School of Engineering and Applied Science (SEAS)
Ahmedabad University

BTech(ICT) Semester VI:Digital Signal Processing

Laboratory Assignment-1

Enrollment No:AU1841131 **Name:Mansi Dobariya**

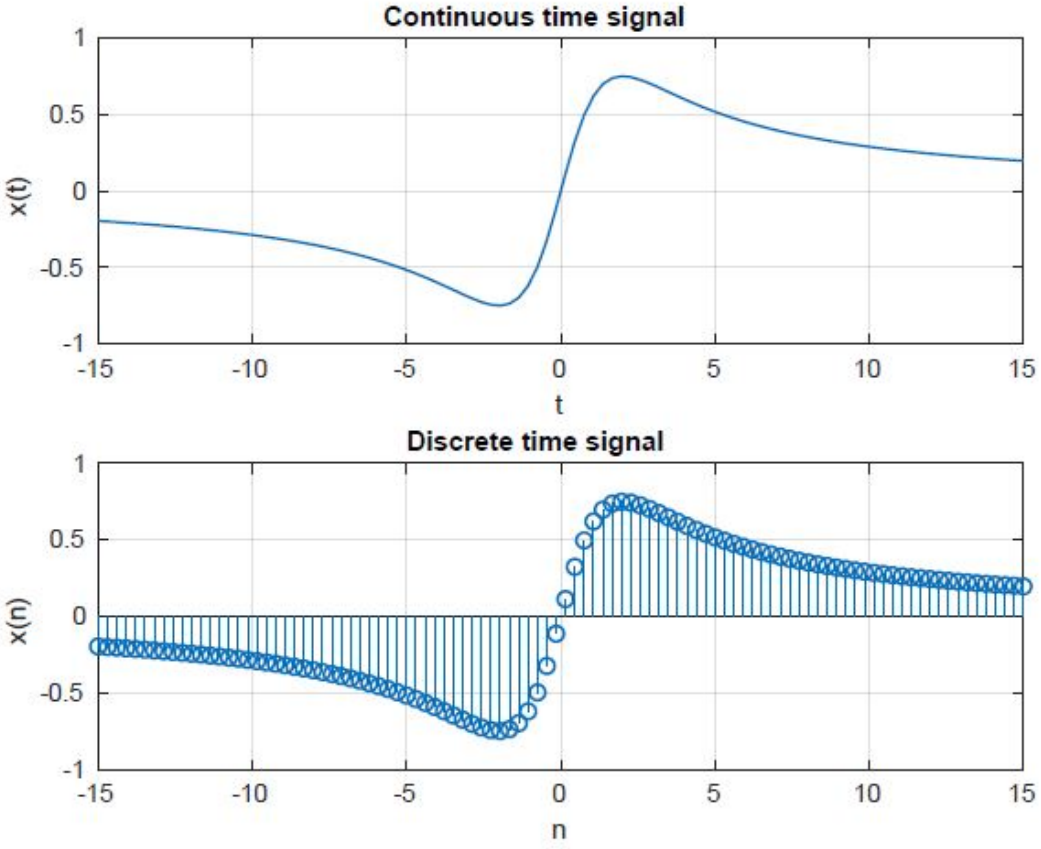
AIM :: Lab1 helps to revise the concept of signal and systems. Solving the properties of signal like addition,multiplication,shifting,folding and unit ramp,impluse and step signal in MATLAB . that's clearly shown in plots to understand better.

1. Generate deterministic continuous time signal having equation $x(t) = 3t/(4 + t^2)$ and discrete time signal having equation $x(n) = 3n/(4 + n^2)$. Use subplot command to display both figures
   Solution Problem-1

   (a) Matlab Script:

```
% To close all the previous figures
clc;
close all;

x=linspace(-15,15);          % To Generate a linearly spaced vectors
y=f(x);                      % function call for given equation

subplot(2,1,1);              % first figure of continuous signal
plot(x,y, 'linewidth', 1);   % To plot the continuous signal
grid on;
xlabel('t');
ylabel('x(t)');
title('Continuous time signal');

subplot(2,1,2);              % second figure of discrete signal
stem(x,y, 'linewidth', 1);   % To plot the discrete signal
grid on;
xlabel('n');
ylabel('x(n)');
title('Discrete time signal');

% function to plot given equation
function y=f(x)
y=(3*x)./(4+x.^2);
end
```

   (b) Simulation Output:



(c) Approach:
Created function to calculate signal for given vector (-15,15),which results continuous signal that I directly plotted with the use of 'plot' function . 'stem' function uses continuous signal and plots discrete signal for those vector.

2. Plot the continuous and discrete time sinusoidal wave for given amplitude, frequency, phase and sampling frequency.
Solution Problem-2

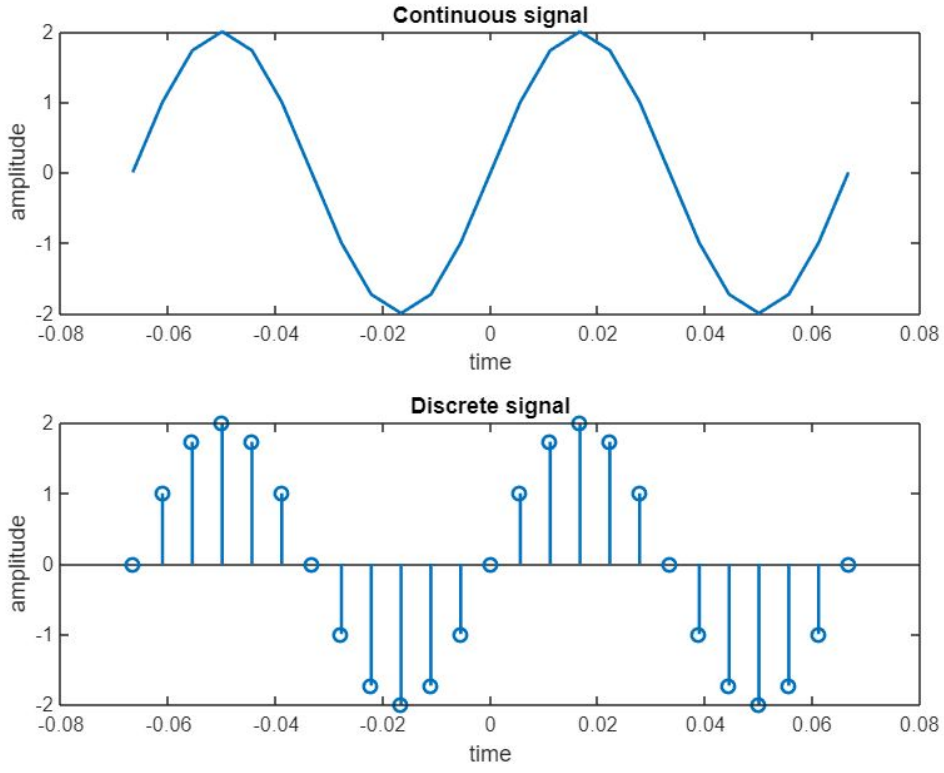
(a) Matlab Script:

```
clc;
close all;
%user has to give input of following values
amplitude = input('Enter the amplitude:: ');
freq= input('Enter the frequency:: ');
phase = input(' Enter the phase:: ');
samplingfreq = input('Enter the sampling frequency:: ');

t=(-1/freq:1/(samplingfreq):1/freq);        % to Generate a linearly spaced
    vectors  ,time=1/frequency

x= 2*pi*freq*t+ phase                       % sinusoidal(2pift+Phase)
wave= amplitude*sin(x);

subplot(2,1,1);
plot(t,wave,'LineWidth',1.5);               %plotting continuous signal
xlabel('time');
ylabel('amplitude');
title('Continuous signal');
```

```
19
20 subplot(2,1,2)
21 stem(t,wave,'LineWidth',1.5);                %plotting discrete signal
22 xlabel('time');
23 ylabel('amplitude');
24 title('Discrete signal');
```

(b) Simulation Output:

```
>> A1_2
Enter the amplitude::
2
Enter the frequency::
15
 Enter the phase::
0
Enter the sampling frequency::
180
```



(c) Approach:
Taking input from user for amplitude,phase and frequencies . I have calculated 2*π*freq*time +phase of sinusoidal wave. I directly plotted with the use of 'plot' function . 'stem' function uses continuous signal and plots discrete signal for those vector.

3. Generate the function for signal addition. Add two sequences x1(n) =$\{1, -1, 2, 5, 1, 5, -1\}$

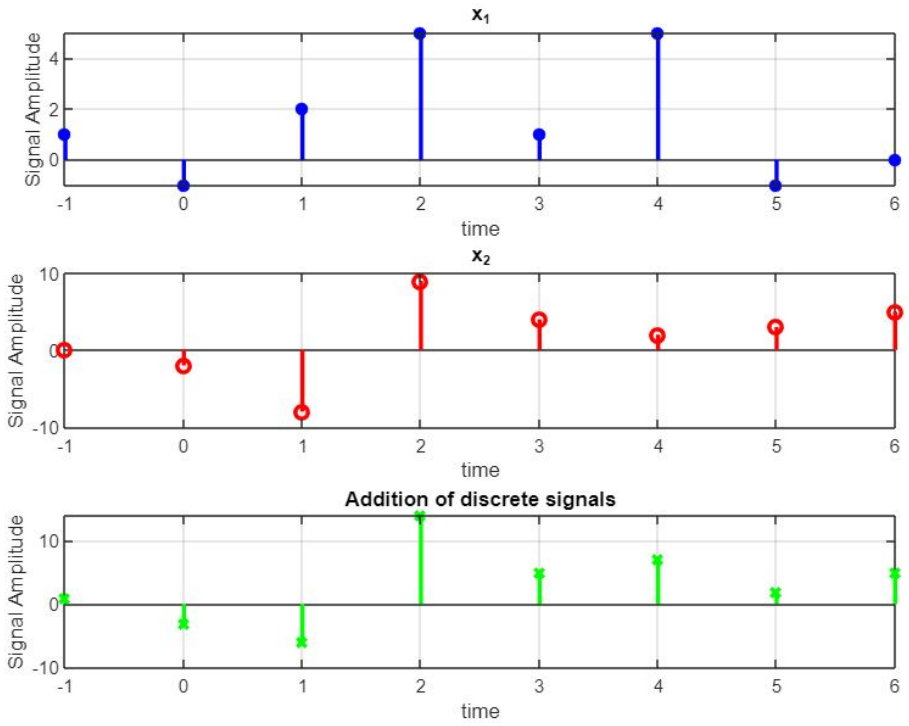
and x2(n) $=\{-2,-8,9,4,2,3,5\}$.

Solution Problem-3

(a) Matlab Script:

```
clc;
close all ;
x1=input( 'Enter x1:: ' );
zp1=input( 'Enter index to zero for signal 1:: ' );
x2=input( 'Enter x2:: ' );
zp2=input( 'Enter index to zero for signal 2:: ' );

x3_left=max(zp1-1, zp2-1);                    % elements to the left of zero
x3_right=max(length(x1)-zp1, length(x2)-zp2);% elements to the right of zero
x3_length=x3_left+x3_right+1;                 % elements of the sum x3
y1=zeros(1,x3_length);                        % new vector with zeros padded to x1
y2=zeros(1,x3_length);                        % new vector with zeros padded to x2

% inserting values from x1 to y1
ypointer=max(1, zp2-zp1+1);
xpointer=1;
for n=1:length(x1)
    y1(ypointer)=x1(xpointer);
    ypointer=ypointer+1;
    xpointer=xpointer+1;
end
% inserting values from x2 to y2
ypointer=max(1, zp1-zp2+1);
xpointer=1;
for n=1:length(x2)
    y2(ypointer)=x2(xpointer);
    ypointer=ypointer+1;
    xpointer=xpointer+1;
end

y3=y1+y2;                                     % output vector x3 is sum of x1,x2
zp3=max(zp1, zp2);
x=linspace(-x3_left, x3_right, x3_length);
xticks(-x3_left:1:x3_right);
subplot(3,1,1);
stem(x, y1, 'b*' , 'linewidth' , 2);
grid on ;
xlabel( 'time' );
ylabel( 'Signal Amplitude' );
title( 'x_1' );
hold on ;
subplot(3,1,2);
stem(x, y2, 'ro' , 'linewidth' , 2);
grid on ;
xlabel( 'time' );
ylabel( 'Signal Amplitude' );
title( 'x_2' );
hold on ;
subplot(3,1,3);
stem(x, y3, 'gX' , 'linewidth' , 2);
grid on ;grid on ;
xlabel( 'time' );
ylabel( 'Signal Amplitude' );
title( 'Addition of discrete signals' );

```

(b) Simulation Output:



(c) Approach:
firstly,I've calculated max of left sided elements between x1 and x2, zero pointer value has left to right numbered so subtracting one from it gives rest number of the left sided elements.And max of subtracting from length zero pointer gives right sided elements. Then taken two (y1,y2)arrays having zero and length of (left sided +right sided + 1)elements
x1 along with its right index shifted values to y1,same for y2 and another indexes have to zero .After doing summation operation of y1 and y2 ,plotted three figures y1,y2 and x3(sum)

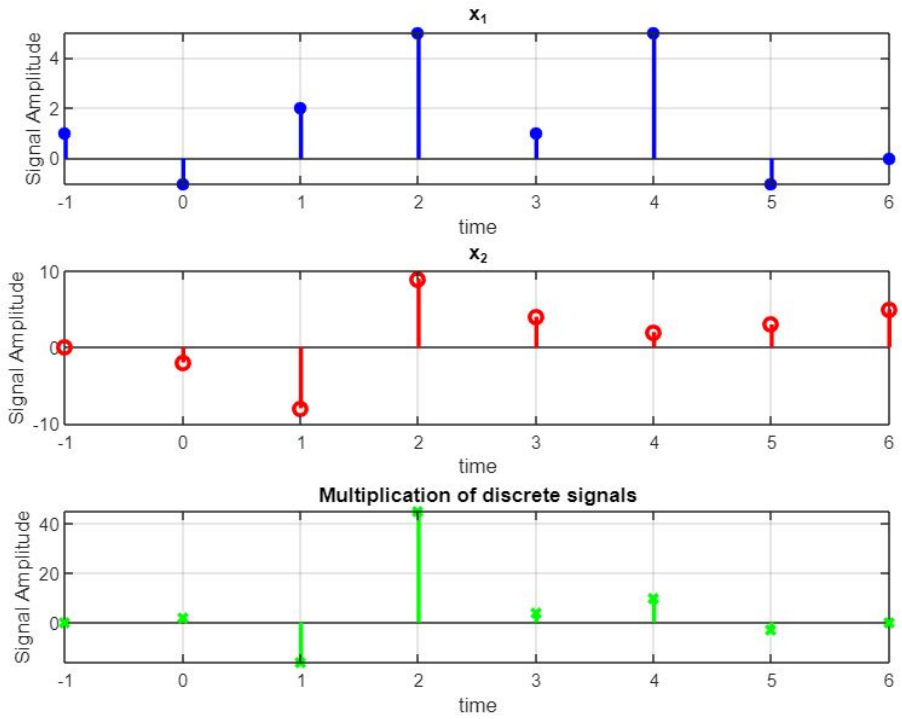
4. Generate the function for signal multiplication. Add two sequences x1(n) $=\{1,-1,2,5,1,5,-1\}$ and x2(n) $=\{-2,-8,9,4,2,3,5\}$.
Solution Problem-4

(a) Matlab Script:

```
clc;
close all ;
x1=input( 'Enter x1:: ' );
zp1=input( 'Enter index to zero for signal 1:: ' );
x2=input( 'Enter x2:: ' );
zp2=input( 'Enter index to zero for signal 2:: ' );

x3_left=max(zp1-1, zp2-1);                    % elements to the left of zero
x3_right=max(length(x1)-zp1, length(x2)-zp2);% elements to the right of zero
x3_length=x3_left+x3_right+1;                 % elements of the sum x3
y1=zeros(1,x3_length);                        % new vector with zeros padded to x1
y2=zeros(1,x3_length);                        % new vector with zeros padded to x2

% inserting values from x1 to y1
ypointer=max(1, zp2-zp1+1);
xpointer=1;
for n=1:length(x1)
    y1(ypointer)=x1(xpointer);
    ypointer=ypointer+1;
```

```
    xpointer=xpointer+1;
end
% inserting values from x2 to y2
ypointer=max(1, zp1-zp2+1);
xpointer=1;
for n=1:length(x2)
    y2(ypointer)=x2(xpointer);
    ypointer=ypointer+1;
    xpointer=xpointer+1;
end

y3=y1.*y2;                                  % output vector x3 is mul of x1,x2
zp3=max(zp1, zp2);
x=linspace(-x3_left, x3_right, x3_length);
xticks(-x3_left:1:x3_right);
subplot(3,1,1);
stem(x, y1, 'b*' , 'linewidth' , 2);
grid on ;
xlabel( 'time' );
ylabel( 'Signal Amplitude' );
title( 'x_1' );
hold on ;
subplot(3,1,2);
stem(x, y2, 'ro' , 'linewidth' , 2);
grid on ;
xlabel( 'time' );
ylabel( 'Signal Amplitude' );
title( 'x_2' );
hold on ;
subplot(3,1,3);
stem(x, y3, 'gX' , 'linewidth' , 2);
grid on ;grid on ;
xlabel( 'time' );
ylabel( 'Signal Amplitude' );
title( 'Multiplication of discrete signals' );

```

(b) Simulation Output:



(c) Approach:
firstly,I've calculated max of left sided elements between x1 and x2, zero pointer value has left to right numbered so subtracting one from it gives rest number of the left sided elements.And max of subtracting from length zero pointer gives right sided elements. Then taken two (y1,y2)arrays having zero and length of (left sided +right sided + 1)elements
x1 along with its right index shifted values to y1,same for y2 and another indexes have to zero .After doing multipllication operation of y1 and y2 ,plotted three figures y1,y2 and x3(mul)

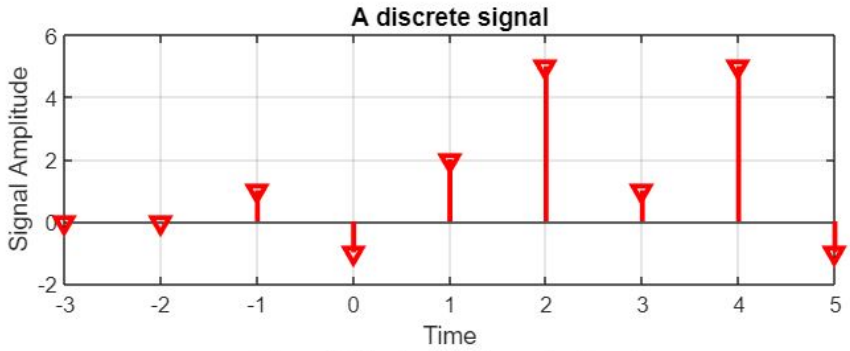
5. Generate the function for timing shifting. For sequences x(n) $=\{1, -1, 2, 5, 1, 5, -1\}$.
Solution Problem-5

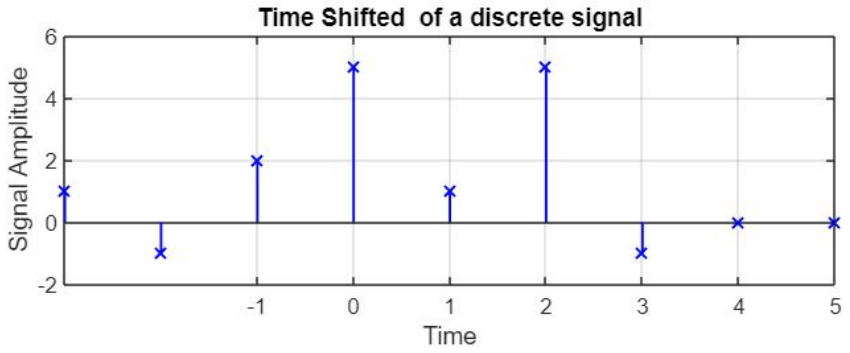
(a) Matlab Script:

```
%if value is -ve then shift signal to right or +ve shift signal to left .
clc;
close all ;
x1=input( 'Enter x1: ' );
zp1=input( 'Enter index to zero for x1:: ' );
shiftValueK=input('Enter shift value: ');

x2_length=length(x1)+abs(shiftValueK);      %total length after shift operation
y1=zeros(1,x2_length);                      %y1 is for zero rest of the value
y2=zeros(1,x2_length);                      %y2 is for shifted one array
if shiftValueK>0                            %left shift
    y1pointer=shiftValueK+1;
    y2pointer=1;
    xpointer=1;
    for n=1:length(x1)
        y1(y1pointer)=x1(xpointer);
        y2(y2pointer)=x1(xpointer);
        y1pointer=y1pointer+1;
        y2pointer=y2pointer+1;
        xpointer=xpointer+1;
    end
elseif shiftValueK<0                        %right shift
    y1pointer=1;
    y2pointer=-shiftValueK+1;
    xpointer=1;
    for n=1:length(x1)
        y1(y1pointer)=x1(xpointer);
        y2(y2pointer)=x1(xpointer);
        y1pointer=y1pointer+1;
        y2pointer=y2pointer+1;
        xpointer=xpointer+1;
    end
end
zp2=zp1+shiftValueK;
x=linspace(1-zp2, length(y2)-zp2, length(y2));
xticks(-x3_left:1:x3_right);9
subplot(2,1,1);
stem(x, y1, 'rV' , 'linewidth' , 2);
grid on ;
xlabel( 'Time' );
ylabel( 'Signal Amplitude' );
title( 'A discrete signal' );
hold on ;
subplot(2,1,2)
stem(x, y2, 'bX' , 'linewidth' , 1);
hold on ;
grid on ;
xlabel( 'Time' );
ylabel( 'Signal Amplitude' );
title( 'Time Shifted  of a discrete signal' );

```

(b) Simulation Output:

```
>> A1_5
Enter x1:
[1 -1 2 5 1 5 -1]
Enter index to zero for x1::
2
Enter shift value:
2
>>
```





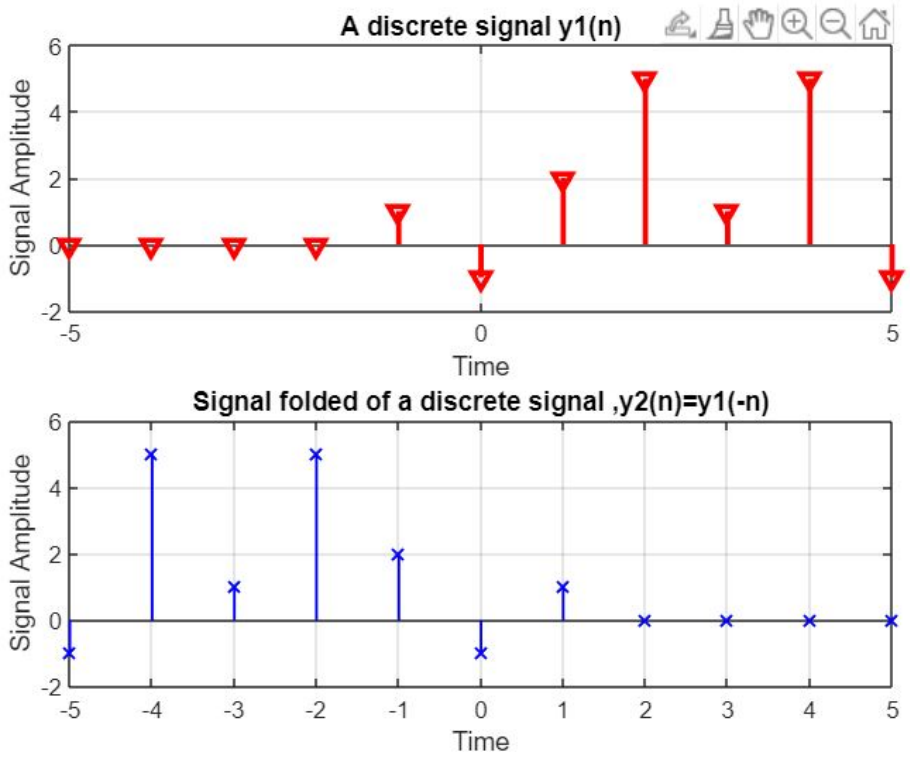
(c) Approach:
I've calculated length of output array which is length of input array + absolute of shift value. with the help of this length zero padding into rest of the empty indexes. Checking the condition of shifting value whether it's +ve ,then signal shifts to left and hence index pointer (y1pointer) starts from shift value.if -ve ,then signal shifts to right and index pointer(y1pointer) starts from negative value of shift.Plotted discrete signal for input and shifted array.

6. Generate the function for signal folding. For sequences x(n) $=\{1, -1, 2, 5, 1, 5, -1\}$.
Solution Problem-6

(a) Matlab Script:

```
clc;
close all ;
x1=input( 'Enter x1:: ' );
zp1=input( 'Enter index to zero for x1:: ' );
x1_right=length(x1)-zp1;                    %right sided elements
x1_left=zp1-1;                              %left sided elements
y2maxofRL=max(x1_right,x1_left);          %max of left or right
y2_length=2*y2maxofRL+1;                  %length of output folded array
y1=zeros(1,y2_length);
y2=zeros(1,y2_length);

zp2=max(x1_right,x1_left)+1;
y1pointer=zp2+1;
y2pointer=zp2-1;
xpointer=zp1+1;
for n=1:x1_right
    y1(y1pointer)=x1(xpointer);
    y2(y2pointer)=x1(xpointer);
    y1pointer=y1pointer+1;
    y2pointer=y2pointer-1;
    xpointer=xpointer+1;
end
y1pointer=zp2-1;
y2pointer=zp2+1;
xpointer=zp1-1;
for n=1:x1_left
    y1(y1pointer)=x1(xpointer);
    y2(y2pointer)=x1(xpointer);
    y1pointer=y1pointer-1;
    y2pointer=y2pointer+1;
    xpointer=xpointer-1;
end
y1(zp2)=x1(zp1);
y2(zp2)=x1(zp1);

x=linspace(-max(x1_left, x1_right), max(x1_left, x1_right), length(y2));
xticks(-y2maxofRL:1:y2maxofRL);
subplot(2,1,1)
stem(x, y1, 'rV' , 'linewidth' , 2);
grid on ;
xlabel( 'Time' );
ylabel( 'Signal Amplitude' );
title( 'A discrete signal y1(n)' );
% Plot command to plot the discrete function y1
hold on ;
subplot(2,1,2)
stem(x, y2, 'bX' , 'linewidth' , 1);
grid on ;
xlabel( 'Time' );
ylabel( 'Signal Amplitude' );
title( 'Signal folded of a discrete signal ,y2(n)=y1(-n)');

```

(b) Simulation Output:



(c) Approach:
Firstly, calculated right sided and left sided elements and length of output array will be max of one elements + 1 itself. Then I've shifted right sided elements to left side with the help of length value from first index to zero pointer and left sided elements to right side with the help of zero pointer to length of array.

7. Generate the function for time multiplication. For sequences x(n) $=\{1, -1, 2, 5, 1, 5, -1\}$.

Solution Problem-7

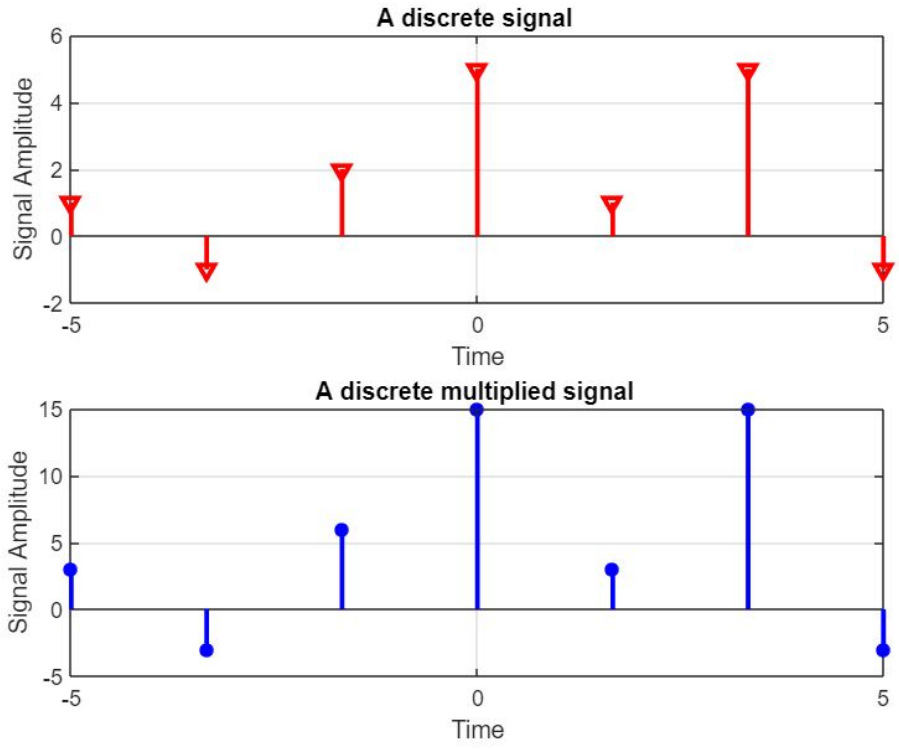
(a) Matlab Script:

```
clc;
close all ;
x1=input( 'Enter x1:: ' );
zp1=input( 'Enter index to zero for x1:: ' );
multipler=input('Enter value of multipler:: ');
x2=zeros(1,length(x1))
x1_right=length(x1)-zp1;
x1_left=zp1-1;
x=linspace(-max(x1_left, x1_right), max(x1_left, x1_right), length(x1));

for n=1:length(x1)
    x2(n)=multipler.*x1(n);
end

subplot(2,1,1)
stem(x, x1, 'rV' , 'linewidth' , 2);
grid on ;
xlabel( 'Time' );
ylabel( 'Signal Amplitude' );
title( 'A discrete signal ' );
hold on ;
subplot(2,1,2)
stem(x, x2, 'b*' , 'linewidth' , 2);
grid on ;
```

```
xlabel( 'Time' );
ylabel( 'Signal Amplitude' );
title( 'A discrete multiplied signal ' );
hold on ;
```

(b) Simulation Output:

```
Enter x1::
[1 -1 2 5 1 5 -1]
Enter index to zero for x1::
2
Enter value of multipler::
3
```



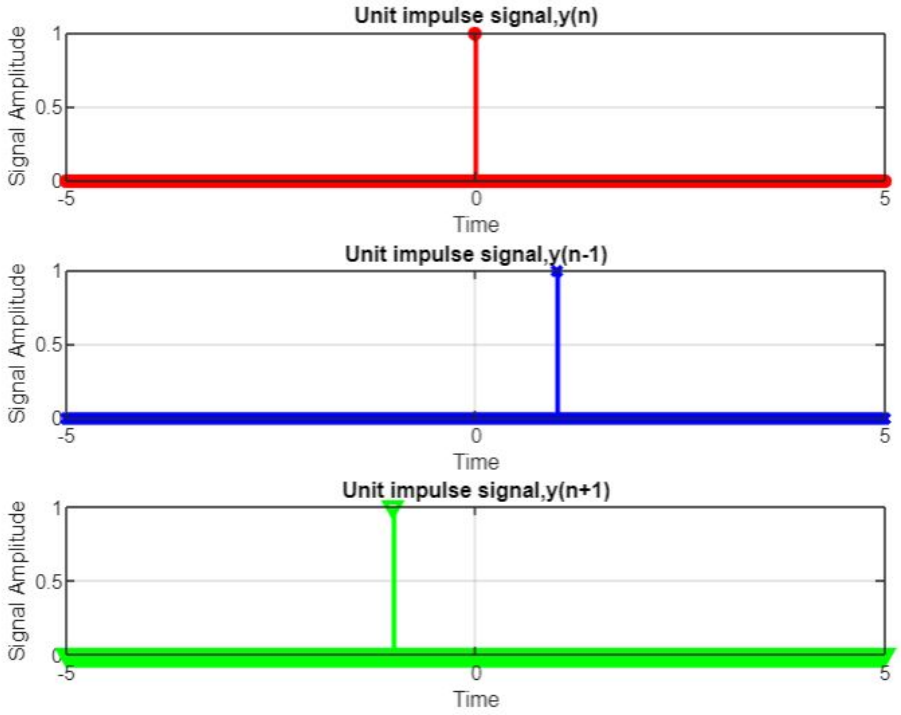
(c) Approach:
I've Calculated length using given array and multiply constant with each element using for loop.

8. Generate function for unit sample signal $\delta(n)$ Also plot $\delta(n-1)$ and $\delta(n+1)$.
Solution Problem-8

(a) Matlab Script:

```
clc;
close all ;
x=-5:0.01:5;
y1=zeros(1,length(x));
y2=zeros(1,length(x));
y3=zeros(1,length(x));
%func call for unit sample signal
for i=1:length(x)
    y1(i)=UnitImpulse(x(i));
    y2(i)=UnitImpulse(x(i)-1);
    y3(i)=UnitImpulse(x(i)+1);
end
xticks(-5:0.01:5);
subplot(3,1,1);
stem(x, y1, 'r*' , 'linewidth' , 2);
grid on ;
xlabel( 'Time' );
ylabel( 'Signal Amplitude' );
title( 'Unit impulse signal,y(n)' );
hold on ;
subplot(3,1,2);
stem(x, y2, 'bX' , 'linewidth' , 2);
grid on ;
xlabel( 'Time' );
ylabel( 'Signal Amplitude' );
title( 'Unit impulse signal,y(n-1)' );
hold on ;
subplot(3,1,3);
stem(x, y3, 'gV' , 'linewidth' , 2);
grid on ;
xlabel( 'Time' );
ylabel( 'Signal Amplitude' );
title( 'Unit impulse signal,y(n+1)' );
%function to calc unit impluse if value is equal to 0 then return 1 elsewhere 0
function z = UnitImpulse(x)
    if x == 0
        z=1;
    else
        z=0;
    end
end

```

(b) Simulation Output:



(c) Approach:
passing the vale of x in parameter and calculate output of Unit impluse signal.If value is 0 to then return 1 elesewhere 0 .

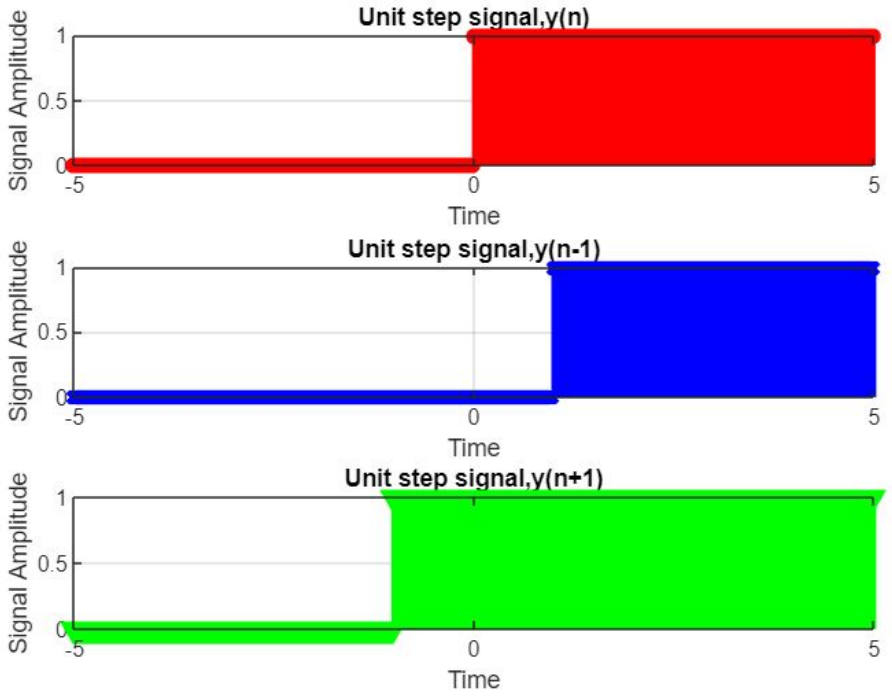
9. Generate function for unit step signal $u(n)$ Also plot $u(n-1)$ and $u(n+1)$.
Solution Problem-9

(a) Matlab Script:

```
clc;
close all ;
x=-5:0.01:5;
y1=zeros(1,length(x));
y2=zeros(1,length(x));
y3=zeros(1,length(x));
%func call for unit step signal
for i=1:length(x)
    y1(i)=UnitStep(x(i));
    y2(i)=UnitStep(x(i)-1);
    y3(i)=UnitStep(x(i)+1);
end
xticks(-5:0.01:5);
subplot(3,1,1);
stem(x, y1, 'r*' , 'linewidth' , 2);
grid on ;
xlabel( 'Time' );
ylabel( 'Signal Amplitude' );
title( 'Unit step signal,y(n)' );
hold on ;
subplot(3,1,2);
stem(x, y2, 'bX' , 'linewidth' , 2);
grid on ;
xlabel( 'Time' );
ylabel( 'Signal Amplitude' );
title( 'Unit step signal,y(n-1)' );
hold on ;
subplot(3,1,3);
stem(x, y3, 'gV' , 'linewidth' , 2);
```

```
grid on ;
xlabel( 'Time' );
ylabel( 'Signal Amplitude' );
title( 'Unit step signal,y(n+1)' );
%function to calc unit step if value is greater or equal to 0 then return 1
    elsewhere 0
function z = UnitStep(x)
    if x >= 0
        z=1;
    else
        z=0;
    end
end

```

(b) Simulation Output:



(c) Approach:
passing the vale of x in parameter and calculate output of Unit impluse signal if value is greater or equal to 0 then return 1 elsewhere 0.

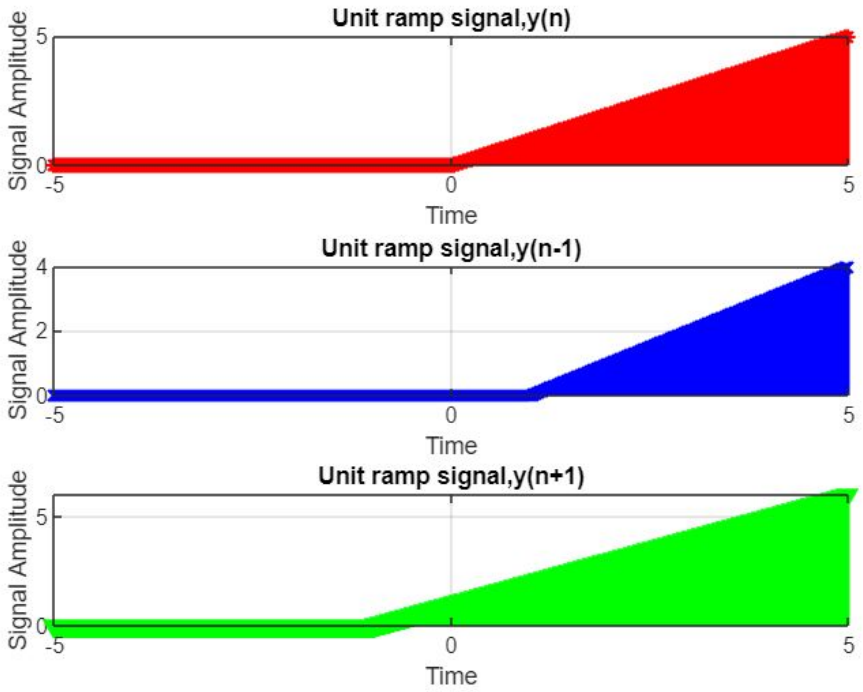
10. Generate function for unit ramp signal $U_r(n)$ Also plot $U_r(n-1)$ and $U_r(n+1)$.
Solution Problem-10

(a) Matlab Script:

```
clc;
close all ;
x=-5:0.01:5;
y1=zeros(1,length(x));
y2=zeros(1,length(x));
y3=zeros(1,length(x));
%func call for unit ramp signal
for i=1:length(x)
    y1(i)=UnitRamp(x(i));
    y2(i)=UnitRamp(x(i)-1);
    y3(i)=UnitRamp(x(i)+1);
end
```

```
xticks(-5:0.1:5);
subplot(3,1,1);
stem(x, y1, 'r*' , 'linewidth' , 1);
grid on ;
xlabel( 'Time' );
ylabel( 'Signal Amplitude' );
title( 'Unit ramp signal,y(n)' );
hold on ;
subplot(3,1,2);
stem(x, y2, 'bX' , 'linewidth' , 1);
grid on ;
xlabel( 'Time' );
ylabel( 'Signal Amplitude' );
title( 'Unit ramp signal,y(n-1)' );
hold on ;
subplot(3,1,3);
stem(x, y3, 'gV' , 'linewidth' , 1);
grid on ;
xlabel( 'Time' );
ylabel( 'Signal Amplitude' );
title( 'Unit ramp signal,y(n+1)' );
%function to calc unit ramp if value is greater or equal to 0 then return that
    value to signal elsewhere 0
function z = UnitRamp(x)
    if x >= 0
        z=x;
    else
        z=0;
    end
end

```

(b) Simulation Output:



(c) Approach:
passing the vale of x in parameter and calculate output of Unit impluse signal if value is greater or equal to 0 then return x elsewhere 0.

11. Plot all the given signals and comment on their output for periodicity writing common

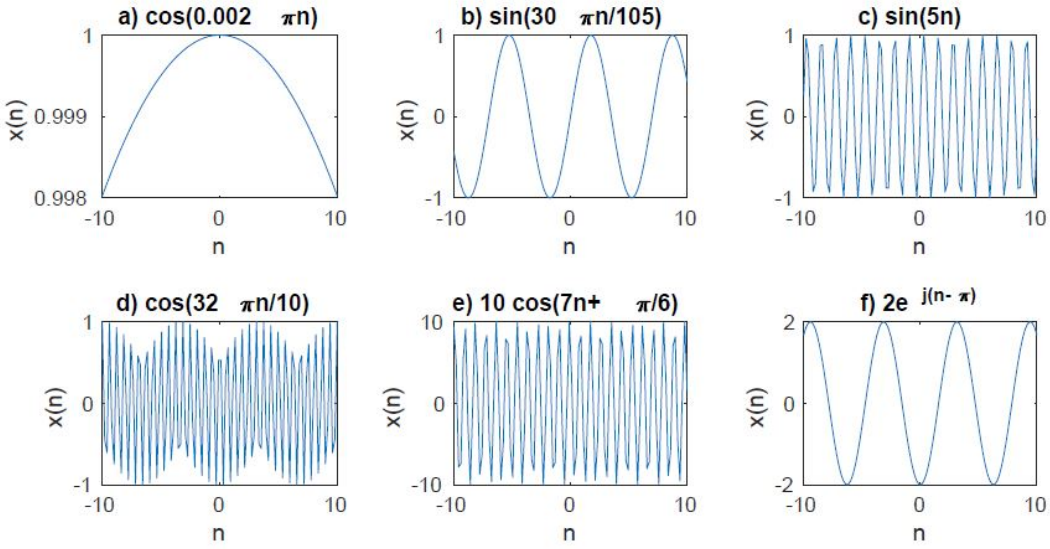
MATLAB code.

Solution Problem-11

(a) Matlab Script:

```
clc;
clear all;
n = linspace(-10,10);
x1=cos(0.002.*pi.*n);
x2=sin(30.*pi.*n./105);
x3=sin(5.*n);
x4=cos(32.*pi.*n./10);
x5=10.*cos((7.*n )+(pi./6));
x6=2.*exp(1j.*(n-pi));
t=tiledlayout(3,3);
nexttile;
plot(n,x1);
title('a) cos(0.002\pin)');
xlabel('n');
ylabel('x(n)');
nexttile;
plot(n,x2);
title('b) sin(30\pin/105)');
xlabel('n');
ylabel('x(n)');
nexttile;
plot(n,x3);
title('c) sin(5n)');
xlabel('n');
ylabel('x(n)');
nexttile;
plot(n,x4);
title('d) cos(32\pin/10)');
xlabel('n');
ylabel('x(n)');
nexttile;
plot(n,x5);
title('e) 10 cos(7n+\pi/6)');
xlabel('n');
ylabel('x(n)');
nexttile;
plot(n,x6);
title('f) 2e^{j(n-\pi)}');
xlabel('n');
ylabel('x(n)');

```

(b) Simulation Output:



(c) Approach:
To plot these signal,i've taken time interval and pass it to sin or cos .Plotted every plot using tile function .
Periodicity ::
a)$2\pi/0.002\pi$ =1000
b)$(2\pi * 105)/(30 * \pi)$ =7
c)$2\pi/5$ =1.256
d)$(2 * \pi * 10)/(32 * \pi)$ =0.625
e)$2\pi/7$ =0.897
f)0-not periodic

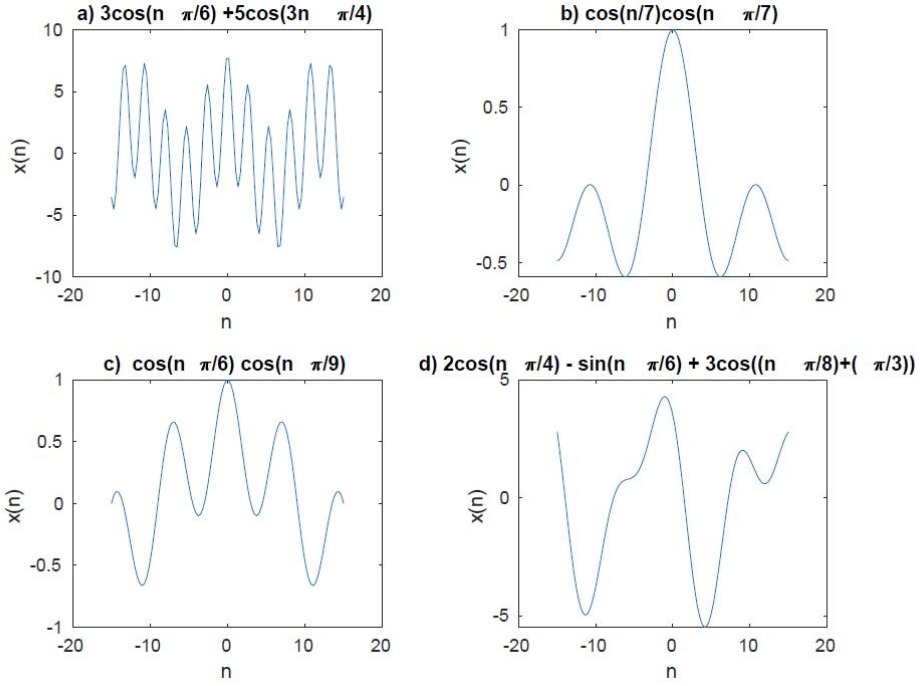
12. Plot all the given signals and comment on their output for periodicity writing common MATLAB code.
Solution Problem-12

(a) Matlab Script:

```
clear;
clear all;
n= linspace(-15,15);
x1 = (3 .* cos( (n .*  pi) ./ 6)) + (5 .* cos(( 3 .* n .*  pi) ./ 4));
x2 = cos(n./7) .* cos( (n.*pi)./7);
x3 = cos((n.*pi)./6) .* cos( (n.*pi)./9);
x4 = (2.* cos((n.*pi)/4)) - sin((n.*pi)/6) + 3.*cos(((n.*pi)./8)+ (pi./3));
tiledlayout(2,2);
nexttile;
plot(n,x1);
title('a) 3cos(n\pi/6) +5cos(3n\pi/4)');
xlabel('n');
ylabel('x(n)');
nexttile;
plot(n,x2);
title('b) cos(n/7)cos(n\pi/7)');
xlabel('n');
ylabel('x(n)');
nexttile;
plot(n,x3);
title('c)  cos(n\pi/6) cos(n\pi/9)');
```

```
xlabel('n');
ylabel('x(n)');
nexttile;
plot(n,x4);
title('d) 2cos(n\pi/4) - sin(n\pi/6) + 3cos((n\pi/8)+(\pi/3))');
xlabel('n');
ylabel('x(n)');

```

(b) Simulation Output:



(c) Approach:
To plot these signal,i've taken time interval and pass it to sin or cos .Plotted every plot using tile function .
Periodicity ::
1)24
2)0-not periodic
3)36
4)48

13. Sample the sinusoid x = sin(2 pi f t), where f = 2 kHz, and plot the sampled signals over the continuous-time signal.
- Let x1 be the signal sampled at 10 kHz.
- Let x2 be the signal sampled at 3 kHz.

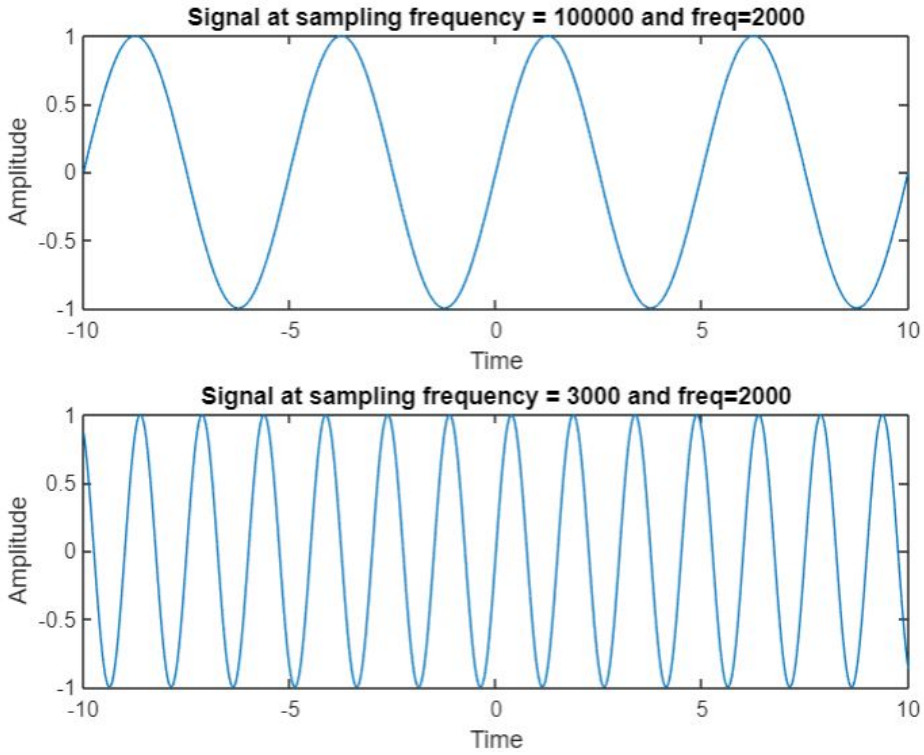
Solution Problem-13

(a) Matlab Script:

```
clc;
clear all;
f = 2000;                                   % frequnecy 2 kHZ
time = -10:0.01:10;                         % range
```

```
x1 = sin (2* pi*f/10000*time);                %Signal 1 at fs = 10,000
subplot (2,1,1);
plot (time ,x1)
xlabel ( 'Time' );
ylabel ( 'Amplitude' );
title ( 'Signal at sampling frequency = 100000 and freq=2000' )
x2 = sin (2* pi*f/3000*time);                 %Signal 2 at fs = 3000
subplot (2,1,2);
plot (time ,x2)
xlabel ( 'Time' );
ylabel ( 'Amplitude' );
title ( 'Signal at sampling frequency = 3000 and freq=2000' )

```

(b) Simulation Output:



(c) Approach:
Taken freq=2kHz plotting two diff sampling freq one is 3kHz and 2nd is 10kHz . Calculating $sin(2 * pi * f/f_s * time)$ and plotted using subplot.