

SS EXPERIMENT LAB 6

TITLE: Verification of different properties of LTI system:
Commutation,

Association, Distribution, Identity and Fourier analysis and
synthesis

NAME: Yash Gupta

ROLL NO: S20200010234

OBSERVATION: In this lab, I learned how to verify different
properties of LTI system and Fourier analysis and synthesis.

Verify the above properties for the following signals **by writing a MATLAB script**. Next, also verify the result **obtained in MATLAB script** with **Analytical method**.

Convolution function:

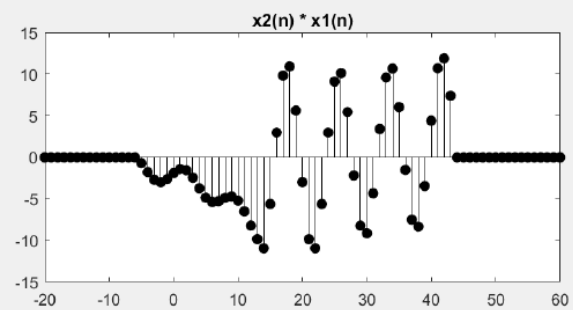
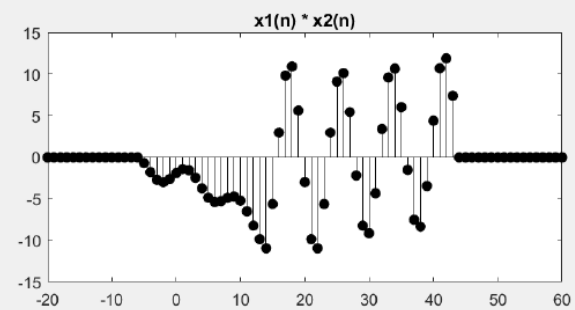
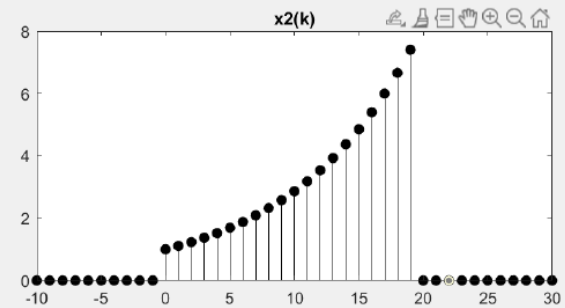
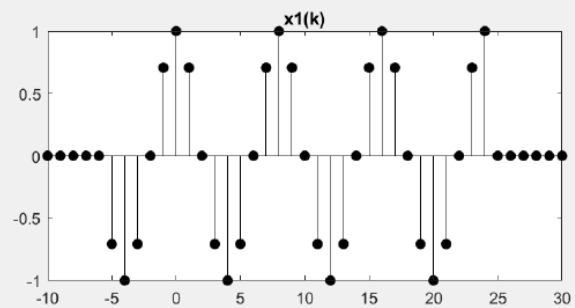
```
1  function [ny,y] = convulation(nx,x,nh,h)
2  nyb=nx(1)+nh(1);
3  nye=nx(length(x))+nh(length(h));
4  ny=nyb:nye;
5  y=zeros(1,length(ny));
6  for i=1:length(ny)
7      [hf, nhf] =sigfold(h,nh);
8      [shf, nshf] = sigshift(hf,nhf, ny(i));
9      [xnew, nm] = sigmult(x,nx,shf,nshf);
10     y(i) = sum(xnew);
11 end
```

Commutative law

```

1 - clear;
2 - close all;
3 - nx1 = -10:30 ;
4 - x1 = cos((pi * nx1)/4).*(stepseq(-5,-10,30) - stepseq(25,-10,30));
5 - nx2 = -10:30 ;
6 - x2 = ((0.9).^-nx2).*(stepseq(0,-10,30) - stepseq(20,-10,30));
7 - %%commutative law:
8 - [ny,y] = convulation(nx2,x2,nx1,x1);
9
10 - subplot(2,2,1);
11 - stem(nx1, x1, 'k', 'filled');
12 - title('x1(k)');
13 - subplot(2,2,2);
14 - stem(nx2, x2, 'k', 'filled');
15 - title('x2(k)');
16 - subplot(2,2,3);
17 - stem(ny, y, 'k', 'filled');
18 - title('x1(n) * x2(n)');
19
20 - [ny,y] = convulation(nx2,x2,nx1,x1);
21
22 - subplot(2,2,4);
23 - stem(ny, y, 'k', 'filled');
24 - title('x2(n) * x1(n)');

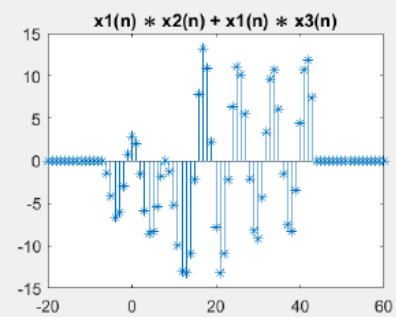
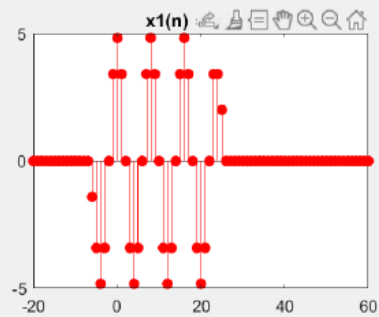
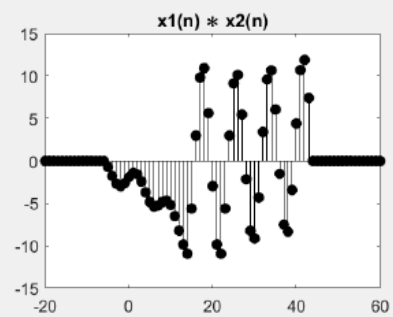
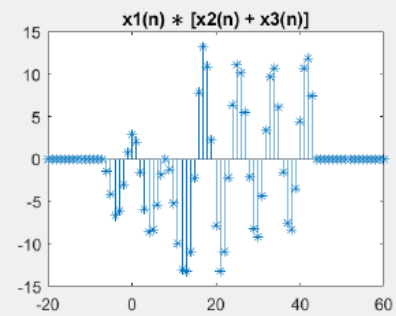
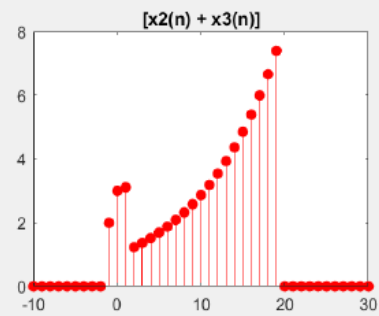
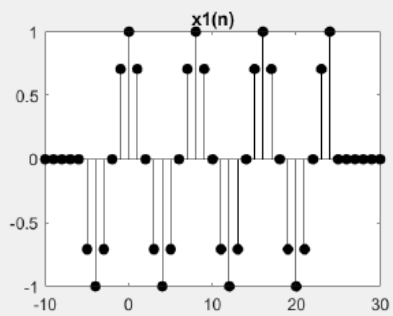
```



Association property

```
1 - clear;
2 - close all;
3 - nx1 = -10:30 ;
4 - x1 = cos((pi * nx1)/4).*(stepseq(-5,-10,30) - stepseq(25,-10,30));
5 - nx2 = -10:30 ;
6 - x2 = ((0.9).^-nx2).*(stepseq(0,-10,30) - stepseq(20,-10,30));
7 - nx3 = -10:10 ;
8 - wn = [0,0,0,0,0,0,0,0,0,1/3,1/3,1/3,0,0,0,0,0,0,0,0];
9 - x3 = round(5 .* wn);
10 - %association property%
11 - [ny,y] = convulation(nx1,x1,nx2,x2);
12 - subplot(2,3,1);
13 - stem(ny, y, 'k', 'filled');
14 - title('x1(n) * x2(n)');
15 - [nz,z] = convulation(ny,y,nx3,x3);
16 - subplot(2,3,2);
17 - stem(nx3, x3, 'k', 'filled');
18 - title('x3(n)');
19 - subplot(2,3,3);
20 - stem(nz, z, 'k', 'filled');

21 - title(' [x1(n) * x2(n)] * x3(n) ');
22 - [ny,y] = convulation(nx2,x2,nx3,x3);
23 - subplot(2,3,4);
24 - stem(ny, y, 'k', 'filled');
25 - title('x2(n) * x3(n)');
26 - [nz,z] = convulation(nx1,x1,ny,y);
27 - subplot(2,3,5);
28 - stem(nx1, x1, 'k', 'filled');
29 - title('x1(n)');
30 - subplot(2,3,6);
31 - stem(nz, z, 'k', 'filled');
32 - title('x1(n) * [x2(n) * x3(n)]');
33
```

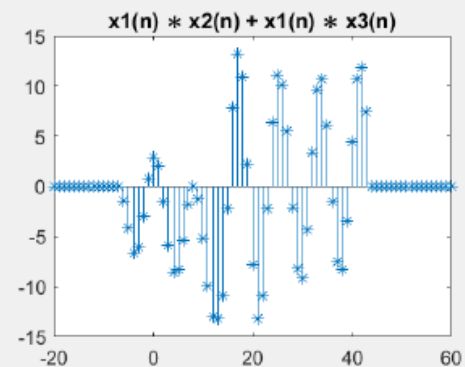
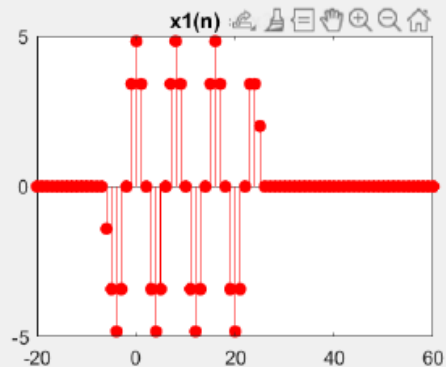
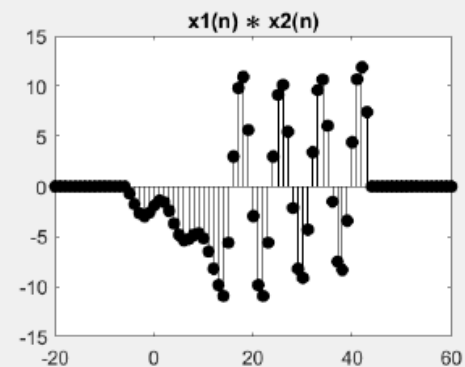
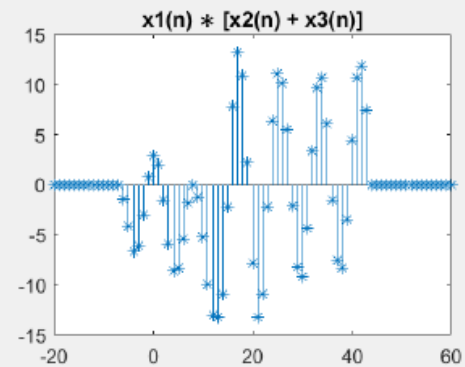
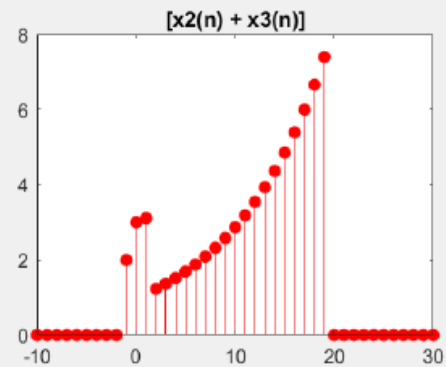
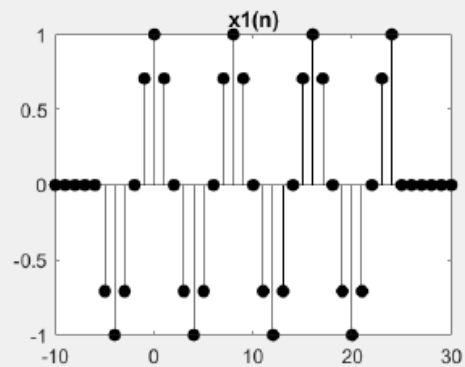


Distribution Law

```

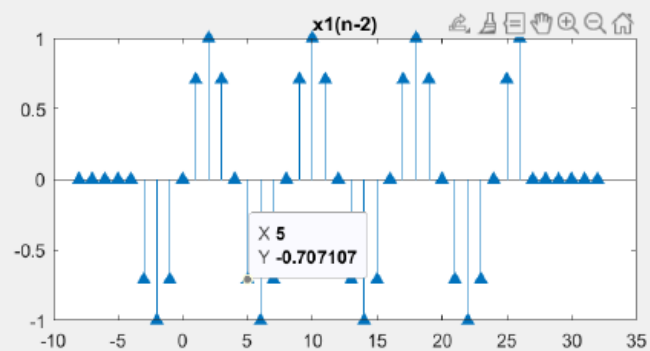
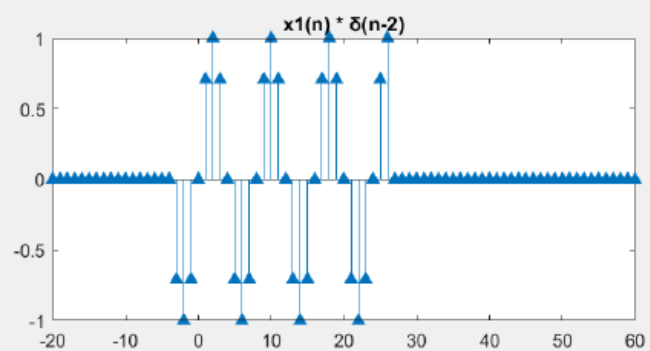
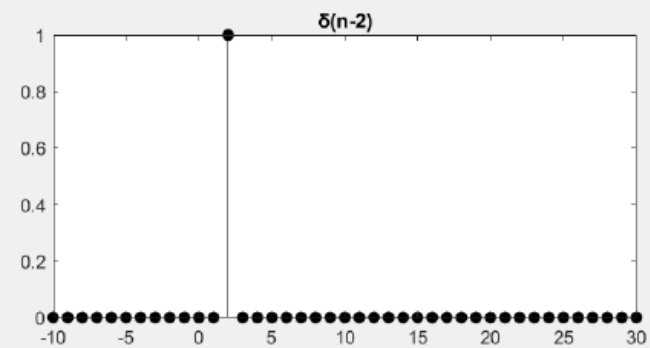
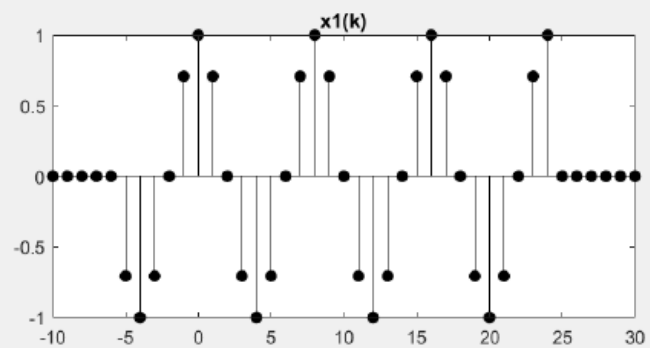
1 clear;
2 close all;
3 nx1 = -10:30 ;
4 x1 = cos((pi * nx1)/4).*(stepseq(-5,-10,30) - stepseq(25,-10,30));
5 nx2 = -10:30 ;
6 x2 = ((0.9).^-nx2).*(stepseq(0,-10,30) - stepseq(20,-10,30));
7 nx3 = -10:30 ;
8 wn = [0,0,0,0,0,0,0,0,0,1/3,1/3,1/3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0];
9 x3 = round(5 .* wn);
10 nx4 = -10:30;
11 x4 = x2 + x3 ;
12 subplot(2,3,1);
13 stem(nx1, x1, 'k', 'filled');
14 title('x1(n)');
15 subplot(2,3,2);
16 stem(nx4, x4, 'r', 'filled');
17 title(['x2(n) + x3(n)'])
18 [ny,y] = convulation(nx1,x1,nx4,x4);
19 subplot(2,3,3);
20 stem(nv, v, '*', 'filled');
21 title('x1(n) * [x2(n) + x3(n)]');
22 [ny,y] = convulation(nx1,x1,nx2,x2);
23 [nz,z] = convulation(nx1,x1,nx3,x3);
24 nw = -20:60;
25 w = y + z;
26 subplot(2,3,4);
27 stem(ny, y, 'k', 'filled');
28 title('x1(n) * x2(n)');
29 subplot(2,3,5);
30 stem(nz, z, 'r', 'filled');
31 title('x1(n) * x3(n)');
32 subplot(2,3,6);
33 stem(nw, w, '*', 'filled');
34 title('x1(n) * x2(n) + x1(n) * x3(n)');

```



Identity Property

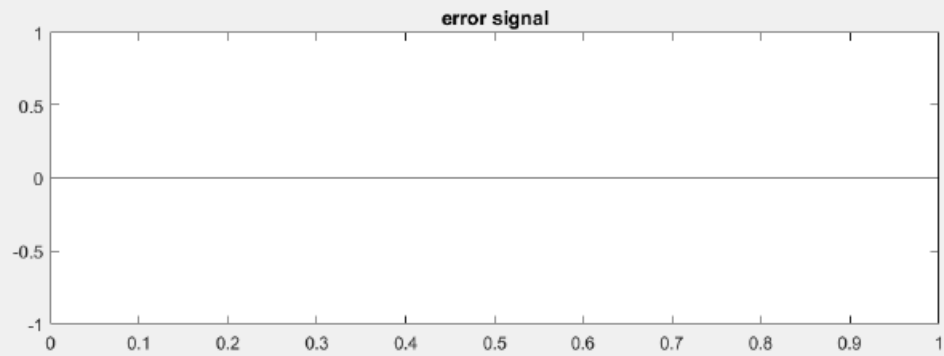
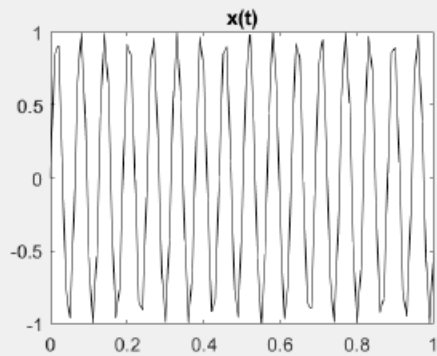
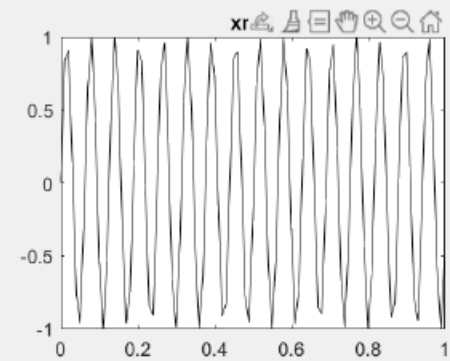
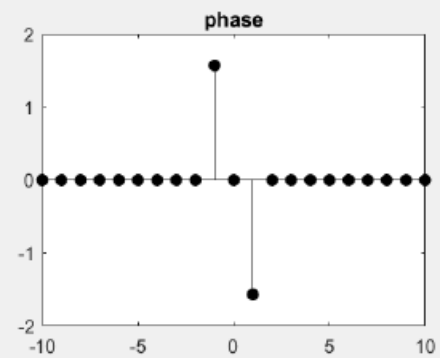
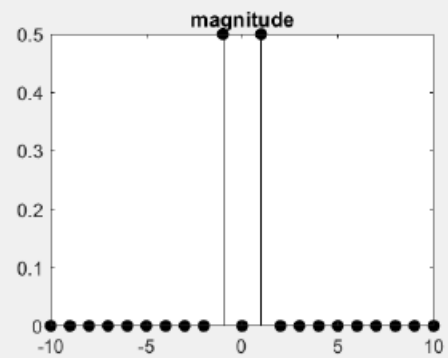
```
1 - clear;
2 - close all;
3 - nx1 = -10:30 ;
4 - x1 = cos((pi * nx1)/4).*(stepseq(-5,-10,30) - stepseq(25,-10,30));
5 - nh = -10:30;
6 - h = impseq(2,-10,30);
7 - % identity property%
8 - [ny,y] = convulation(nx1,x1,nh,h);
9 - subplot(2,2,1);
10 - stem(nx1, x1, 'k', 'filled');
11 - title('x1(k)');
12 - subplot(2,2,2);
13 - stem(nh, h, 'k', 'filled');
14 - title('δ(n-2)');
15 - subplot(2,2,3)
16 - stem(ny, y, '^', 'filled');
17 - title('x1(n) * δ(n-2)');
18 - [z,n] = sigshift(x1,nx1,2);
19 - subplot(2,2,4);
20 - stem(n, z, '^', 'filled');
21 - title('x1(n-2)');
22
```



Task 2: Q1 a

```
2 - close all;
3 - clear;syms x;syms t;
4 - x = sin(100*t);
5 - w0 = 100;
6 - N = 10;
7 - ak = zeros(1,2*N+1);
8 - T = 2*pi/w0; %calculating the period and store in T
9 - syms t;
10 %loop for calculating fourier coefficient
11 - for k = -N:N
12 -     ak(1,1+k+N) = 1/T * int(x * exp(-1j*k*w0*t), 0, T) ;% ak is fourier coefficient
13 -     disp(ak(1+k+N));
14 - end
15 - k = -N:N;
16 - mag = sqrt(real(ak).*real(ak) + imag(ak).* imag(ak));%magnitude of fourier coefficient
17 - subplot(2,3,1);
18 - stem(k,mag,'k','filled');
19 - title('magnitude');
20
21 - phase = angle(ak);
22 - subplot(2,3,2);
23 - stem(k,phase,'k','filled');
24 - title('phase');

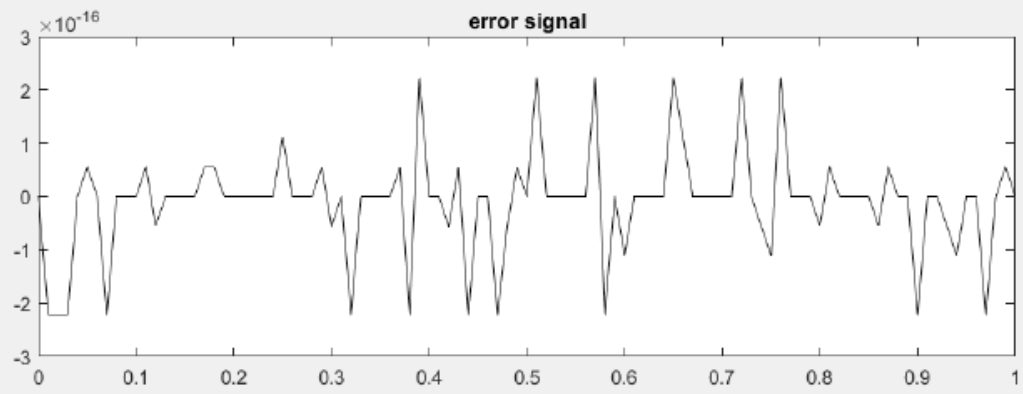
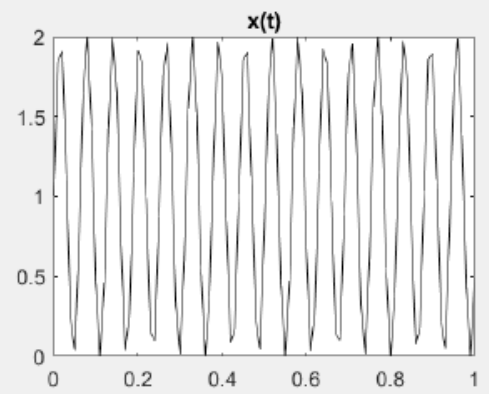
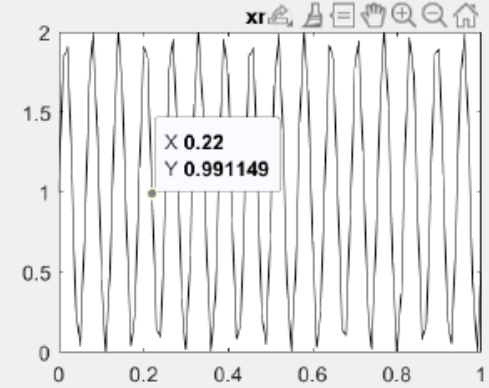
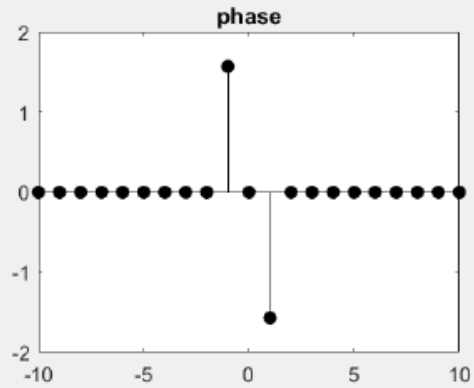
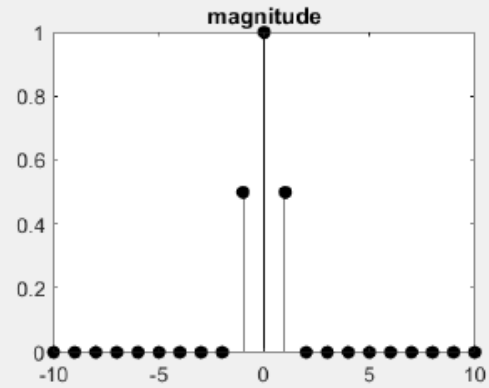
1 %synthesis of fourier series
2 - t = 0:0.01:1;
3 - yt = zeros(size(t));
4 - for k = -N:N
5 -     yt = yt + (ak(1+k+N).*exp(1j*k*w0*t));
6 - end
7 - subplot(2,3,3);
8 - plot(t,yt,'k');
9 - title('xn(t)');
10
11 - subplot(2,3,4);
12 - x = sin(100*t);
13 - plot(t,x,'k');
14 - title('x(t)');
15
16 - subplot(2,3,[5,6]);
17 - plot(t,x-yt,'k');%error signal(e(t))
18 - title('error signal');
19
```



Q1b

```
4- syms x;
5- syms t;
6- x = 1 + sin(100*t);
7- w0 = 100;
8- N = 10;
9- ak = zeros(1,2*N+1);
10- T = 2*pi/w0; %calculate the period and store in T
11- syms t;
12- %loop for calculating fourier coefficient
13- for k = -N:N
14-     ak(1,1+k+N) = 1/T * int(x * exp(-1j*k*w0*t), 0, T) ;% ak is fourier coefficient
15-     disp(ak(1+k+N));
16- end
17- k = -N:N;
18- mag = sqrt(real(ak).*real(ak) + imag(ak).*imag(ak));%magnitude of fourier coefficient
19- subplot(2,3,1);
20- stem(k,mag,'k','filled');
21- title('magnitude');
22-
23- phase = angle(ak);
24- subplot(2,3,2);
25- stem(k,phase,'k','filled');
26- title('phase');
```

```
1 %synthesis of fourier series
2 t = 0:0.01:1;
3 yt = zeros(size(t));
4 for k = -N:N
5     yt = yt + (ak(1+k+N).*exp(1j*k*w0*t));
6 end
7
8 subplot(2,3,3);
9 plot(t,yt,'k');
10 title('xn(t)');
11 subplot(2,3,4);
12
13 x = 1 + sin(100*t);
14 plot(t,x,'k');
15 title('x(t)');
16
17 subplot(2,3,[5,6]);
18 plot(t,x-yt,'k');%error signal(e(t))
19 title('error signal');
20
```



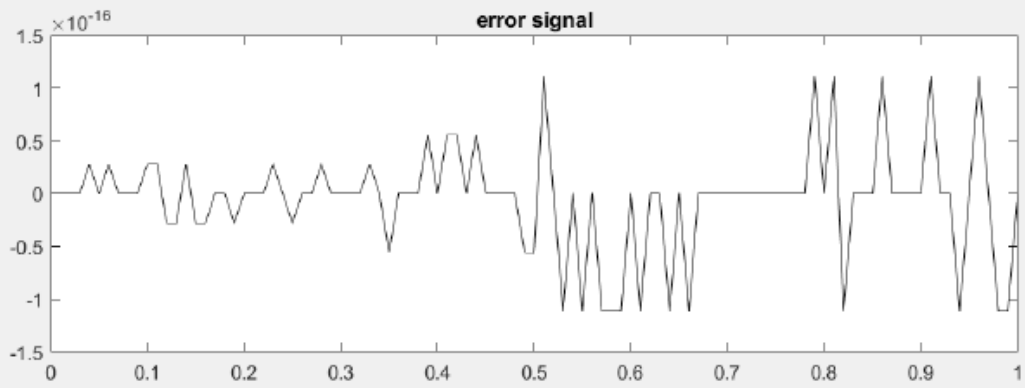
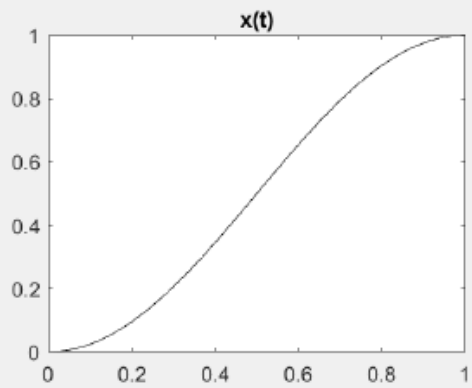
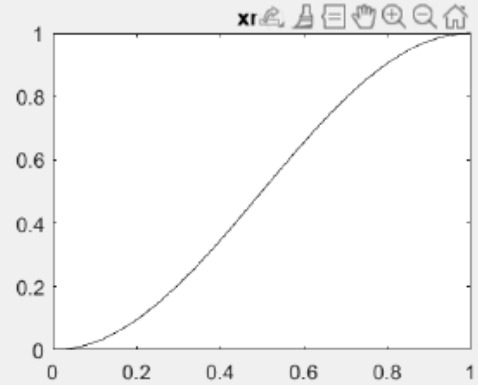
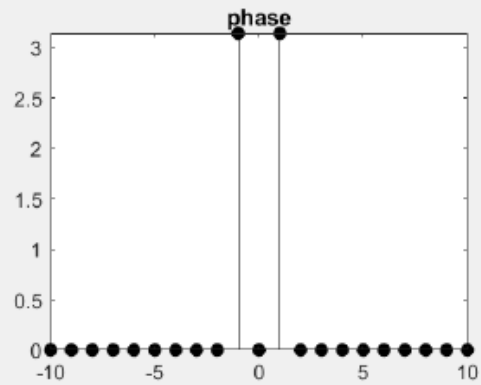
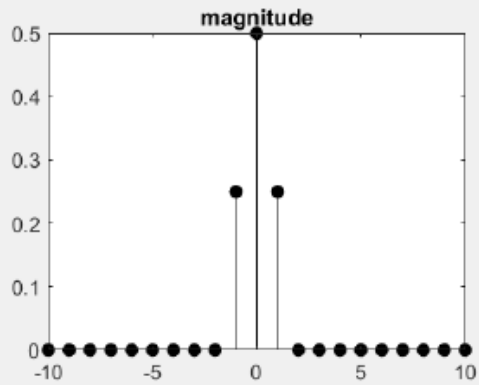
Q1c

```
4- syms x;
5- syms t;
6- w0 = pi;
7- x = (1/2)*(1-cos(w0 * t));
8- N = 10;
9- ak = zeros(1,2*N+1);
10- T = 2*pi/w0; %calculate the period and store in T
11- syms t;
12- %loop for calculating fourier coefficient
13- for k = -N:N;
14-     ak(1,1+k*N) = 1/T * int(x * exp(-1j*k*w0*t), 0, T) ;%ak is fourier coefficient
15-     disp(ak(1+k*N));
16- end
17- k = -N:N;
18- mag = sqrt(real(ak).*real(ak) + imag(ak).* imag(ak));%magnitude of fourier coefficient
19- subplot(2,3,1);
20- stem(k,mag,'k','filled');
21- title('magnitude');
22-
23- phase = angle(ak);
24- subplot(2,3,2);
25- stem(k,phase,'k','filled');
26- title('phase');
```

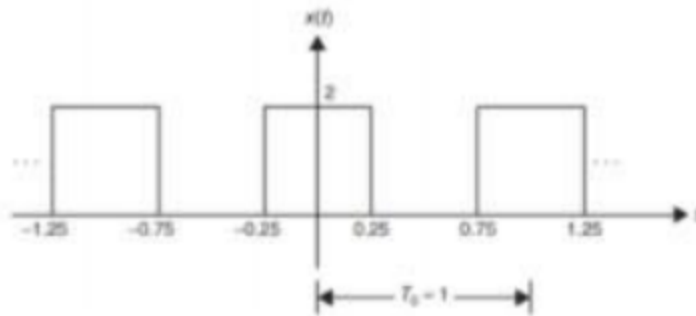
```

1- t = 0:0.01:1;
2- yt = zeros(size(t));
3- for k = -N:N
4-     yt = yt + (ak(1+k*N).*exp(1j*k*w0*t));
5- end
6- subplot(2,3,3);
7- plot(t,yt,'k');
8- title('xn(t)');
9- subplot(2,3,4);
10- x = (1/2)*(1-cos(w0 * t));
11- plot(t,x,'k');
12- title('x(t)');
13- subplot(2,3,[5,6]);
14- plot(t,x-yt,'k');%error signal(e(t))
15- title('error signal');
16

```

(d)

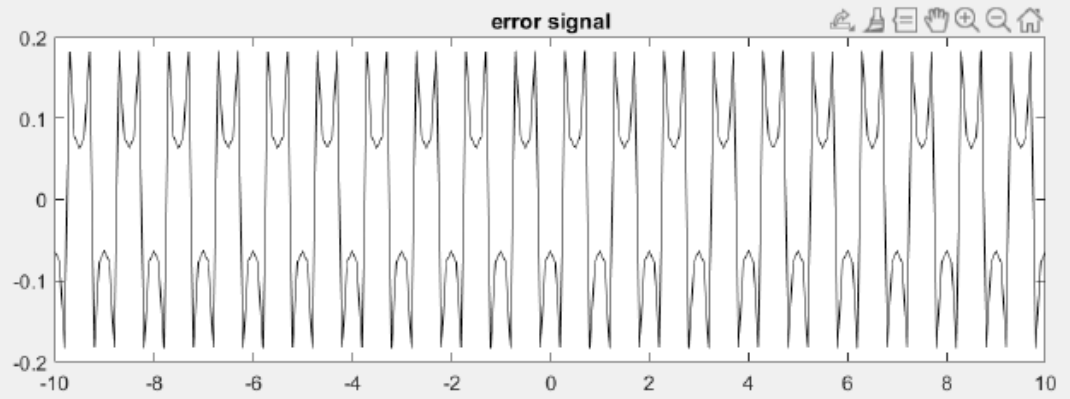
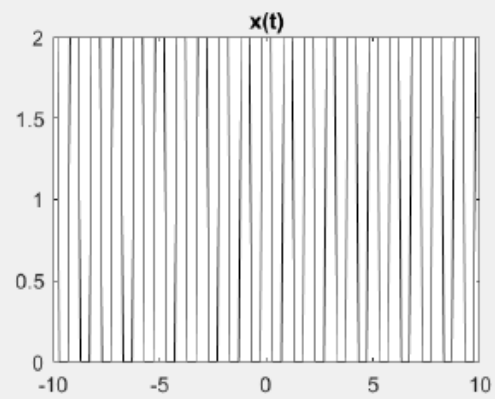
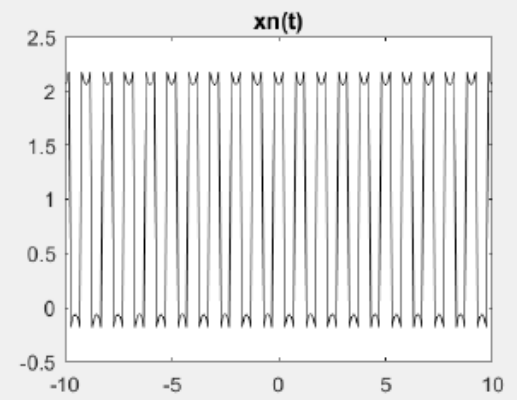
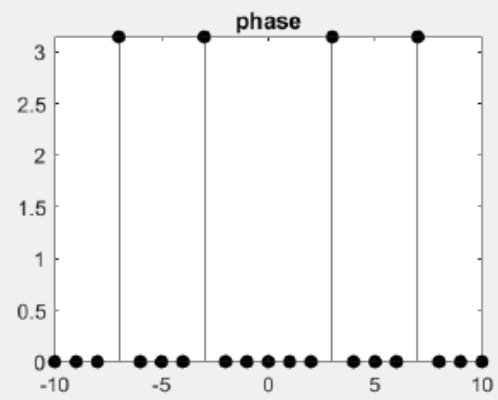
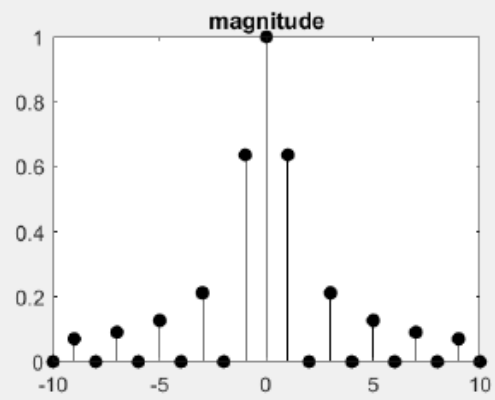


```
5 - x = zeros(size(t));
6 - for p = -10:10
7 -     x = x + 2*(heaviside(t+p+0.25) - heaviside(t+p-0.25));
8 - end
9 - w0 = 2*pi;
10 - N = 10;
11 - ak = zeros(1,2*N+1);
12 - T = 2*pi/w0;
13 - syms t;
14 - for k = -N:N;
15 -     ak(1,1+k+N) = 1/T * int(x * exp(-1j*k*w0*t), 0, T) ;%ak is fourier coefficient
16 -     disp(ak(1+k+N));
17 - end
18 - k = -N:N;
19 - mag = sqrt(real(ak).*real(ak) + imag(ak).*imag(ak));
20 - subplot(2,3,1);
21 - stem(k,mag,'k','filled');
22 - title('magnitude');
23
24 - phase = angle(ak);
25 - subplot(2,3,2);
26 - stem(k,phase,'k','filled');
27 - title('phase');
```

```

1      %synthesis of fourier series;
2      t = -10:0.1:10;
3      yt = zeros(size(t));
4      for k = -N:N
5          yt = yt + (ak(1+k+N).*exp(1j*k*w0*t));
6      end
7      subplot(2,3,3);
8      plot(t,yt,'k');
9      title('xn(t)');
10     subplot(2,3,4);
11     t = -10:0.1:10;
12     x = zeros(size(t));
13     for p = -10:10
14         x = x + 2*(heaviside(t+p+0.25) - heaviside(t+p-0.25));
15     end
16     plot(t,x,'k');
17     title('x(t)');
18     subplot(2,3,[5,6]);
19     plot(t,x-yt,'k');%error signal(e(t))
20     title('error signal');
21

```



THANK YOU