Student Name: Emirhan Uçar (25265), Emre Eren (25139)
Date of Submission:28.02.20

EE312 LAB REPORT#1
SPRING 2020

Information

*We used this piece of code to visualize and listen to sounds and aimed to see what happens.

```
size('bird.wav');
length(y)
whos y;
whos fs;
TotalTime = length(y)./fs;
t = 0:TotalTime/(length(y)):TotalTime-TotalTime/length(y);

figure(1);
plot(t,y);
xlabel('Time[n]'); ylabel('Amplitude');



size('bird.wav');
length(x)
whos x;
whos fs;
TotalTime2 = length(x)./fs;
t2 = 0:TotalTime2/(length(x)):TotalTime2-TotalTime2/length(x);
```

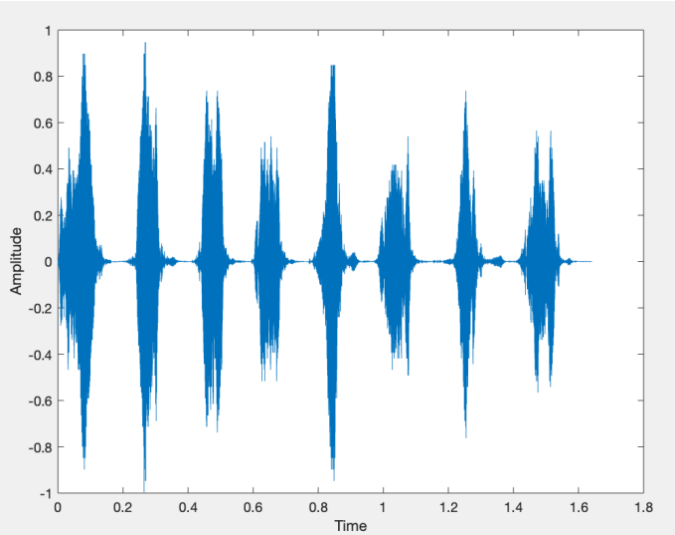
```
figure(2);
plot(t2,x);
xlabel('Time[n]'); ylabel('Amplitude');

sound(x,fs);
pause();
sound(y,fs);
```

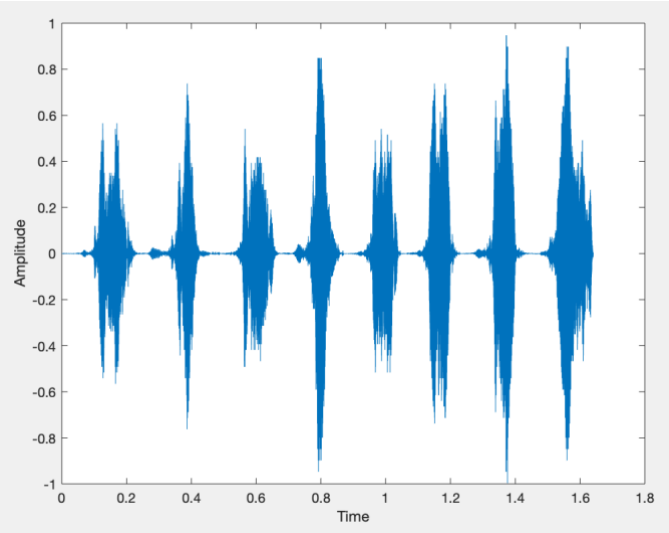
Problem 1

a)

*This piece of code forms a new sound signal which is reverse version of original signal 'bird.wav'.

```
close all;

[x,fs]=audioread('bird.wav');

y=x(end:-1:1);
sound(x,fs);
```





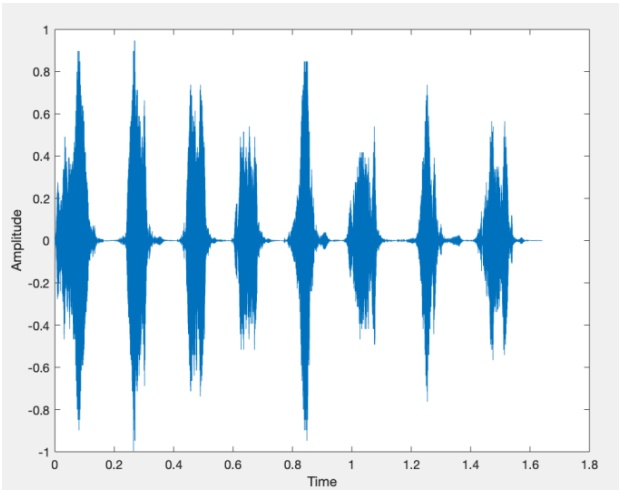

```
sound(y,fs);
```

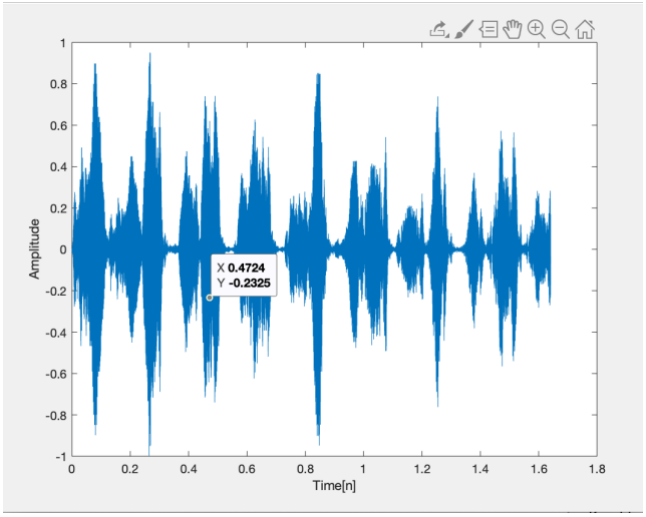
b)

*These two process actually do same thing. One of them labeled as (**) creates a vector which copies elements in x and put zeros at the end of the vector, then places element in x into zeros. As a consequence, new vector is consisted of sum of x and half of delayed version of x. In other process (***), this addition is done by for loop it makes it for each index one by one.

```
close all;
clear all;
clc;
[x,fs]=audioread('bird.wav');
% Read the wav file
N=length(x);
prm=1000; % Parameter for the process
x=x(:); % make x row vector
y = [x zeros(1,prm)] + 0.5*[zeros(1,prm) x]; (**)
% or do the long way (a little different then above,
% what is different?)
```

```
y2=x;
for i=prm+1:N (***)
y2(i)=y2(i)+0.5*x(i-prm);
end
```

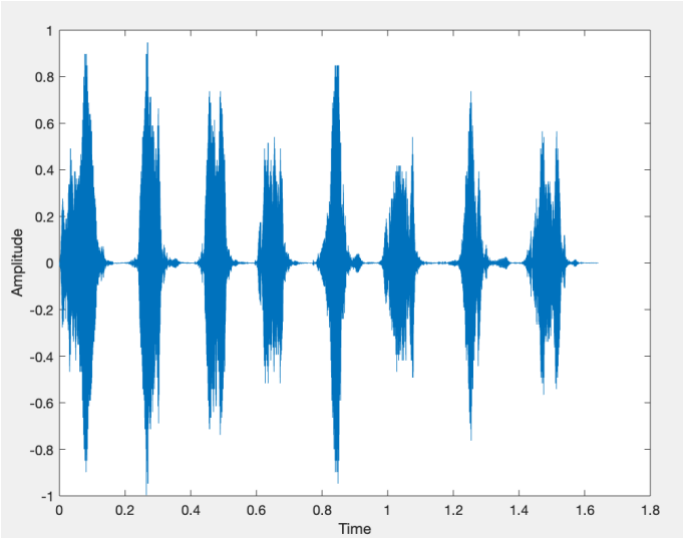


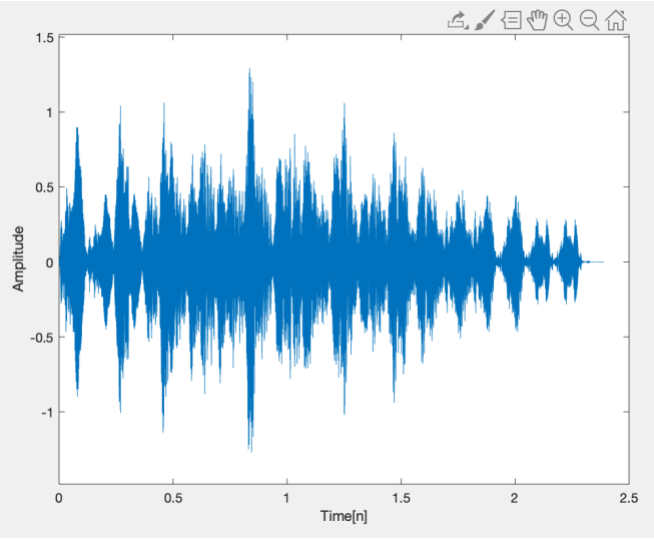


c)

* In this process, firstly y2 is equalized to x, then, halved delayed version of x added to y2. Also, delay changes for every iteration of for loop in scale from 1000 to 6000.

```
close all;
clear all;
clc;
[x,fs]=audioread('bird.wav');
N = length(x);
x = x';
y2=x;
for i=1000:1000:6000
    y2=[y2 zeros(1,1000)]+0.5*[zeros(1,i) x];
end
```





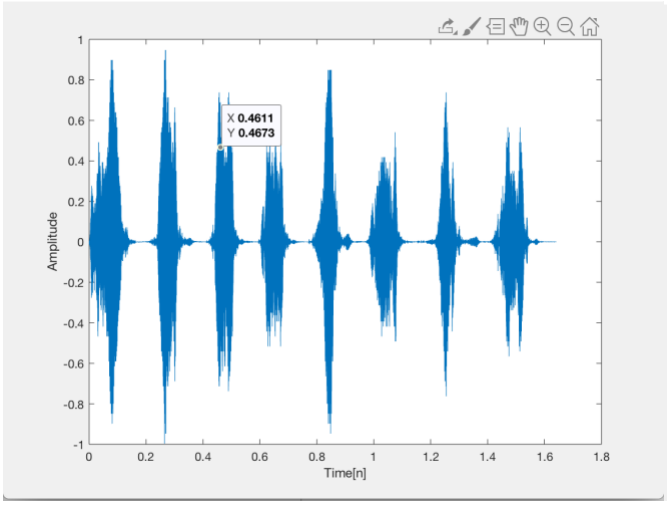
d)

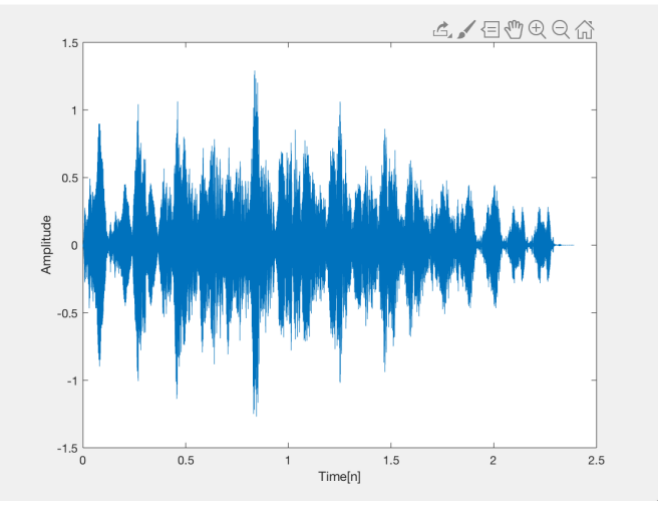
*

My ear cannot detect any differnece between convolution and filtering but there should be differences because their length aren't same. Basically, according to definition of convolution increasing length of sample is expected. (kanka burayi da bi kontrol edersen süper olur)!!!!!!!!!!1!!!

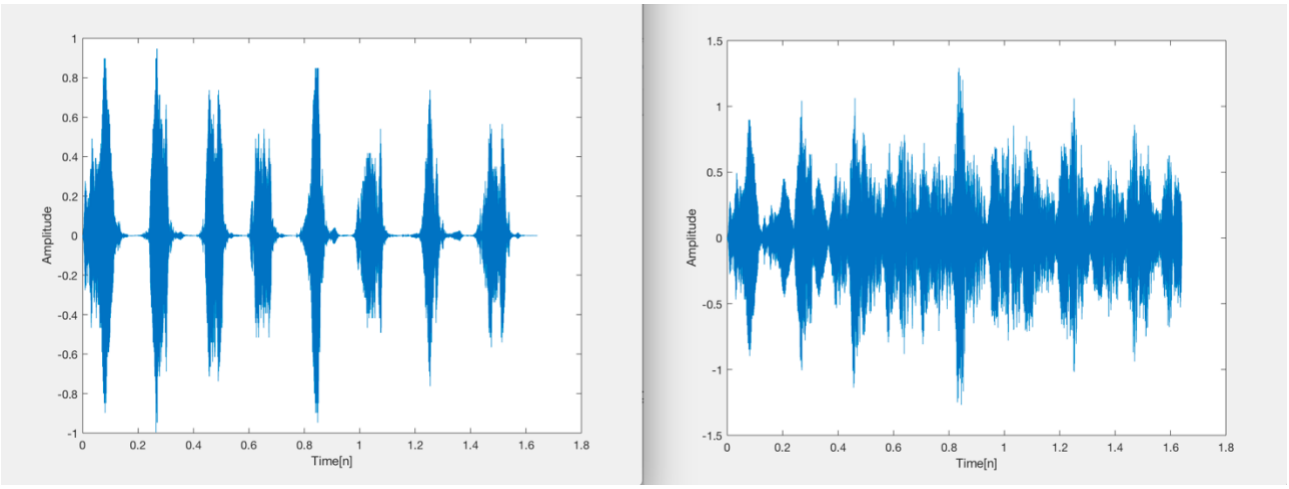
in this example, when x filtered its length does not change, however when x convolved with h length changes.

Convolved version:





Filtered version:



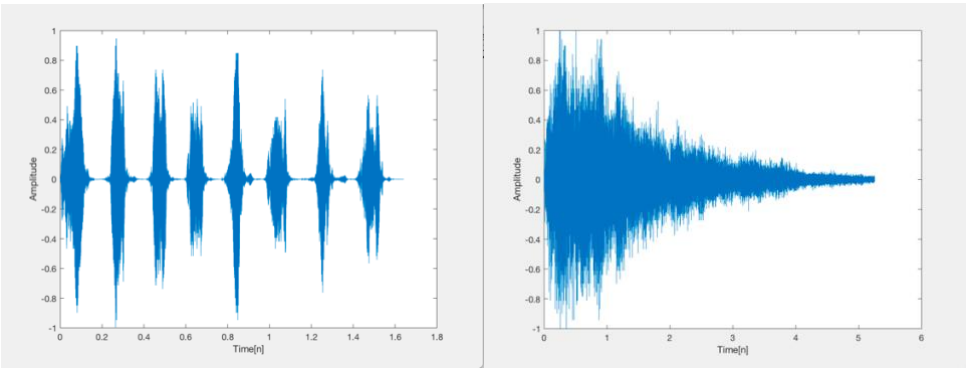
e) this equation increase length of the signal and decrease the sample rate (kanka buna bi bak istersen !!!!!!!!!!!!!!!!)

```
h =1;

for i=1000:1000:6000
    h= [h; zeros(1000,1)]+0.5*[zeros(i,1); 1];
end
```

f) bunu aciklayamadim kanka

```
for i=1:1:N
    if mod(i,2) == 0
        y2 = x(i*0.5);
    else
        y2 = zeros(1,N);
    end
end
```

Problem 2

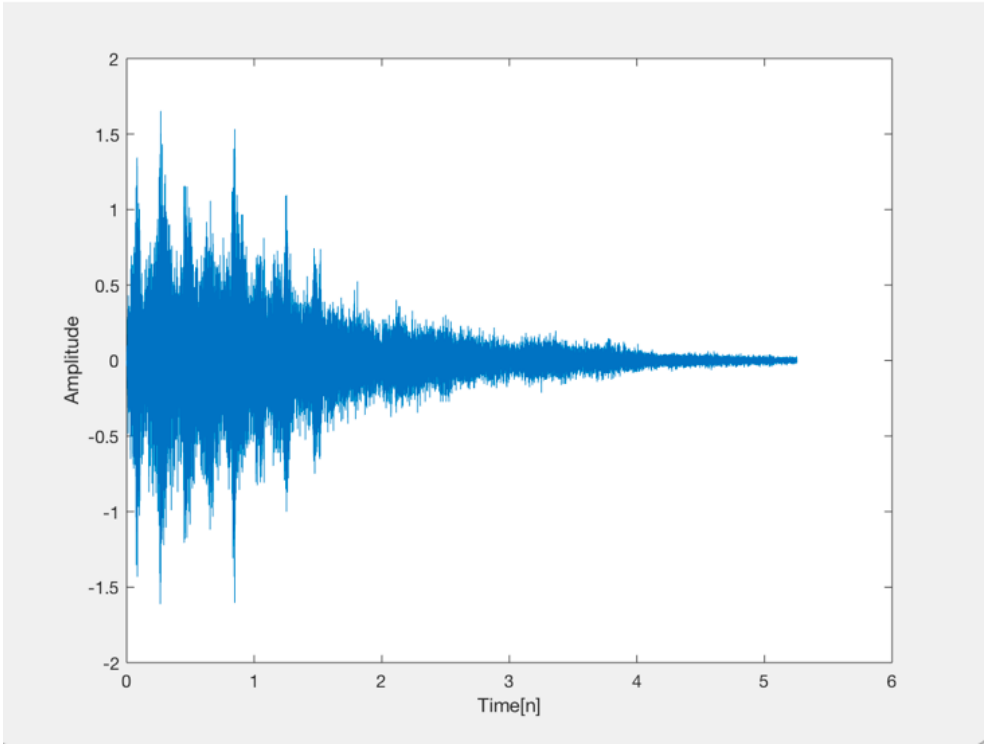


a) this proccess mixes „bird.waw“ and „gong.waw“ then play it togather. (first plot is x, second is z) (x is „bird.waw“ and z is „gong.waw“) code;

```
for i=1:1:N2
   z(i) = x1(i) + y(i);
```

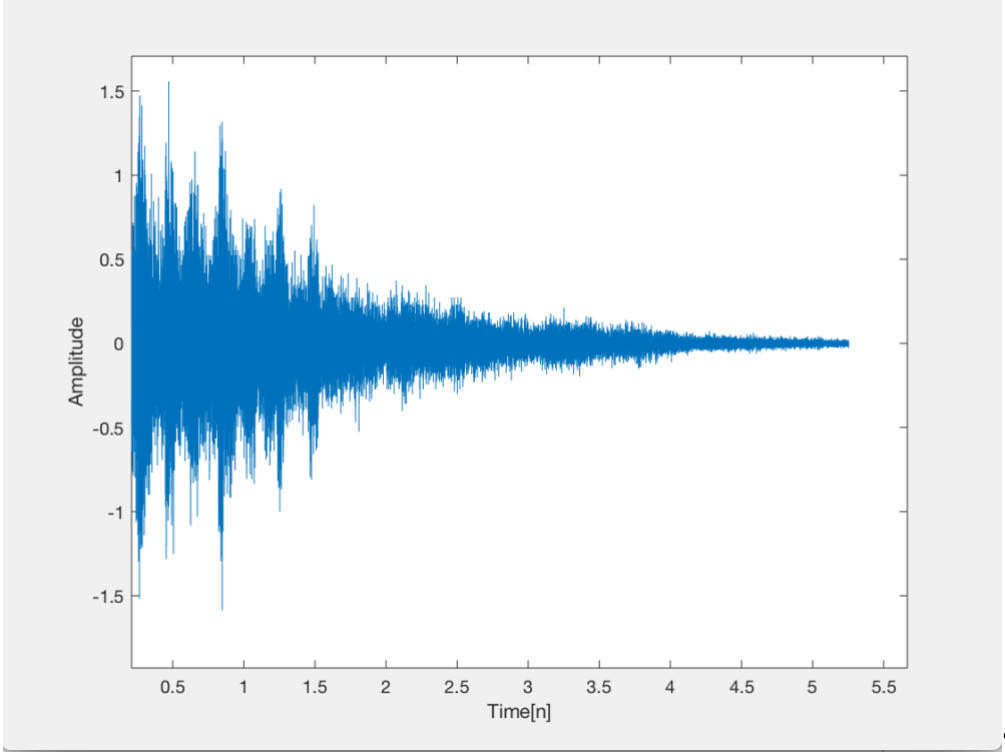
```
en
```

Plot-> y = x + z



b) plot-> y = x-z

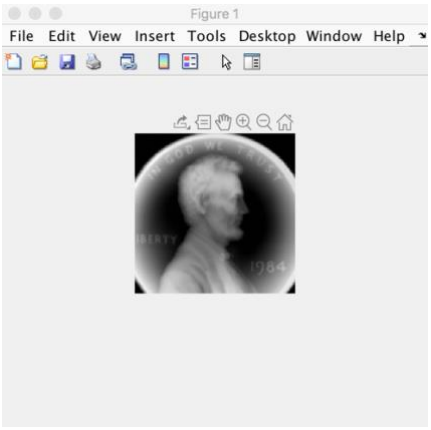
```
for i=1:1:N2
   z(i) = x1(i) - y(i);
end
```



This example is same as part a). because in this example we reversed gong wave then we add it to bird wave. Therefore, there is

no differences compared to bird wave in terms of what we hear. However, there is a difference in terms of plot. Because we reversed gong wave (-z).

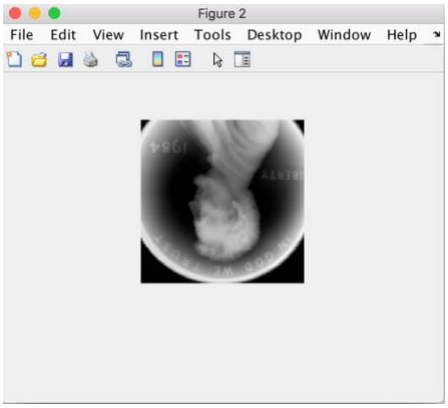
Problem 3



(original penny picture)

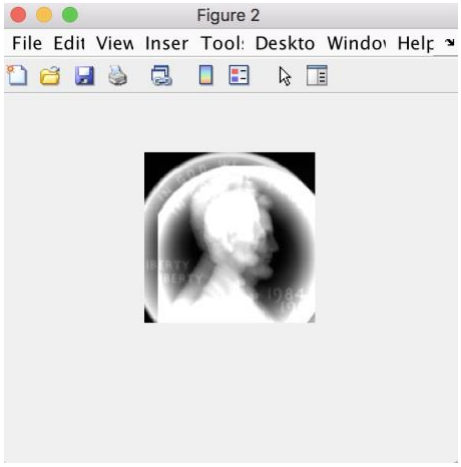
a) in this example penny is turned down with this piece of code;

```
y=P(end:-1:1,end:-1:1);
```



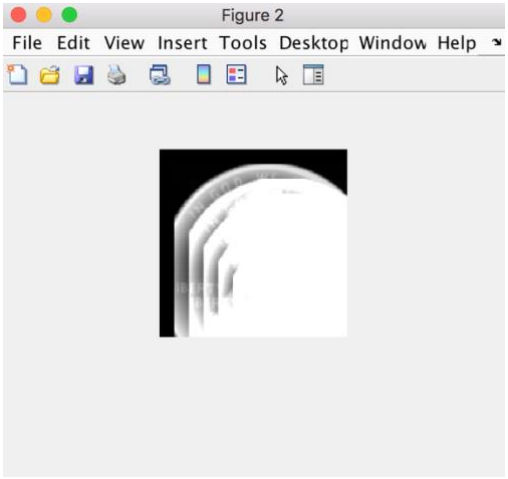
b) in this example original picture is slided by 10 pixels right and down then added with original signal. code;

```
for i=1:1:h
    for j=1:1:w
        if (i < 11)|| (j < 11)
            y(i,j)=P(i,j);
        else
            y(i,j)=P(i,j)+P(i-10,j-10);
        end
    end
```



c) ? anlamadim kanka bunun ne yaptigini. code;

```
for k=10:10:100
    for i=k+1:h
        for j=k+1:w
            y(i,j)=y(i,j)+P(i-k,j-k);

        end
    end
end
```

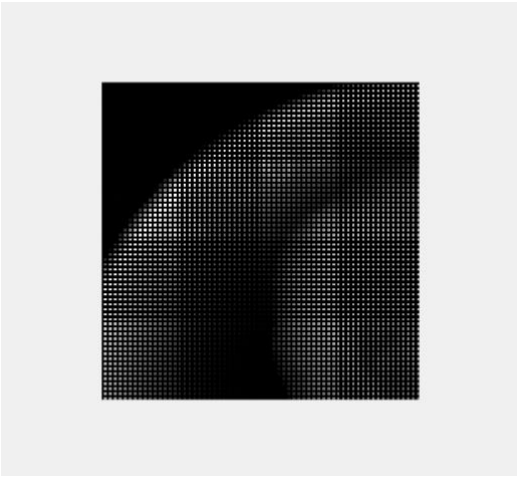


d) In this example equation ???????????????? with following pieace of code; Kanka koddan emin degilim hatali oldugunu dusunuyorum sen de bi bak istresen

```
for i=1:1:h
    for j=1:1:w
        if mod(i,2) == 0 && mod(j,2) == 0
          y(i/2,j/2)= P(i,j);
        end
    end
end
```

e) in this example following piece of code makes zoom in to the original peny. code;

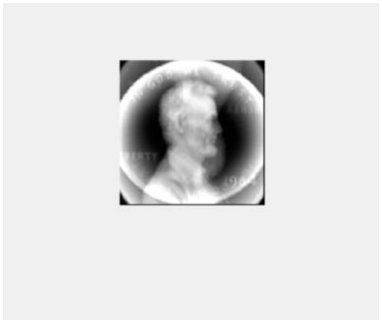
```
for i=1:1:h
    for j=1:1:w
        if mod(i,2) == 0 && mod(j,2) == 0
           y(i,j)=P(i/2,j/2);
        else
           y(i,j)=0;
        end
    end
end
```



f) in this code, penny is reversed respect to x axis and its brightness is halved then it is added to original penny. code;

```
[h,w]=size(P);
y =P(end:-1:1,end:-1:1);
z= zeros(h,w);

for i=1:h-1
```

```
        for j=1:w-1
            z(i,j) = P(i,j) + 0.5*y(i,j);
         end
end
```

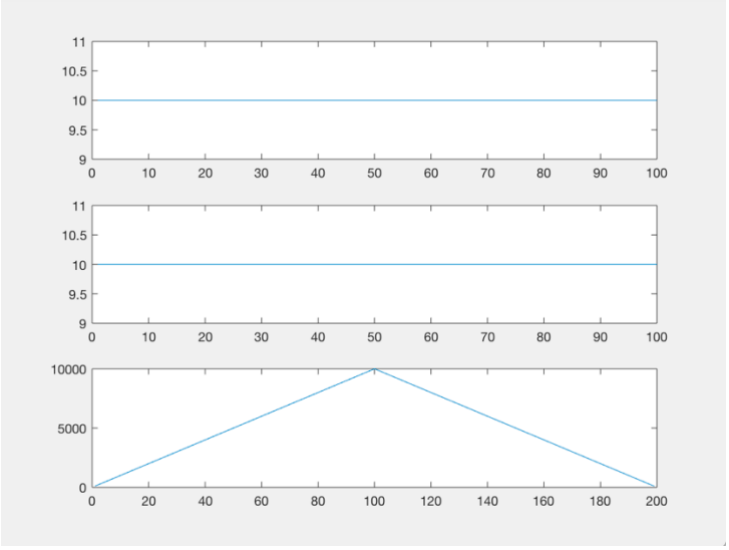


Problem 4:

Code;

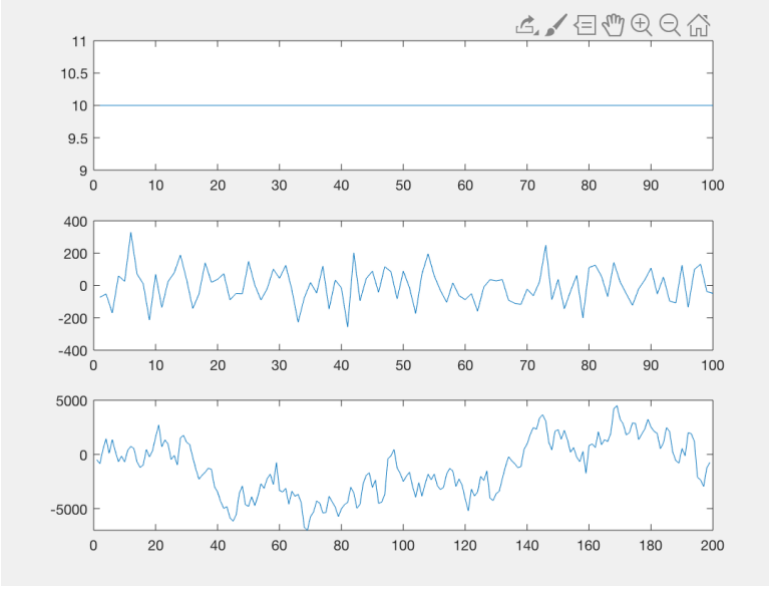
```
d = ...; % define delay amount in samples
sigma = ...; % define noise standard deviation
x = 10*ones(1,100); % x is a rectangular pulse of length 100 samples
y = [zeros(1,d) x]; % form noiseless y
noisey = sigma * randn(size(y)); % form the noise signal same size as y
y = y + noisey; % form noisy y
rxy = xcorr(x,y); % cross correlation between x and y

figure;
subplot(311);
plot(x);
subplot(312);
plot(y);
subplot(313);
plot(rxy);

[a, b] = xcorr(x,y);
dhat = -b(a==max(a))
```

1) delay(d) = 0 and sigma = 0: dhat = 0



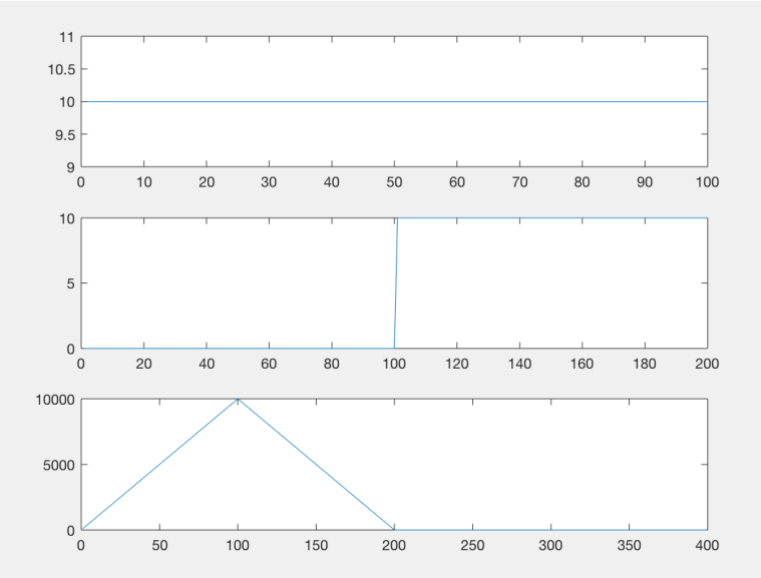
In this example if we set both d and sigma is 0, we obtain both signal x and y are same, and they absolutely correlated each other.

2) delay = 0 and sigma = 100: dhat = -1



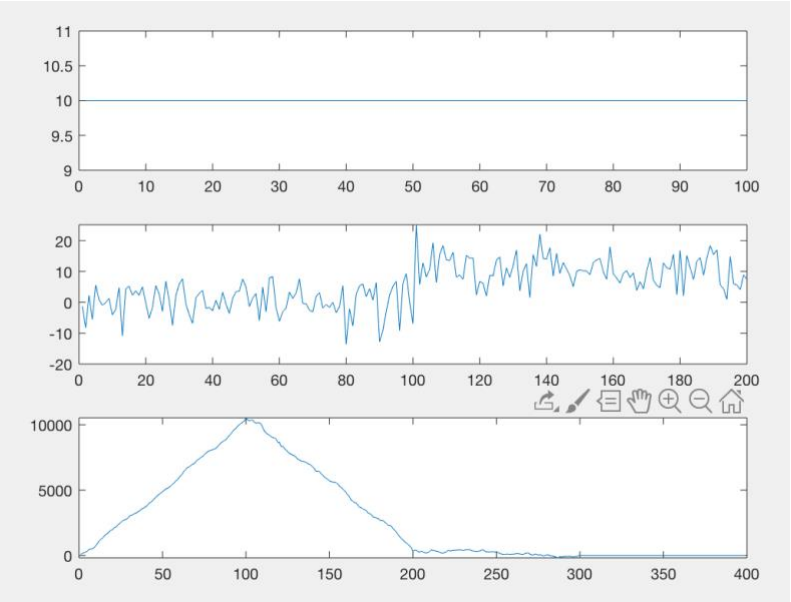
if we do not touch delay and increase sigma y was corrupted compare to the x, according to this observation and can be seen in cross correlation graph which is last one, there is no correlation between x and y. Moreover, when we increase sigma dhat becomes inconsistent. Because noise of signal is increased when sigma is large enough.

3) delay = 100 and sigma = 0: dhat = 100



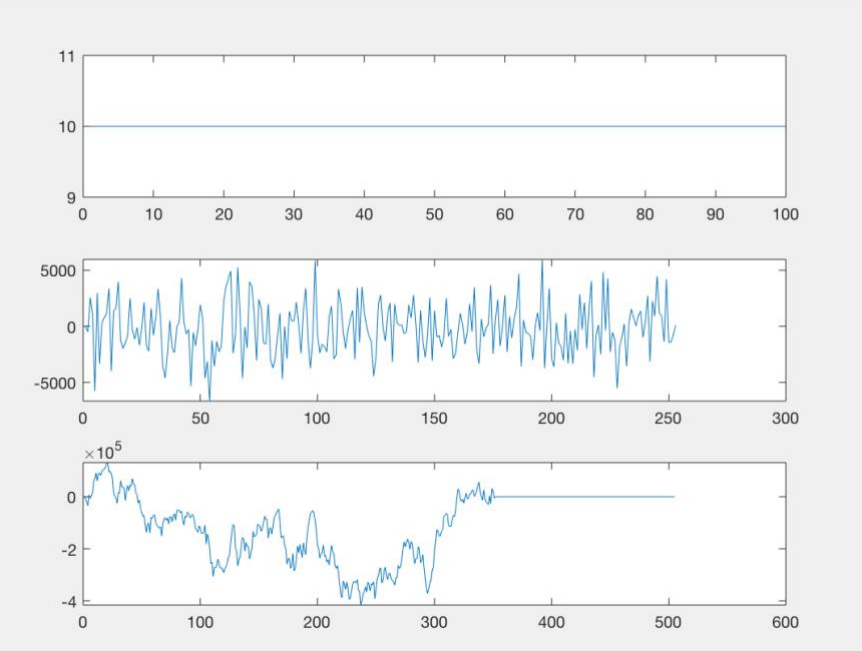
As can be seen in this plot and compare it by others, if sigma is 0, cross correlation can be easily seen between x and y. because if we increase amount of sigma, corruption on y will increase and relation between x and y is broken. Moreover, when sigma is 0, dhat is absolutely consistent. In this case d = 100 but it does not affect dhat because delay does not corrupt (not give noise) signal.

4) delay = 100 and sigma = 5: dhat = 100



If we little increase sigma, corruption on y can be easily seen however, since amount of sigma is small, we can obtain correlation between x and y. Also, since sigma is small dhat does not inconsistent.

5) delay = 153 sigma = 2312: dhat = 22



If we increase sigma lot more y becomes more and more corrupted. There is no correlation between x and y, as can be seen in the last plot. The main observation in this plot, we can see if we increase sigma a lot dhat becomes more and more inconsistent. In example 4 delay amount is 100 which is near 153, sigma amount is 5 and dhat is 100. This means when sigma is increased, noise of signal would be increased, then dhat becomes more and more inconsistent. Because corrupted signal cannot be obtained clearly.