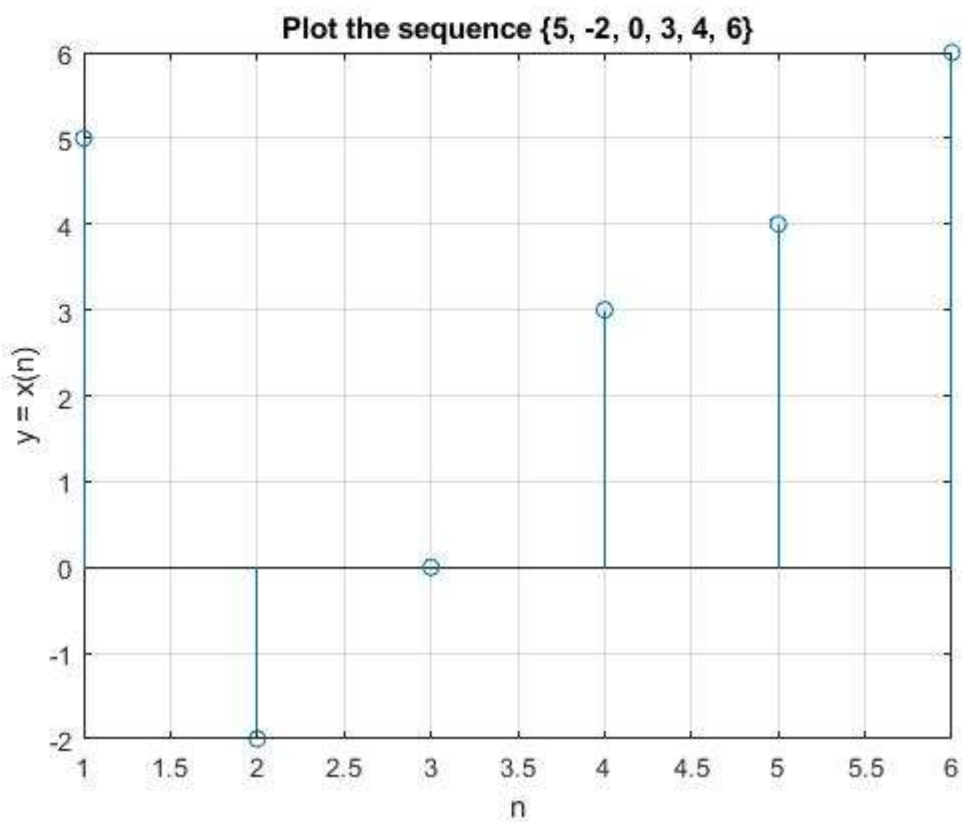
Task-1: Plot the sequence, {5, -2, 0, 3, 4, 6}

Solution:

```
%Plot the sequence, {5, -2, 0, 3, 4, 6}.

y = [5 -2 0 3 4 6];

stem(y);

title("Plot the sequence, {5, -2, 0, 3, 4, 6},'r'");

xlabel('n');

ylabel('y = x(n)');

grid on;
```

Output:



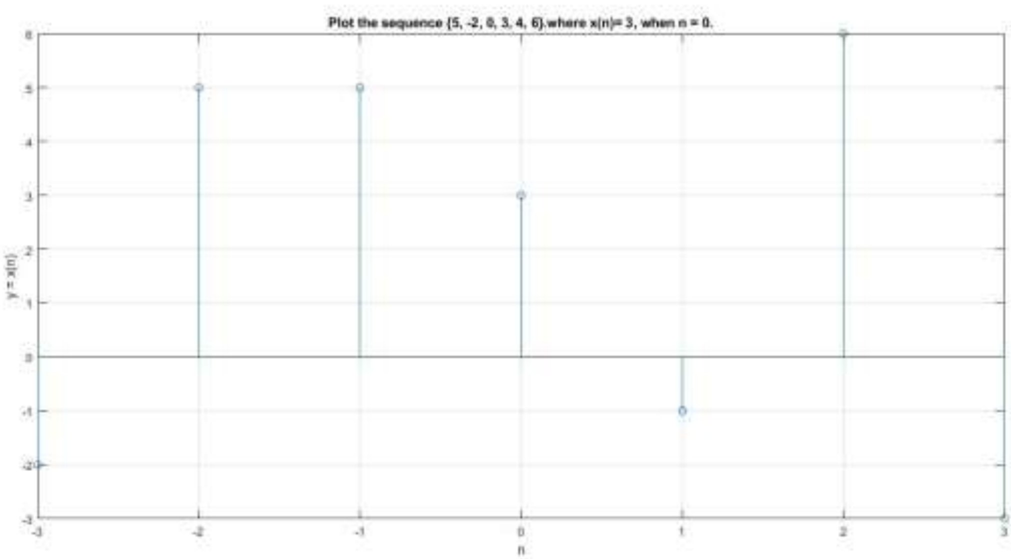
Observation: This is the discrete time sequence where n are the samples and x(n) are the corresponding discrete values. In where x-axis represents n samples and y-axis represents x(n) signal.

Task-1.1: Plot the sequence, x = {-2, 5, 5, 3, -1, 6, -3}

Solution:

```
%Plot the sequence {-2, 5, 5, 3, -1, 6, -3}.
%where x(n)= 3, when n = 0
y = [-2 5 5 3 -1 6  -3];

x = [-3:3]

stem(x,y)

xlabel('n')

ylabel('y = x(n)')

grid on
```

Output:



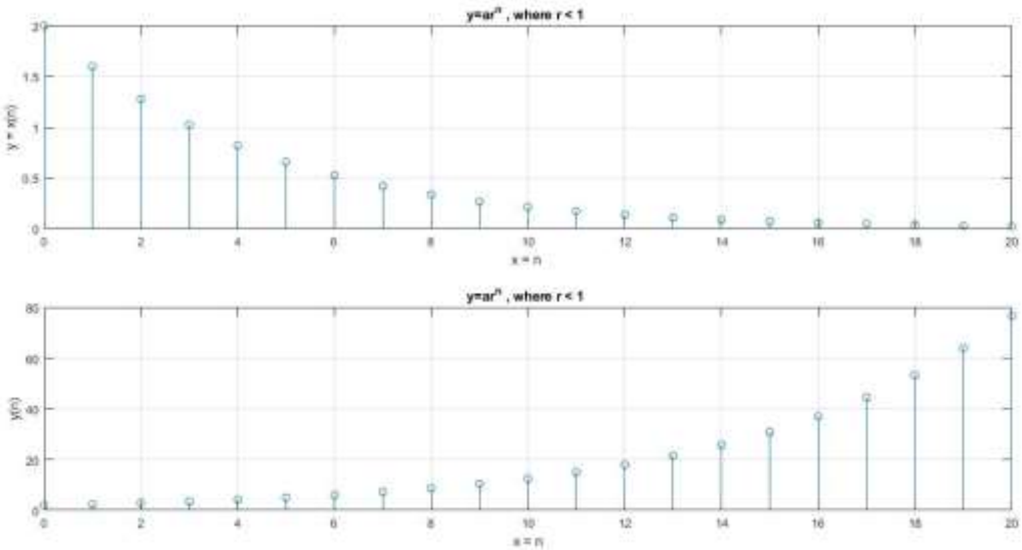
Observation: This is the discrete time sequence where n are the samples and x(n) are the corresponding discrete values. In where n = 0, and x(0) = 5. In where x-axis represents n samples and y-axis represents x(n) signal.

Task-2: Plot of the exponential sequence, $y(n) = ar^n$ for $a = 2$, $r = 0.8$ and 1.2

Solution:

```
a = 2;              % given
r = 0.8;            % given

n = 0 : 1 : 20;  % assume 0 to 20 [0 20] values
increasing by 1

y = a*r.^n;       % notice  r.^n

subplot(2, 1, 1);

stem(n , y);
title('y=ar^n , where r < 1 ');

xlabel('x = n ');

ylabel('y = x(n)');
grid on;

r = 1.2 ;
y = a*r.^n;

subplot(2,1, 2);
stem(n, y);
title('y=ar^n , where r < 1 ');

xlabel('x = n');

ylabel('y(n)');

grid on;
```

Output:



Observation: This is the exponential sequence $y(n) = ar^n$ for $a = 2$, $r = 0.8$ and 1.2 where n are the samples and y(n) are the corresponding discrete values. In where x-axis represents n samples and y-axis represents y(n) signal. When n is increased then y(n) is also increased exponentially.

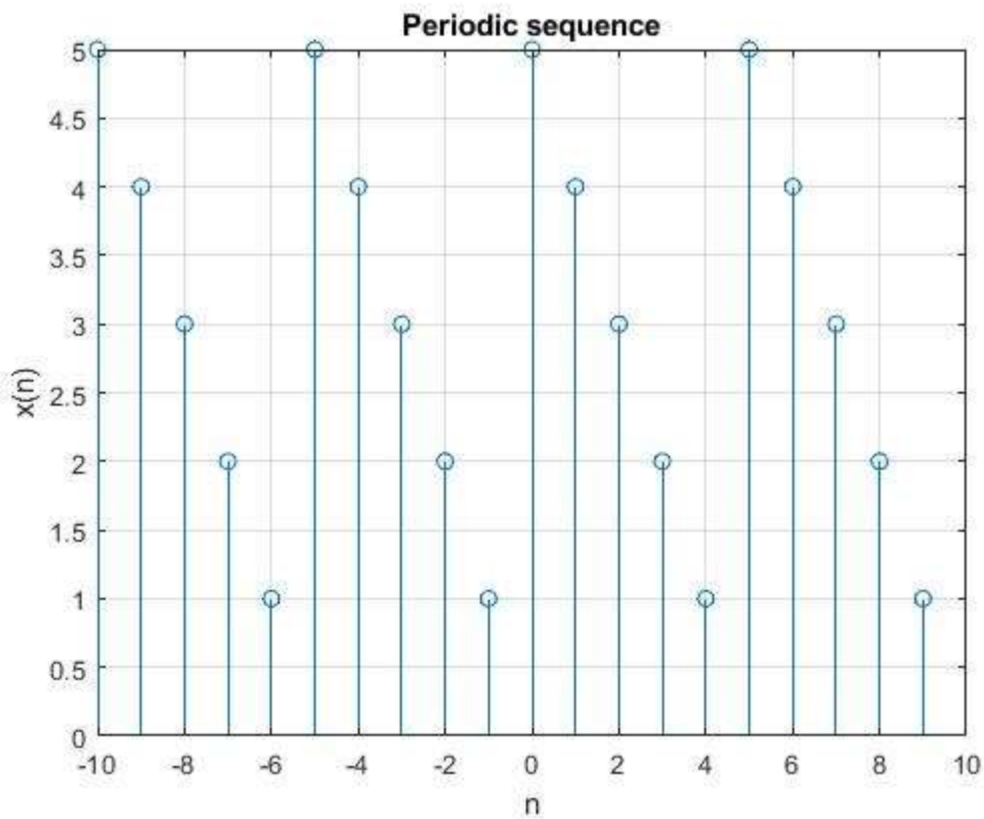
Task3: Plot the periodic sequence, $x[n]$ = {….5,4,3,2,1, 5,4,3,2,1, 5,4,3,2,1, 5,4,3,2,1……..} for n = -10 to 9.

Solution:

```
% Plot the periodic sequence,
% x[n] = {….5,4,3,2,1, 5,4,3,2,1, 5,4,3,2,1,
5,4,3,2,1……..}
% for n = -10 to 9
n = [-10:9 ];
x=[5,4,3,2,1];
p=4;
xy=x' * ones(1,p); % xy indicates, p=4 column of
vectors of [ 5, 4 , 3, 2 , 1]
xy =(xy(:)); %a long column vector will be converted
stem(n,xy);
```

```
title("Periodic sequence");
xlabel('n');
ylabel('x(n)');
grid on;
```

Output:



Observation: This is the periodic sequence *x*[*n*] = {….5,4,3,2,1, 5,4,3,2,1, 5,4,3,2,1, 5,4,3,2,1……..} for *n* = -10 to 9.where n are the samples and y(n) are the corresponding discrete values. In where x-axis represents n samples and y-axis represents y(n) signal. When n is increased then y(n) is also increased periodically.

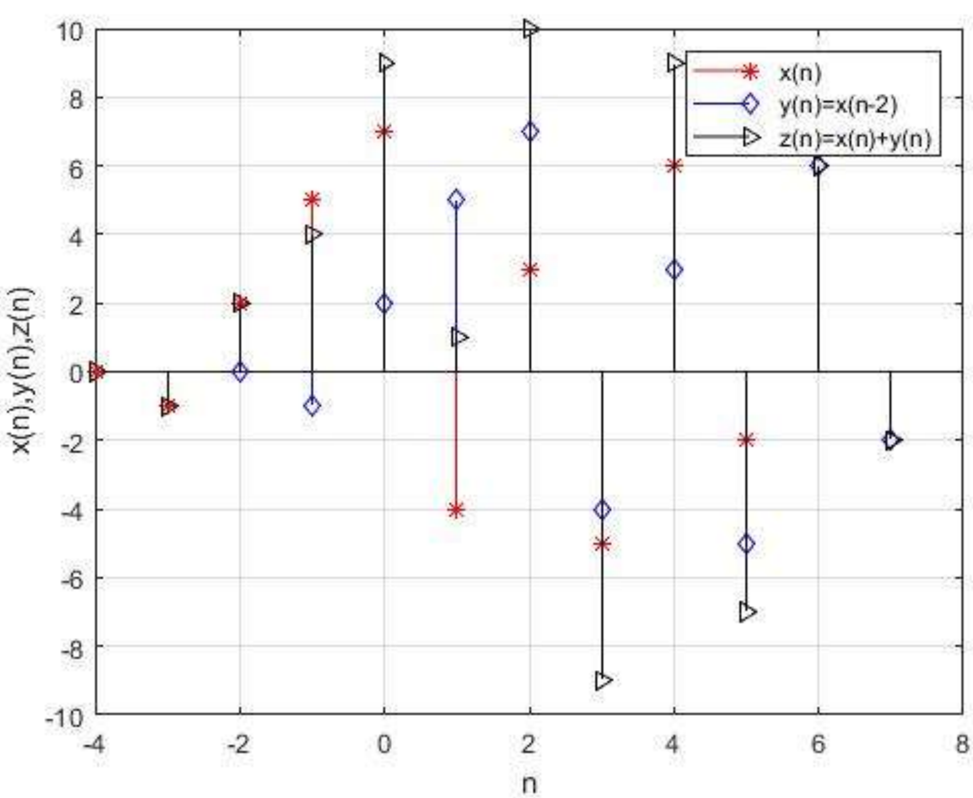
Task4 :Plot the sequence of *x*(*n*)={0, -1, 2, 5, 7, -4, 3, -5, 6, -2 }, *y*(*n*) = *x*(*n*-*2*), and *z*(*n*) = *x*(*n*)+*y*(*n*).

Solution:

```
n = -4:5;
x=[0 -1 2 5 7 -4 3 -5 6 -2];
```

```
stem(n,x,'r*');
hold on;
k=2;
m=n+k;
y=x;
stem(m,y,'bd');
hold on;
r=min(min(n),min(m)):max(max(n),max(m)); %duration of z
z1=zeros(1,length(r));
z2=z1;  %initialization
z1(find((r>=min(n))&(r<=max(n))==1))=x; %x with duration of z
z2(find((r>=min(m))&(r<=max(m))==1))=y; %y with duration of z
z=z1+z2;
stem(r,z,'k>');
grid on;
xlabel('n');
ylabel('x(n),y(n),z(n)')
legend('x(n)','y(n)=x(n-2)', 'z(n)=x(n)+y(n)')
```

Output:



Observation: This is the shifting sequence of *x*(*n*)={0, -1, 2, 5, 7, -4, 3, -5, 6, -2 }, *y*(*n*) = *x*(*n*-2), and *z*(*n*) = *x*(*n*)+*y*(*n*).where n are the samples and y(n) are the corresponding discrete values. In where x-axis represents n samples and y-axis represents y(n) signal. y(n) shifted 2 times to the right.

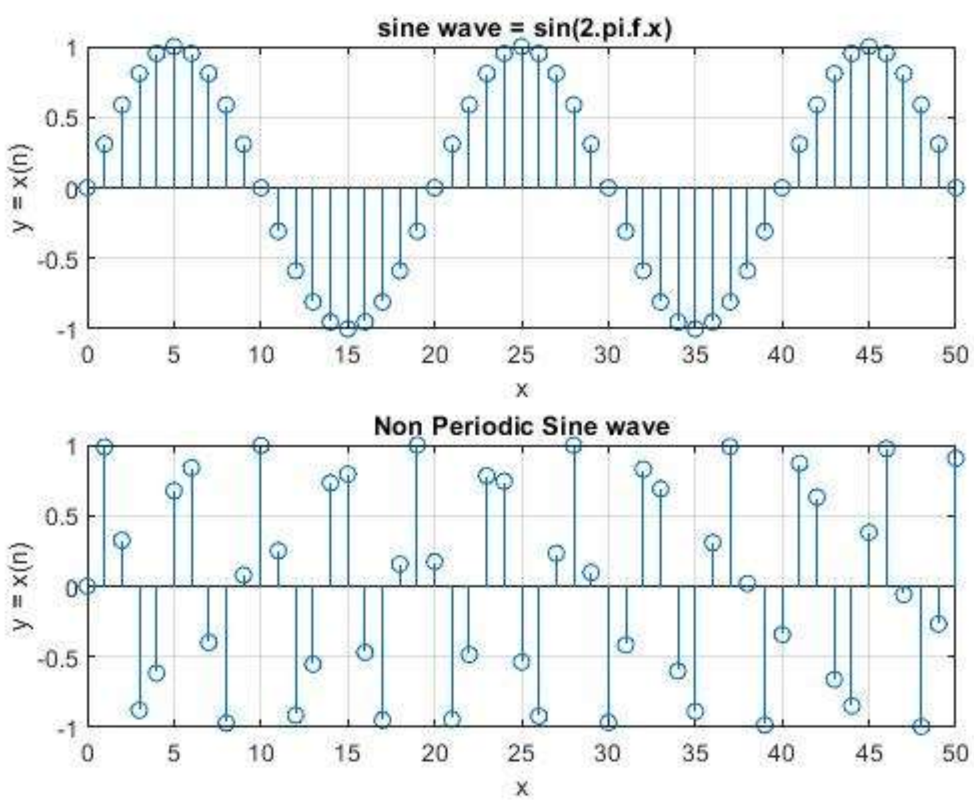
Task5: For discrete time sequence, a sinusoidal wave is periodic if its frequency is a rational number like, *f* = *M* / *K*. Period of the wave is the denominator i.e. *K*.

Solution:

```
% For discrete time sequence, a sinusoidal wave is
periodic
% if its frequency is a rational number like,
% f = M / K. Period of the wave is the denominator i.e.
K.

% periodic sinusoidal wave
x = [0 : 1 : 50]; % assume 0 to 50 descrete values
increasing by 1

N = 20;           % assume cycle completed when x =  20
,
                  % after 20, next cycle occurs, so
period = 20
                  % so N = period
f = 1/N;

y = sin(2 * pi * f * x); % sine wave = sin(2.pi.f.x)

subplot(2,1,1);

stem(x, y);

title(' sine wave = sin(2.pi.f.x)');

xlabel('x');

ylabel('y = x(n)');
```

```
grid on;

% Non-periodic sinusoidal wave

N = sqrt(20);  % root(N) = irrational so it produces
non periodic

f= 1/N;

y = sin(2*pi*f*x);

subplot(2,1,2);

stem(x,y);

title('Non Periodic Sine wave');

xlabel('x');

ylabel('y = x(n)');

grid on;
```

Output:



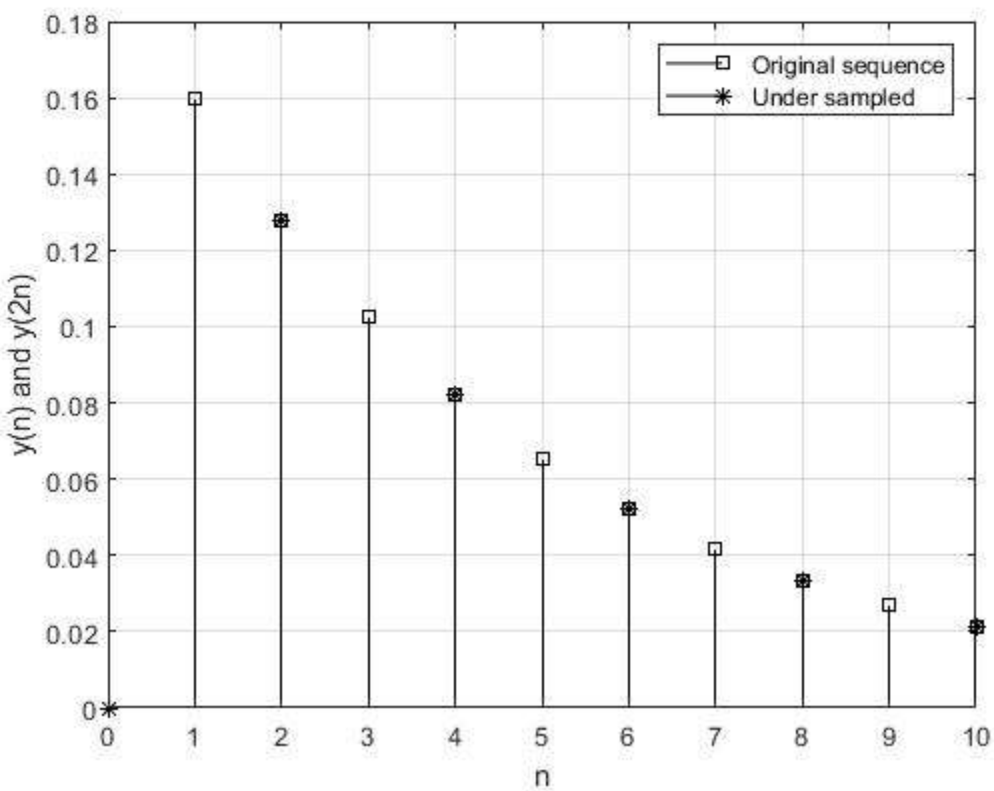
Observation: This is the discrete time sequence of x(n). As we know a sinusoidal wave is periodic if its frequency is a rational number like, $f = M / K$. Period of the wave is the denominator i.e. K. Where n are the samples and y(n) are the corresponding discrete values. In where x-axis represents n samples and y-axis represents y(n) signal.

Task6: The sampling rate of $y(2n)$ is half the rate of $y(n)$ therefore y(2n) is called the under sampled version of $y(n)$. Let us verify the phenomenon.

Solution:

```
a=0.2;
r=0.8;
N=10;
for n=1:N,
s(n)=n;
y(n)=a*r.^n;
end
M=N/2;
for m=1:M,
m=2*m;
p(m)=m;
z(m)= y(m);
end
stem(s,y,'ks')%Original sequence y(n)
hold on
stem(p,z,'k*')%under sampled sequence y(2n)
xlabel('n')
ylabel('y(n) and y(2n)')
grid on
legend('Original sequence', 'Under sampled')
```

Output:



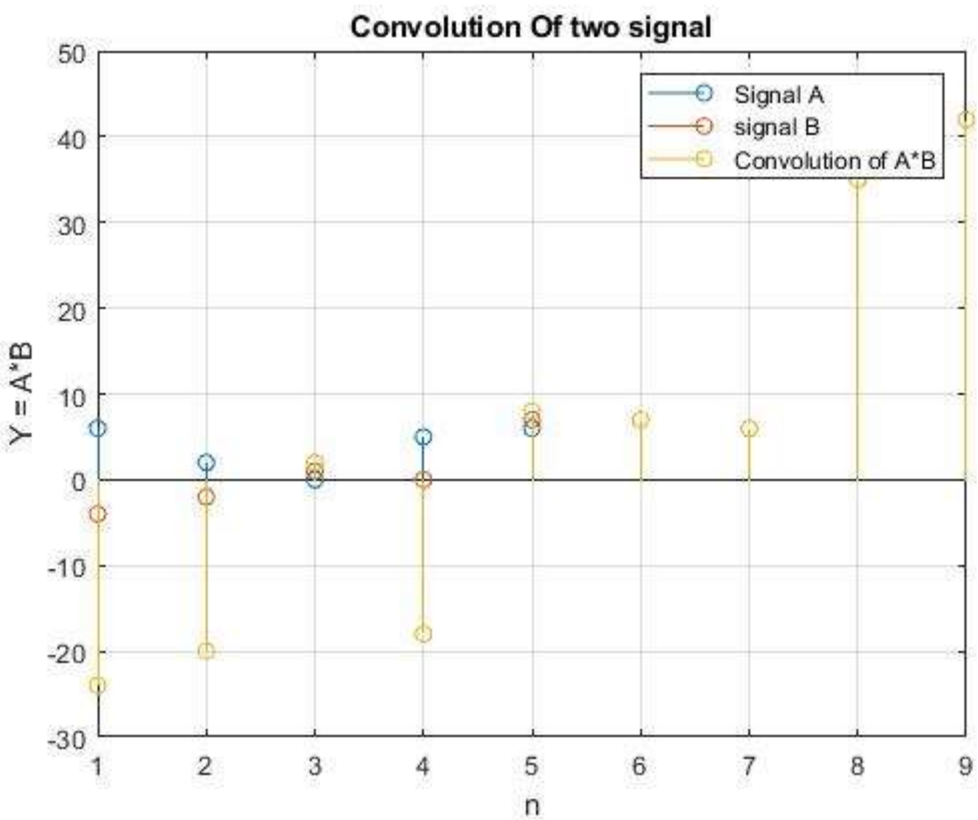
Observation: This is the discrete time sequence of x(n). The sampling rate of *y*(*2n*) is half the rate of *y*(*n*) therefore y(2*n*) is called the under sampled version of *y*(*n*). Where n are the samples and y(n) are the corresponding discrete values. In where x-axis represents n samples and y-axis represents y(n) signal.

Task7: Determine convolution of two sequences *A* = {6, 2, 0, 5, 6} and *B* = {-4, -2, 1, 0, 7} and find the length of resultant sequence.
Solution:

```
% Determine convolution of two sequences
% A = {6, 2, 0, 5, 6} and B = {-4, -2, 1, 0, 7}
% Find the length of resultant sequence.
A = [6 2 0 5 6];
B = [-4 -2 1 0 7];
Y = conv(A,B);
stem(A);
hold on;
```

```
stem(B);
hold on;
stem(Y);
title('Convolution Of two signal');
xlabel('n');
ylabel('Y = A*B');
legend('Signal A', 'signal B', 'Convolution of A*B');
grid on;
```

Output:

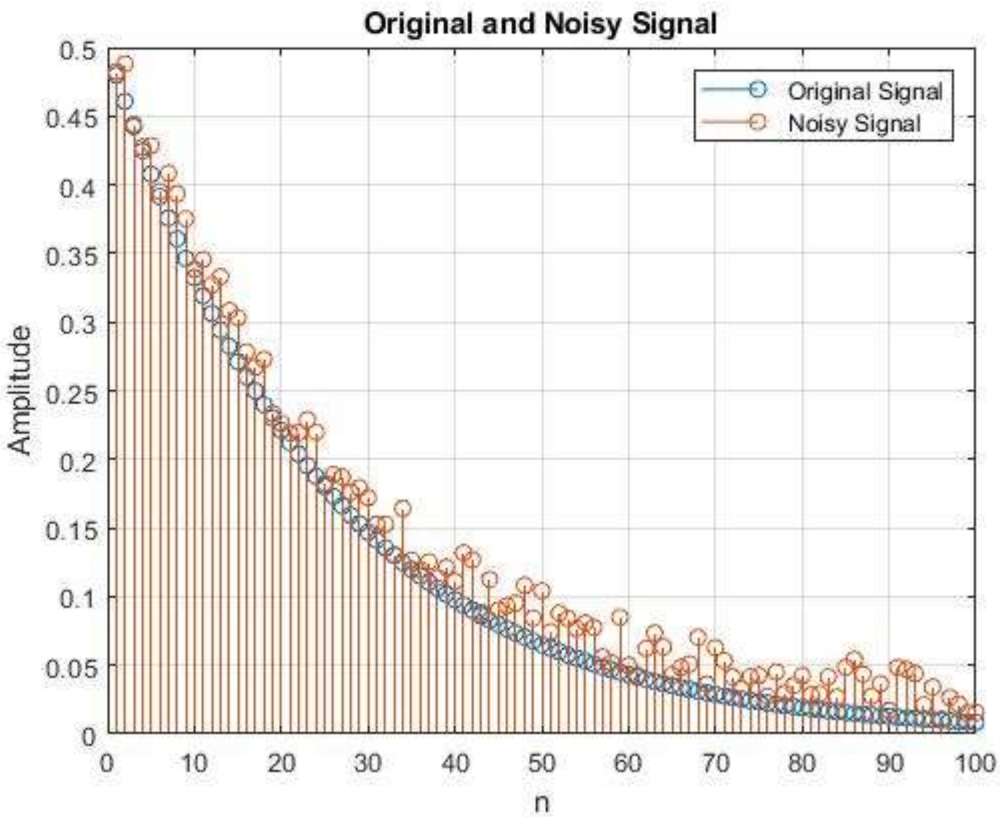


Observation: This is the discrete time sequence of x(n). Here I found convolution of two sequences *A* = {6, 2, 0, 5, 6} and *B* = {-4, -2, 1, 0, 7} and the length of resultant sequence. Where n are the samples and y(n) are the corresponding discrete values. In where x-axis represents n samples and y-axis represents y(n) signal.

Task-8: Generate 100 random numbers between 0 and 1. Add them with an exponential sequence as a random noise. Plot both noisy and noiseless signals.
Solution:

```
% Generate 100 random numbers between 0 and 1.
% Add them with an exponential sequence as a random
noise.
% Plot both noisy and noiseless signals.

a = 0.5;  % assume
r = 0.96; % assume
N = 100; % as we want to generate 100 random numbers

for n = 1:N
    e(n) = n;
    y(n) = a*r.^n;
    Y(n) = y(n) + 0.04*rand(); % noise 0.04 adding with
random number
end
stem(e,y)
hold on
stem(e,Y)
hold on
title('Original and Noisy Signal')
xlabel('n')
ylabel('Amplitude')
legend('Original Signal','Noisy Signal')
grid on
```

Output:



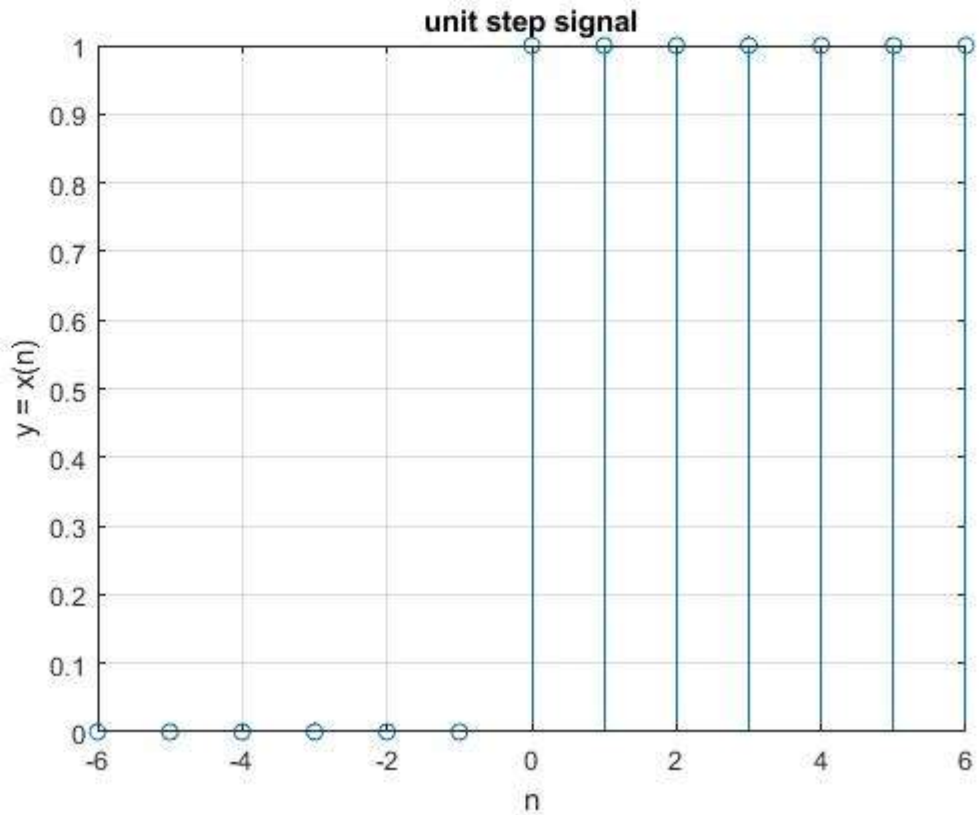
Observation: This is the discrete time sequence of x(n). Here I found 100 random numbers between 0 and 1. Add them with an exponential sequence as a random noise. Then plot both noisy and noiseless signals. Where n are the samples and y(n) are the corresponding discrete values. In where x-axis represents n samples and y-axis represents y(n) signal.

Task-9.1: Generate a unit step sequence for *n* = [-6, 6]

Solution:

```
%Generate a unit step sequence for n = [-6, 6]
n = [-6:6];
y = [0 0 0 0 0 0 1 1 1 1 1 1 1];
stem(n, y)
title('unit step signal')
xlabel('n')
ylabel('y = x(n)')
grid on
```

Output:



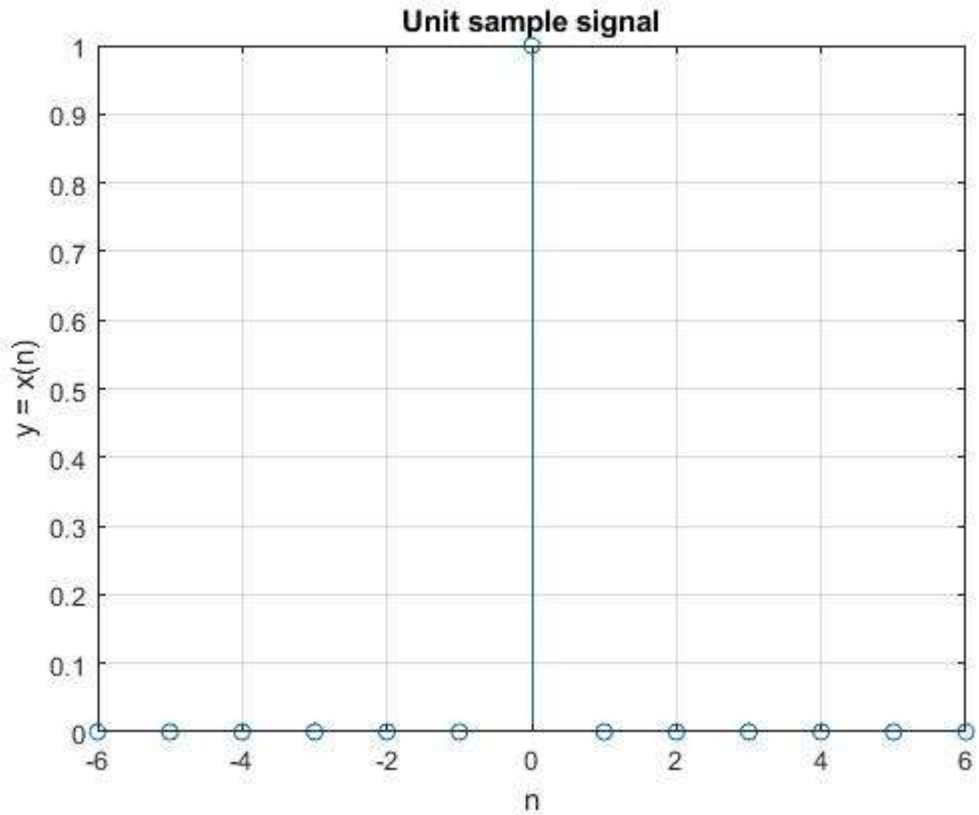
Observation: This is the unit step sequence for *n* = [-6, 6]. Where n are the samples and x(n) are the corresponding discrete values. In where x-axis represents n samples and y-axis represents x(n) signal.

Task-9.2: Generate a unit sample sequence for n = [-6, 6].

Solution:

```
n = [-6:6];
y = [0 0 0 0 0 0 1 0 0 0 0 0 0];
stem(n, y)
title('Unit sample signal')
xlabel('n')
ylabel('y = x(n)')
grid on
```

Output:



Observation: This is the unit sample sequence for *n* = [-6, 6]. Where n are the samples and x(n) are the corresponding discrete values. In where x-axis represents n samples and y-axis represents x(n) signal.

Task-10: Computes the two-dimensional convolution of matrices

$$A = \begin{vmatrix} 1 & -2 & 0 \\ 3 & 4 & -11 \\ 6 & -5 & 2 \end{vmatrix}$$

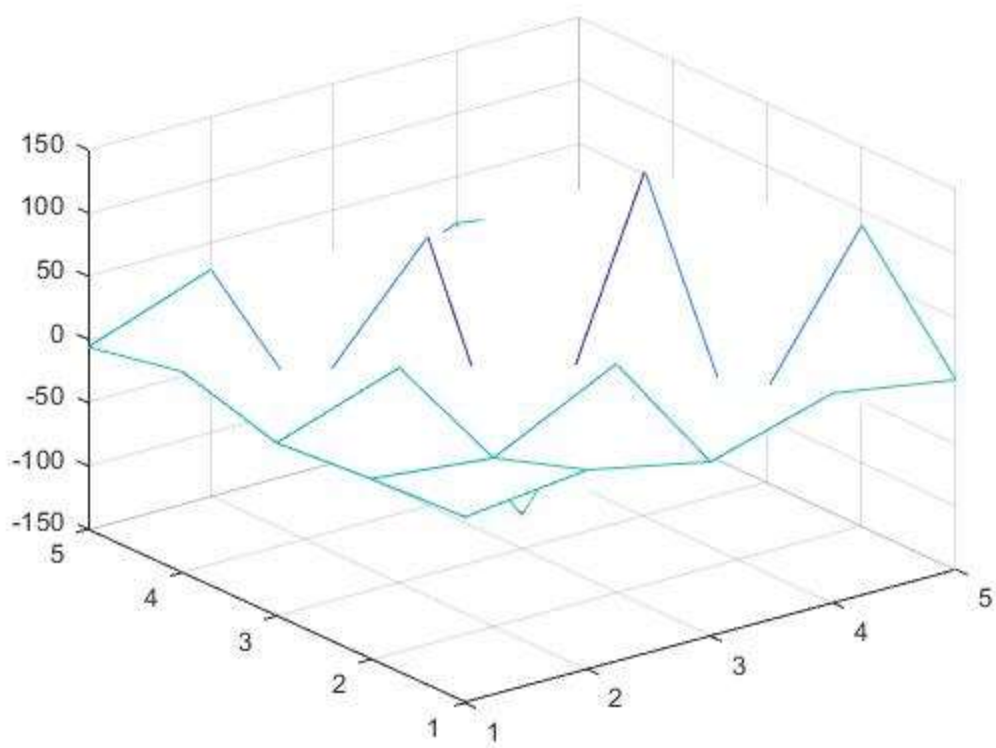
$$B = \begin{vmatrix} -3 & 2 & -8 \\ 2 & -7 & 0 \\ -1 & 4 & -1 \end{vmatrix}$$

Solution:

```
A = [ 1 -2 0; 3 4 -11; 6 -5 2]

B = [-3 2 -8; 2 -7 0; -1 4 -1]
```

```
Y = conv2(A,B) % for two dimentional we used conv2 ,
for 1 dimention used conv
mesh(Y)

% It is used to generate 3D surface plot of which x, y
co-ordinates
% are decided by column and row indices of the input
matrix 'Z'.
```

Output:



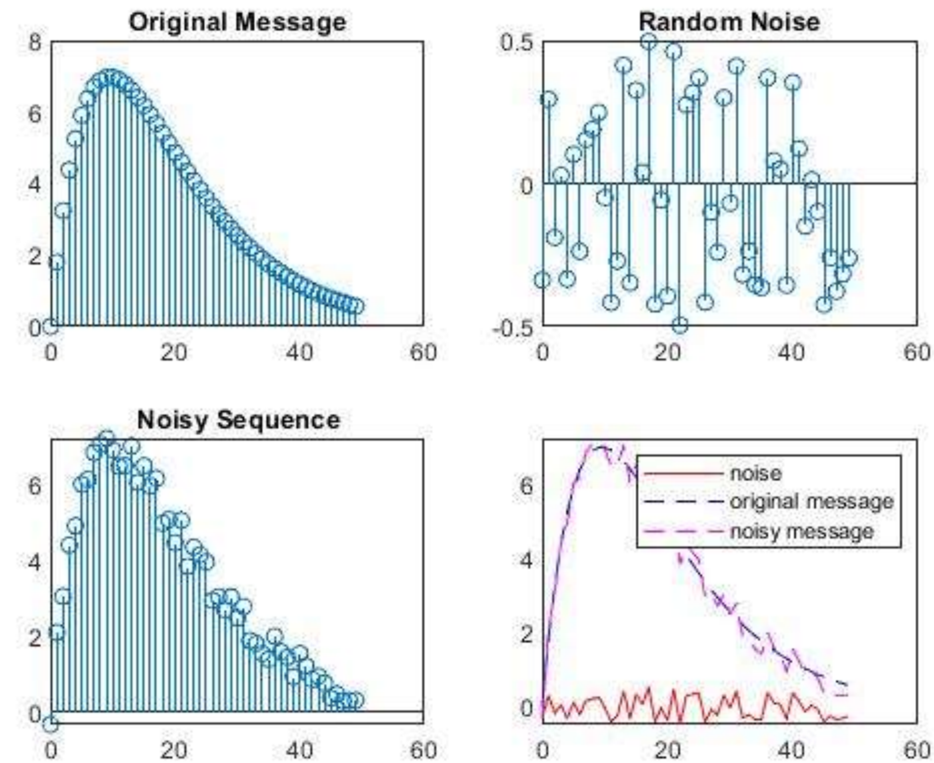
Observation: This is the discrete sequence of x(n). Here computing the two-dimensional convolution of matrices using mesh. Where n are the samples and x(n) are the corresponding discrete values. In where x-axis represents n samples and y-axis represents x(n) signal.

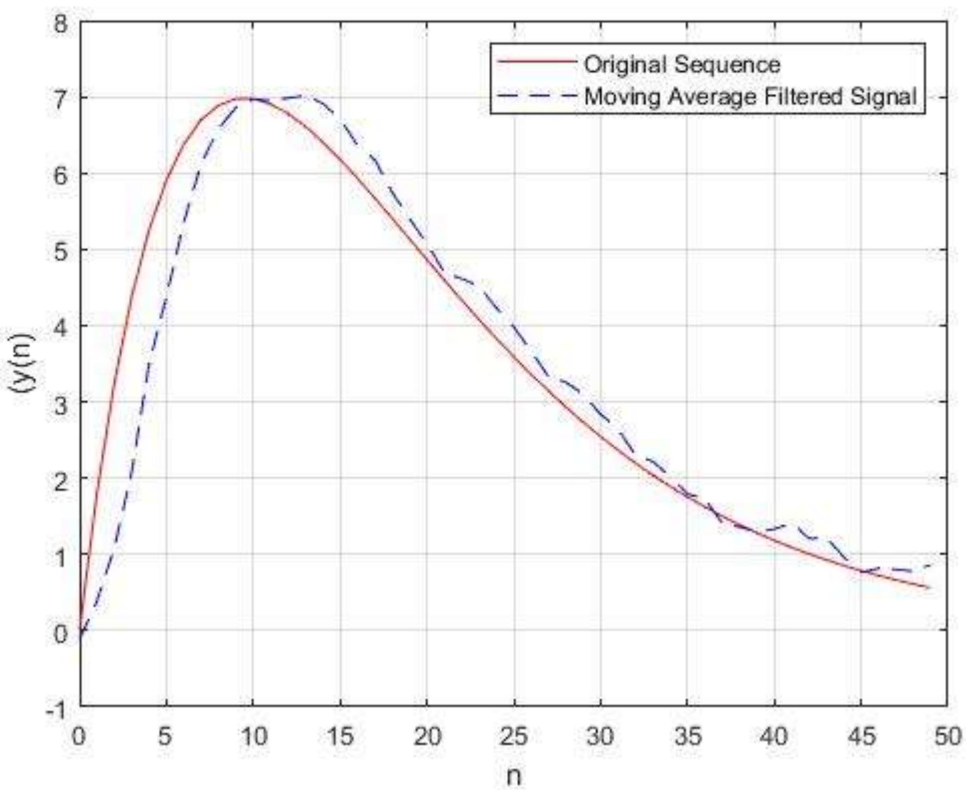
Task-11: Generate 50 random numbers in the range [1,-1] and add it with the sequence *x(n)* =2*n*(0.9)*n*. Pass the noisy signal through a moving average filter and observe the signal before and after filtering.

Solution:

```
N=50;%number of samples
m=0:1:N-1;
d=rand(N,1)-0.5;%noise with mean 0 and lies between -
0.5 to 0.5
s=2*m.*(0.9.^m);%original Sequence
x=s+d'; %noisy sequence
subplot(2,2,1)
stem(m,s)
title('Original Message')
subplot(2,2,2)
stem(m,d)
title('Random Noise')
subplot(2,2,3)
stem(m, x)
title('Noisy Sequence')
subplot(2,2,4)
plot(m, d, 'r-', m, s, 'b--', m, x, 'm--')
legend('noise','original message', 'noisy message')
M=input('Value of M=');% Value of M from key board
b=ones(M, 1)/M;
y=filter(b,1,x);
plot(m,s,'r-',m,y,'b--')
legend('Original Sequence', 'Moving Average Filtered
Signal')
grid on
xlabel('n')
ylabel('(y(n)')
```

Output:


Original Message
Random Noise
Noisy Sequence
noise
original message
noisy message


Original Sequence
Moving Average Filtered Signal
(y(n))
n

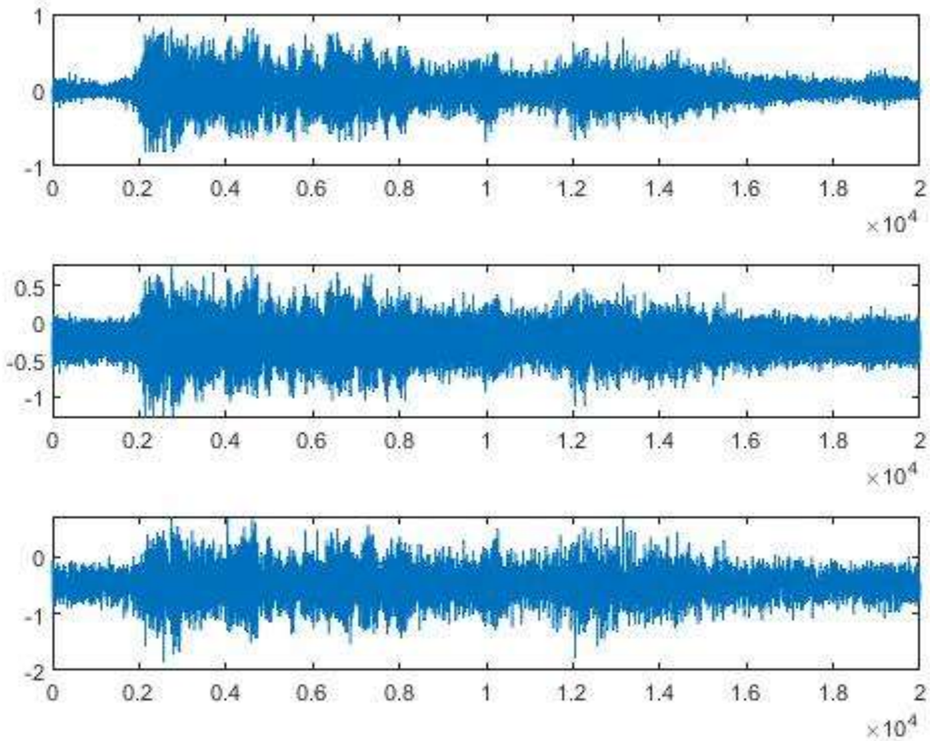
Observation: Here 50 random numbers in the range [1,-1] and adding it with the sequence *x(n)* =2*n*(0.9)*n*. Passing the noisy signal through a moving average filter and observe the signal before and after filtering. Where n are the samples and x(n) are the corresponding discrete values. In where x-axis represents n samples and y-axis represents x(n) signal.

Task-12: listen to the speech signal with noise removal.

Solution:

```
load handel %original music signal
u=y(1:20000);
sound(u);

d=0.5*rand(length(u),1)-0.5;%noise with mean 0 and lies
between -0.25 to 0.25
x=u+d; %noisy sequence
sound(x)

M=5;% Value of M
b=ones(M,1)/M;
z=2*filter(b,1,x);

sound(z)
subplot(3,1,1)
plot(u)

subplot(3,1,2)
plot(x)
subplot(3,1,3)

plot(z)
```

Output:



Observation: Here listening to the speech signal with noise removal. Where n are the samples and x(n) are the corresponding discrete values. In where x-axis represents n samples and y-axis represents x(n) signal.

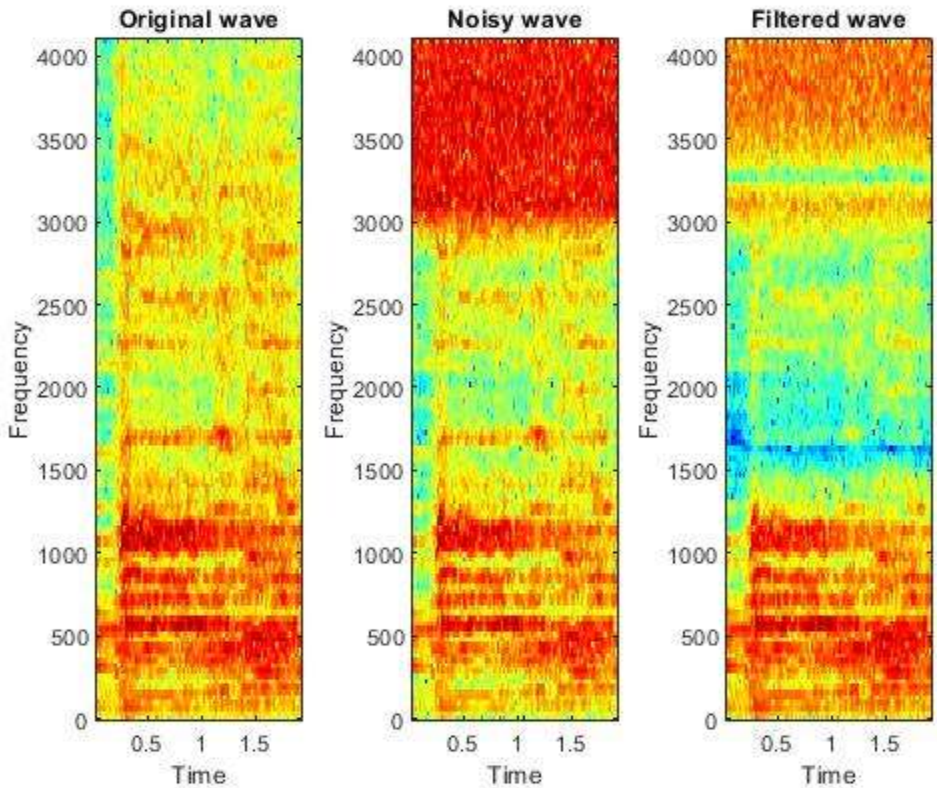
Task-13: Elimination of noise by moving average method. Let us first load a voice or music signal and add some noise with it. Finally we will filter the signal and observe the signals and corresponding spectrograms.

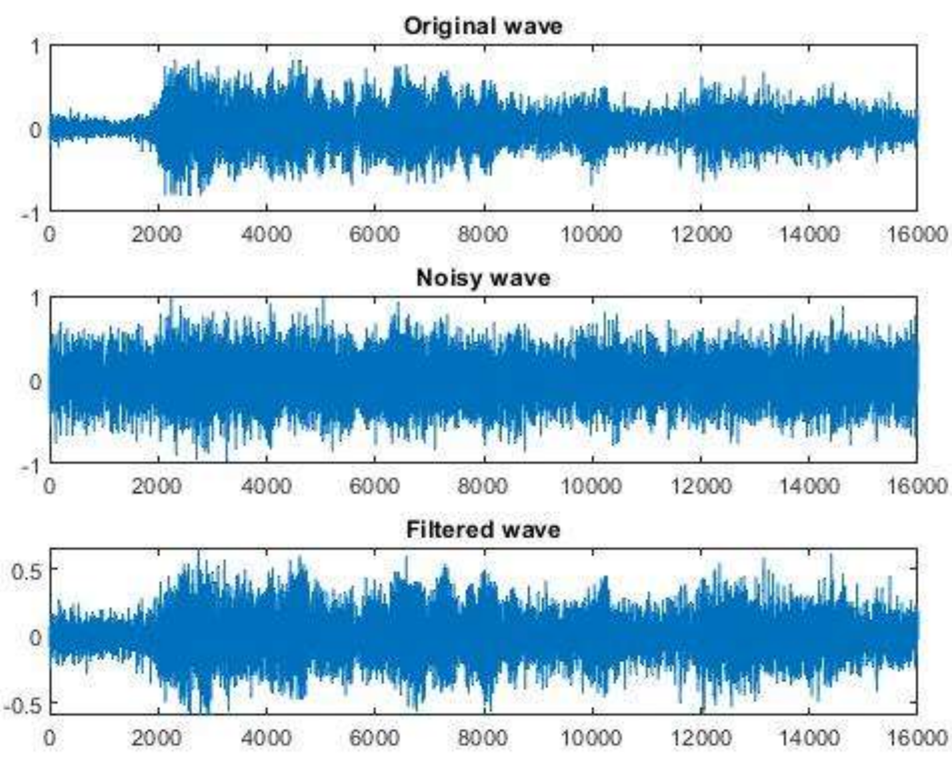
Solution:

```
load handel %original signal
u=y(1:16000);

[num,den]=ellip(4,3,40,0.75,'high');
noise=filter(num,den,randn(length(u),1));

x=u+noise;
```

```
x=x/max(max(x));
M=5;% 5 sample will be averaged

b=ones(M,1)/M;
z=2*filter(b,1,x);

figure(1)
subplot(1,3,1)

specgram(u,[],Fs)
title('Original wave')

subplot(1,3,2)
specgram(x,[],Fs)

title('Noisy wave')
subplot(1,3,3)

specgram(z,[],Fs)
title('Filtered wave')

figure(2)
subplot(3,1,1)

plot(u)
title('Original wave')

subplot(3,1,2)
plot(x)

title('Noisy wave')
subplot(3,1,3)

plot(z)
title('Filtered wave')
```

Output:


Original wave
Noisy wave
Filtered wave
Frequency
0
500
1000
1500
2000
2500
3000
3500
4000
Time
0.5
1
1.5


Original wave
Noisy wave
Filtered wave
1
0
-1
0.5
-0.5
0
2000
4000
6000
8000
10000
12000
14000
16000

Observation: Here I am using elimination of noise by moving average method. At first load a voice or music signal and add some noise with it. Finally I will filter the signal and observe the signals and corresponding spectrograms. Where n are the samples and x(n) are the corresponding discrete values. In where x-axis represents n samples and y-axis represents x(n) signal.