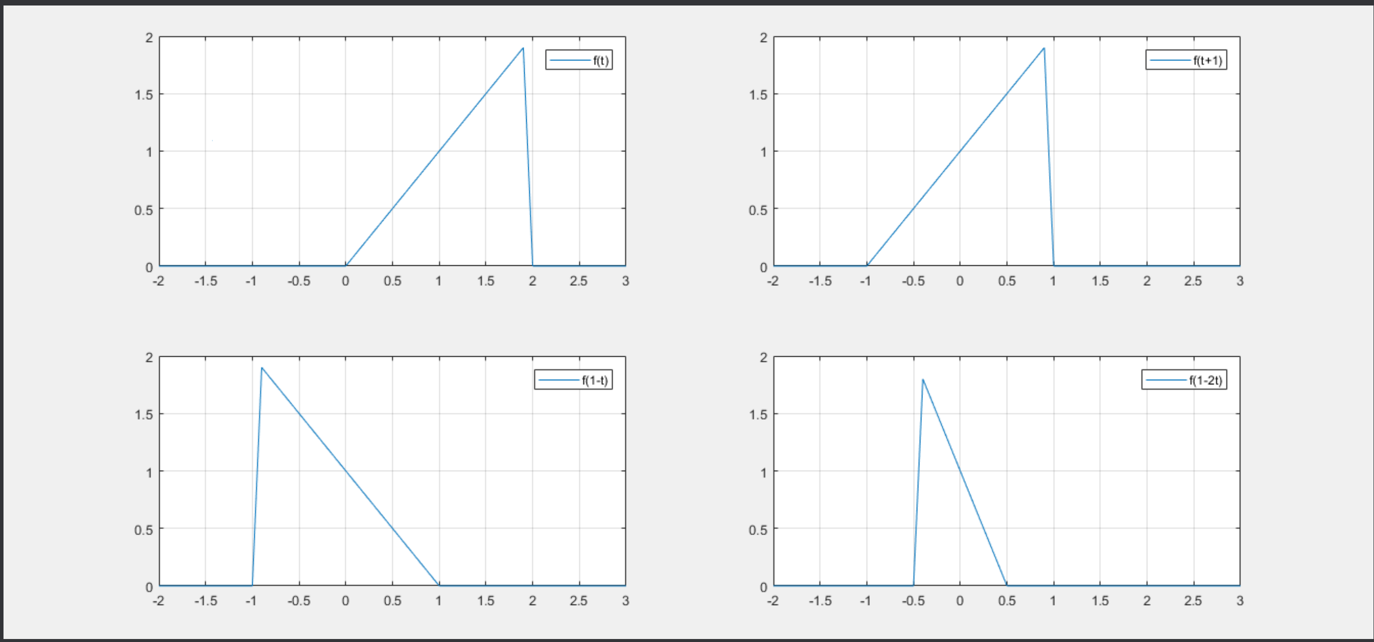



```
delta2[x_] = Piecewise[{{20, -5/100 <= x ≤ 5/100}, {0, Abs[x] > 5/100}}]
plot1 = Plot[delta2[x], {x, -1, 1},
  PlotLabel → Placed[Automatic, Above], Frame → True]
```

Part2:MATLAB

Part2-1: $f(t) = t(u(t) - u(t - 2))$

We have f(t) and looking for f(t+1),f(1-2t),f(1-t) which comes in the following:




```
clc;
clear;
u=@(t)double(t>=0);
t=-2:0.1:3;
f1=t.*(u(t)-u(t-2));
f2=(t+1).*(u(t+1)-u(t-1));
f3=(1-t).*(u(1-t)-u(-1-t));
f4=(1-2*t).*(u(1-2*t)-u(-1-2*t));
subplot(2,2,4);
plot(t,f4,'DisplayName','f(1-2t)');
grid on;
legend('show');
subplot(2,2,3);
plot(t,f3,'DisplayName','f(1-t)');
grid on;
legend('show');
subplot(2,2,2);
plot(t,f2,'DisplayName','f(t+1)');
grid on;
legend('show');
subplot(2,2,1);
plot(t,f1,'DisplayName','f(t)');
grid on;
legend('show');
```

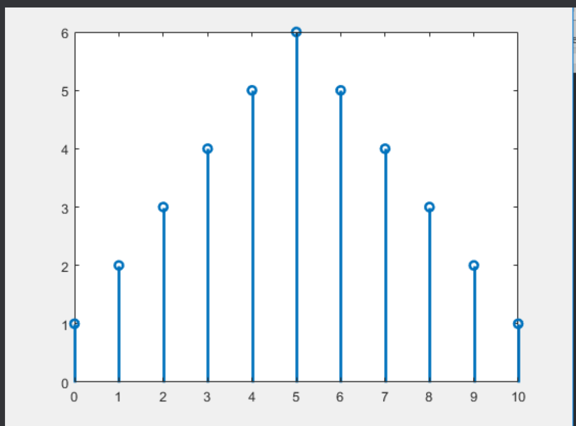
Part2-2: in this part we shall learn how to use conv and stem and then we learned so☺.

Editor - C:\Users\Mohammad Reza\Desktop\matproject\part2.m

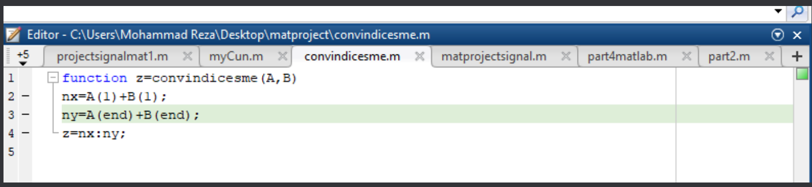
projectsignalmat1.m | myCun.m | convindicesme.m | matprojectsignal.m | part4matlab.m | part2.m

```
clear ; clc
np=0:5;
k=ones(size(np));
ni=convindicesme(np,np);

y=conv(k,k);
figure
stem(ni,y,'linewidth',2)
```



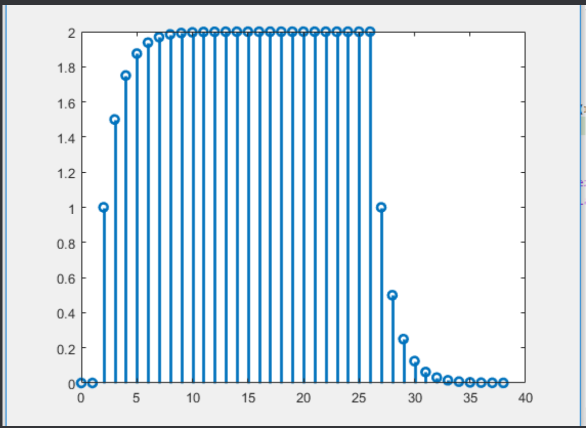
Part2-3: Now we need to make a function to give us the interval points to take care:for this fact we got the first and the last points as we learned in signals&systems class that if 2 seperated signals with t1,t2 and t3,t4 interval points convolves their convindice would be t1+t3,t2+t4;



Editor - C:\Users\Mohammad Reza\Desktop\matproject\convindicesme.m

```
function z=convindicesme(A,B)
nx=A(1)+B(1);
ny=A(end)+B(end);
z=nx:ny;
```

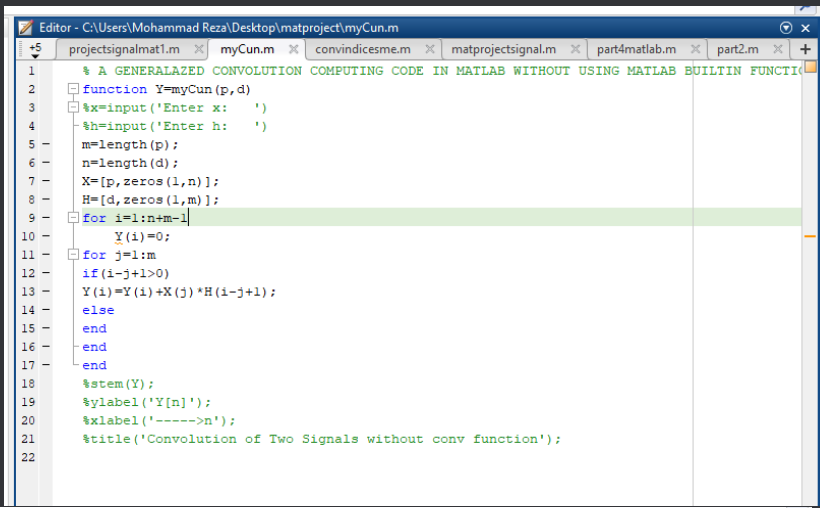
Part2-4: As said in the asking question we've got functions to convolve so then we do it as the steps goes on:



Editor - C:\Users\Mohammad Reza\Desktop\matproject\matprojectsignal.m

```
clear;
clc;
u = @( t ) double( t >= 0) ;
nx=0:1:14;
nh=0:1:24;
ny=convindicesme(nx,nh);
x=@(nx) double((1/2).^(nx-2).*u(nx-2));
h=@(nh) u(nh);
y=conv(x(nx),h(nh));
%title('plot of signalyn=xn*xn','interpreter','latex','frontsize',16);
xlabel('$n$','interpreter','latex');
ylabel('$y[n]$','interpreter','latex');
stem(ny,y,'linewidth',2);
```

Part2-5: we implemented our myCun function to have it convolve, then we tried hard to match inputs and outputs:



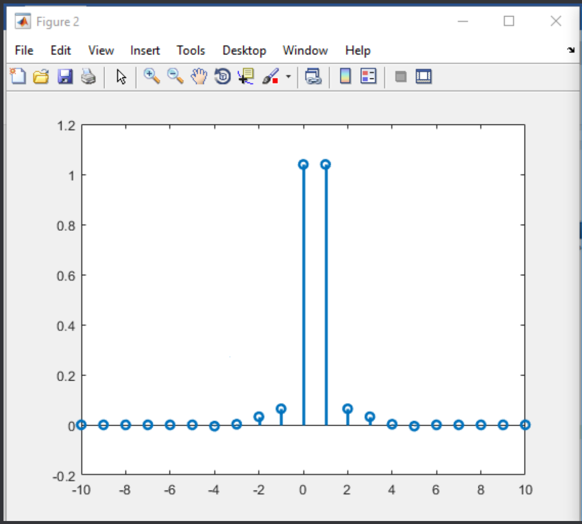
Editor - C:\Users\Mohammad Reza\Desktop\matproject\myCun.m

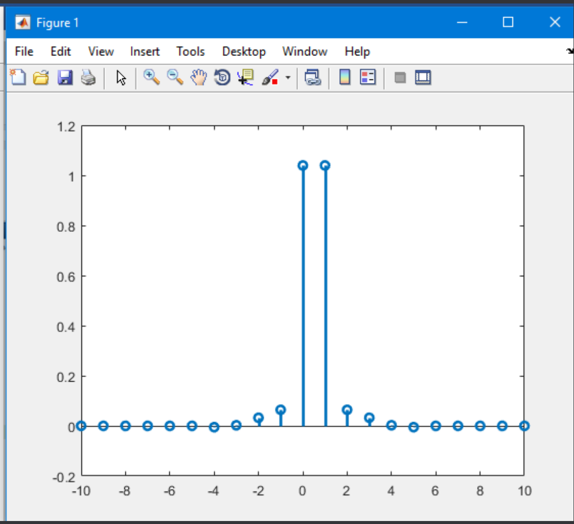
```
% A GENERALAZED CONVOLUTION COMPUTING CODE IN MATLAB WITHOUT USING MATLAB BUILTIN FUNCTI
function Y=myCun(p,d)
%x=input('Enter x:   ')
%h=input('Enter h:   ')
m=length(p);
n=length(d);
X=[p,zeros(1,n)];
H=[d,zeros(1,m)];
for i=1:n+m-1
    Y(i)=0;
for j=1:m
if(i-j+1>0)
Y(i)=Y(i)+X(j)*H(i-j+1);
else
end
end
end
%stem(Y);
%ylabel('Y[n]');
%xlabel('----->n');
%title('Convolution of Two Signals without conv function');
```

Editor - C:\Users\Mohammad Reza\Desktop\matproject\part4matlab.m

```
u=@(t) double(t >= 0);
n = -5:1:5;
h= sinc(2*(pi)*n).*(u(n+4)-u(n-5));
x=u(n)-u(n-2);

tic
y1=conv(h,x);
t1=toc
figure
stem(convindicesme(n,n),y1,'linewidth',2)
tic
y2=myCun(h,x);
t2=toc
figure
stem(convindicesme(n,n),y2,'linewidth',2)
```





As it’s up to me there’s no difference between mine and matlab conv in the result but the duration or the elapsed time consumed to execute:

T1=matlabtime=0.0037 s

T2=mytime=0.0039 s

To compare my time is larger because we used too much for in our code so it takes much more time to work.