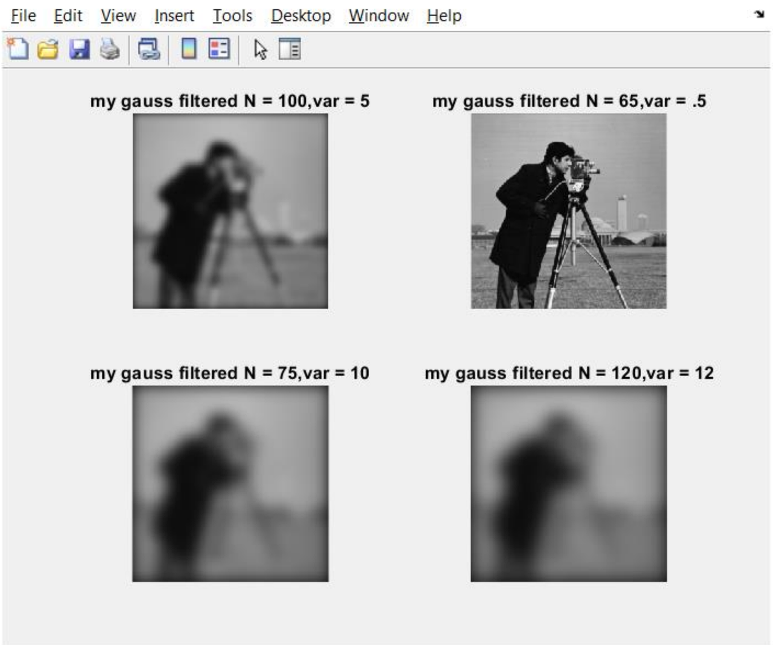
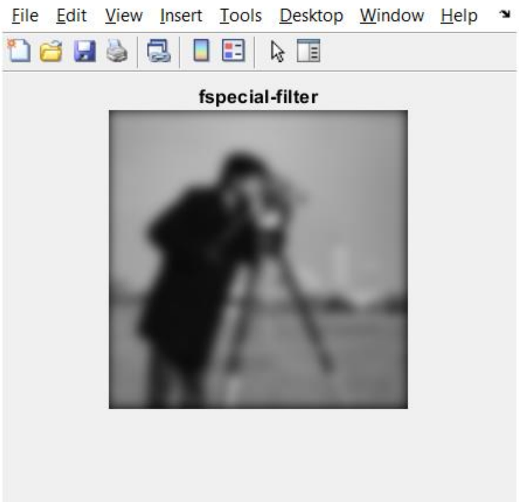
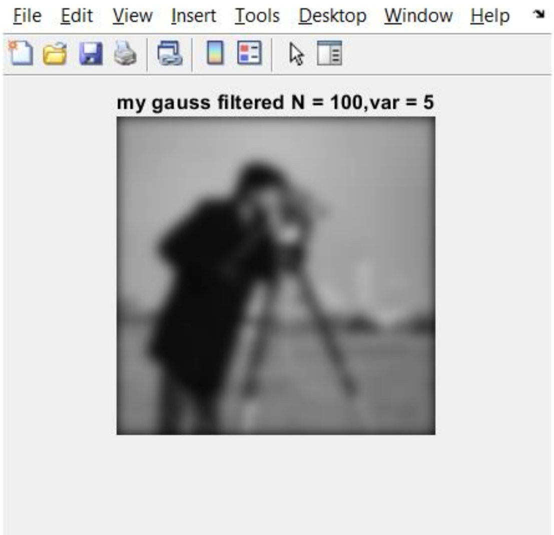
ASSIGNMENT 2 REPORT

Q1,1)



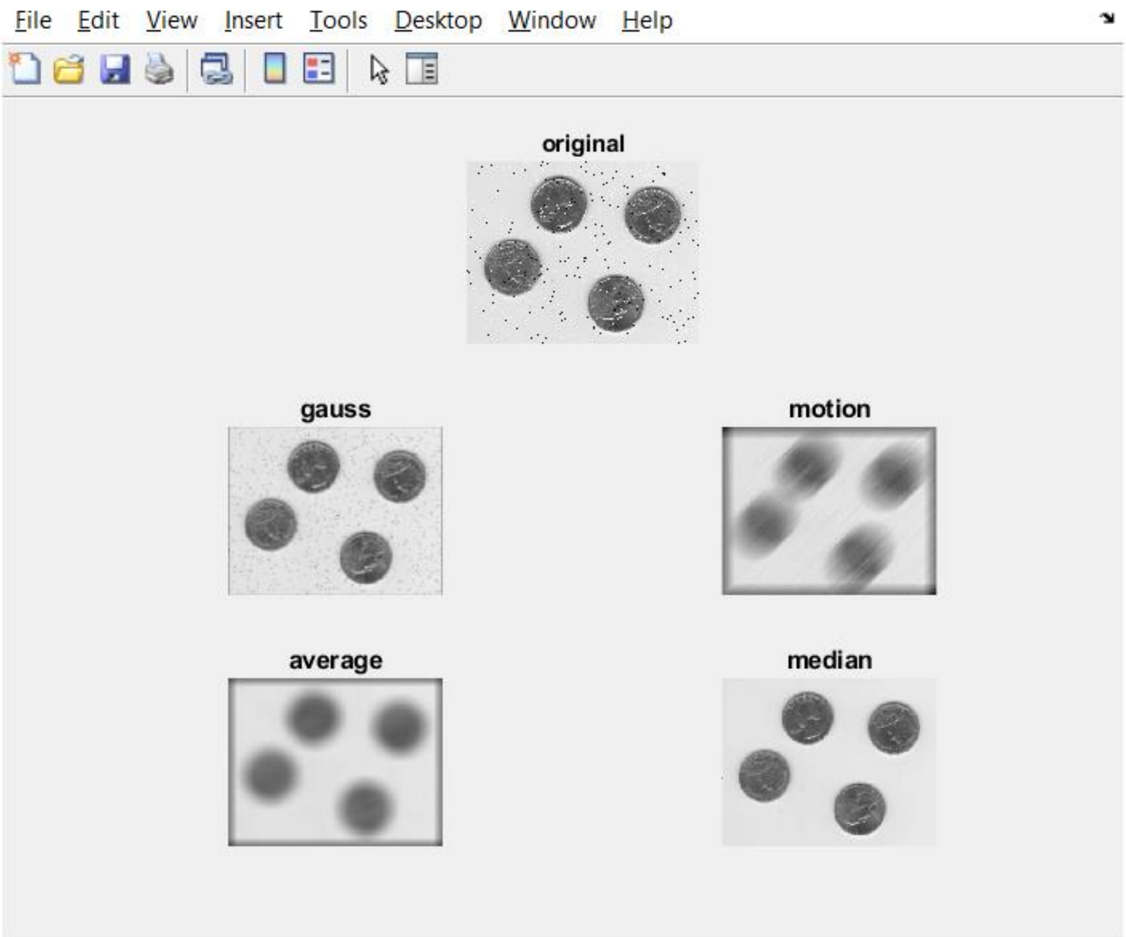
Q1,3)





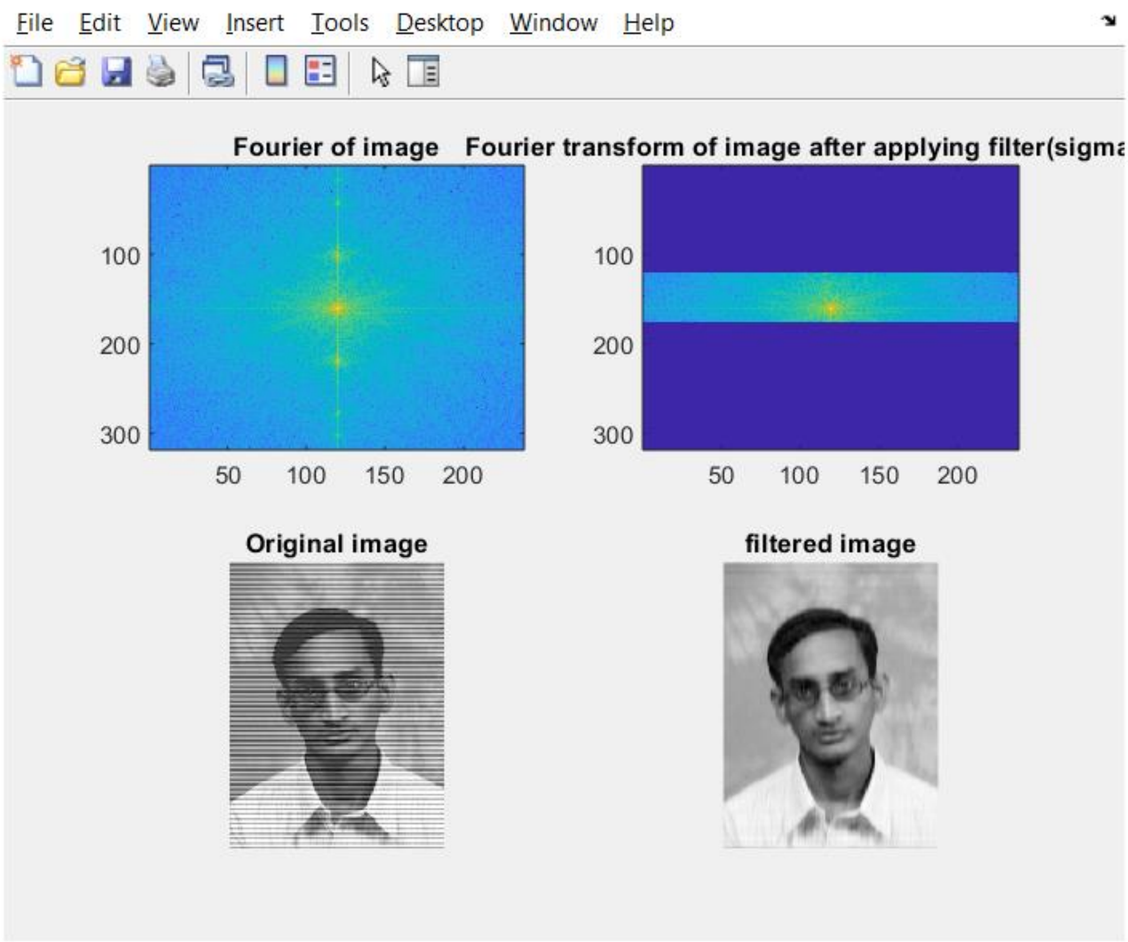
Q1,4)

This is a salt and pepper type noise so median filter would be best suitable for it.



Q1,5)

Plotting the fft we see the distribution of noise as distributed in a rectangle around the data .So I created a filter of rectangle which filters out the noise keeping the data.



Q2 a)

We have **N** square filters each of size **F**. Let the size of the image be **(Width, Height, Channels).** The convolution is done with a step size of **S** units, and the input is also padded with a zero padding of **Z.**

The size of image after zero padding will be **[W + 2Z, H + 2Z, Channels]**.

Assuming we zero pad after applying each filter

Now the filter with step length S units applied. Hence the output size will be $$ output width = (W - F + 2*Z)/S + 1 output height = (H- F + 2*Z)/S + 1.

Total convolution in a channel = output width * output height. We will apply this recursively. Let the output of image after (i-1)^th convolution be W_{i-1},H_{i-1}. Then W_{i} = (W_{i-1} - F + 2Z)/S + 1 H_{i} = (H_{i-1} - F + 2Z)/S+1.

Q2 b)

For one convolution:- The number of multiplications at each step is F^2 and the number of additions is F^2-1. The total number of additions and multiplications for one input channel will be (2F^2 -1) * output width * output height.

For N such convolution:

- Number of additions after N convolutions will be ∑{i=0}^{n-1} (W_i * H_i) * (F^2-1) * channels
- Number of multiplications after N convolutions will be ∑sum_{i=0}^{ n-1} (W_i * H_i) * F^2 * channels
- Total operations will be ∑{i=0}^{n-1} (W_i * H_i) * (2F^2-1) * channels

Q3)

In this problem, we need to figure of the phone number dialed, if we know the sounds of the individual dial tones.

Approach

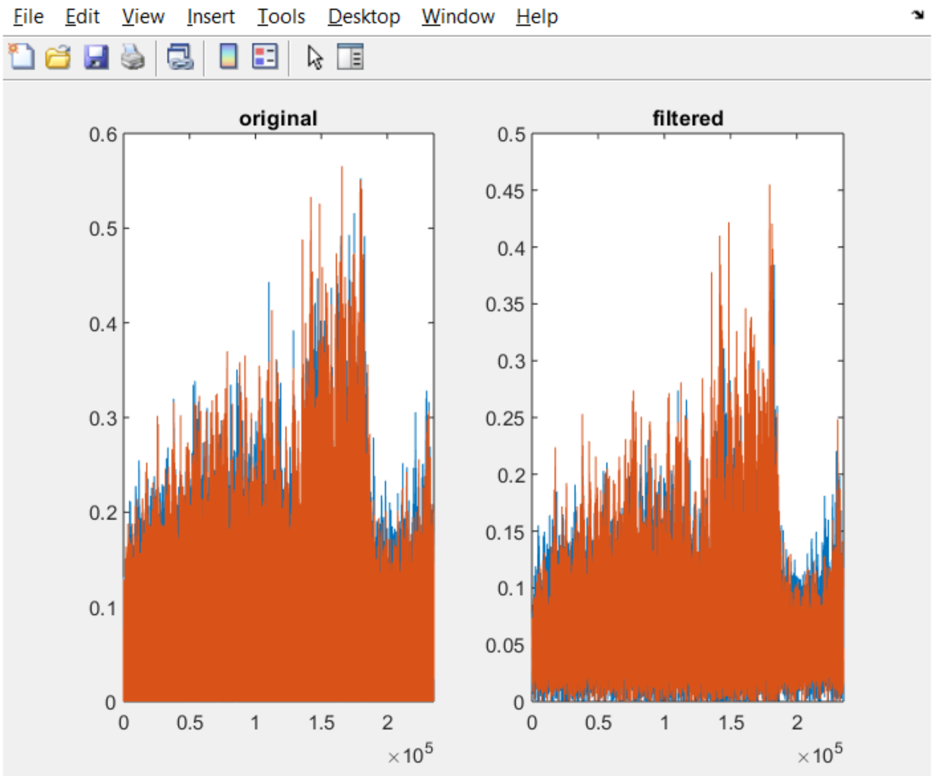
- First store the different outputs in a an array of different tunes.
- Then figure out the total number of dial tones present in input signal and also the start and end time of each dial tone in the number.
- Now in the input audio file separate into chunks of tones of numbers according to sample rate.
- For each sample, take the product o of that sample with each of the dial tone. The dial tone with which we get the maximum value of product will be the dial tone in that particular sample.

Q4)

My signal is sig3.

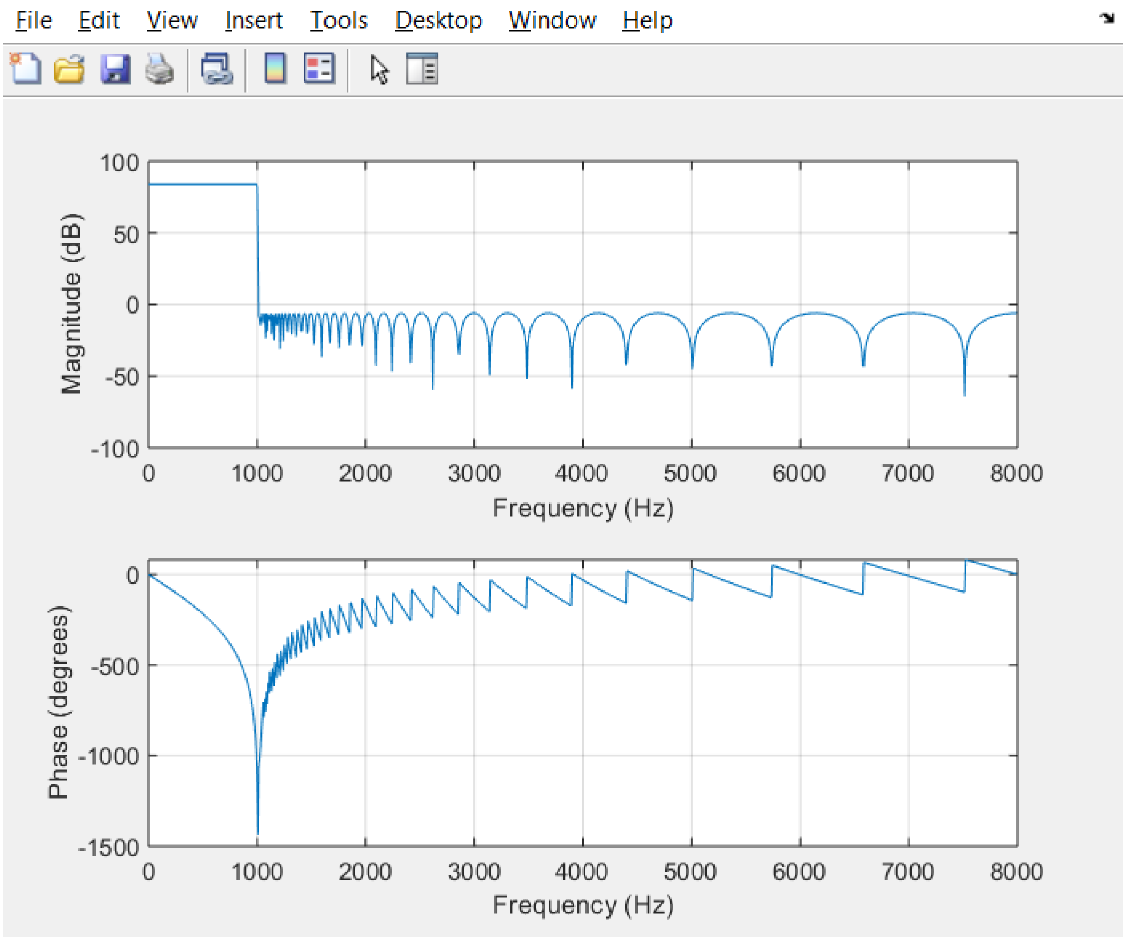
I have formed two solutions:

1)Here I take the fft of the signal and then remove those values which are less than 3*mean of the fft frequencies;

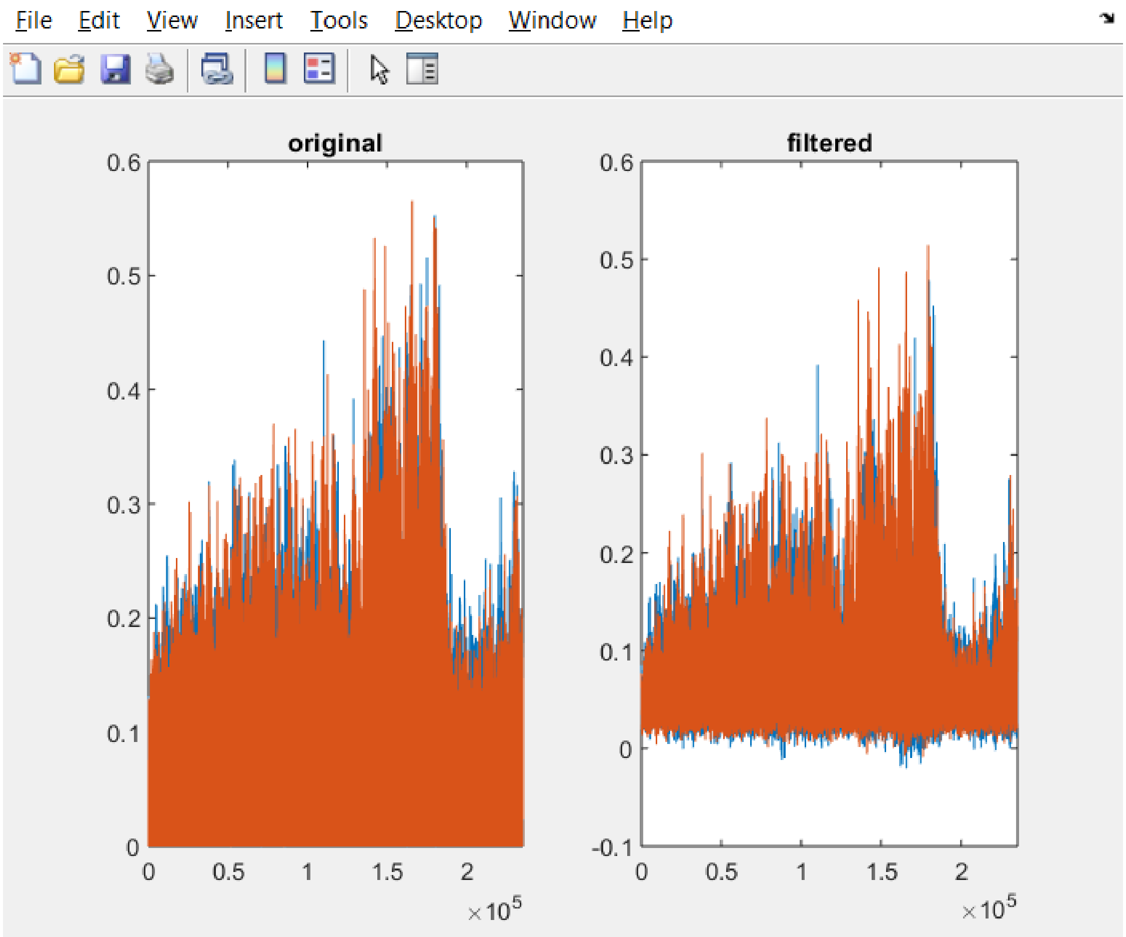


But in this case the distortion of the signal is to a great extent.

2)Here I have designed a low pass Chebyshev filter to filter out the gaussian(white noise)



File Edit View Insert Tools Desktop Window Help
Magnitude (dB)
100
50
0
-50
-100
0 1000 2000 3000 4000 5000 6000 7000 8000
Frequency (Hz)
Phase (degrees)
0
-500
-1000
-1500
0 1000 2000 3000 4000 5000 6000 7000 8000
Frequency (Hz)

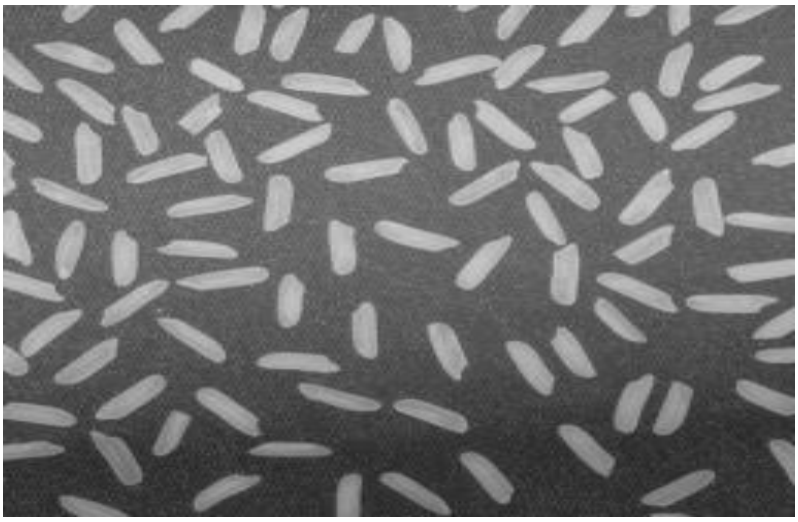


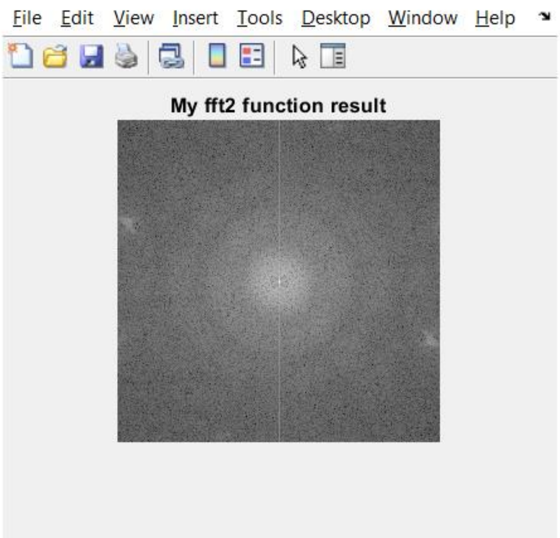
Q5)

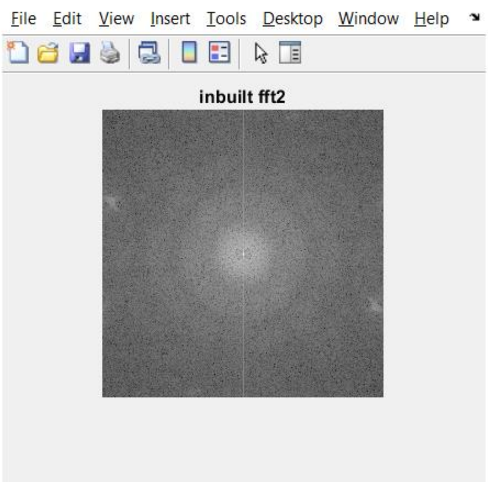
- To compute 2D FFT we need to
  - do 1D FFT on each row (real to complex) . . . . (1)
  - do 1D FFT on each column resulting from (1) (complex to complex)
- So, we need to write code 1D FFT. I used this algorithm to write code for 1D FF

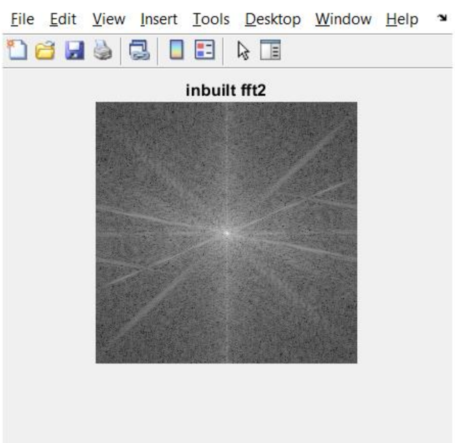
RECURSIVE-FFT(a)

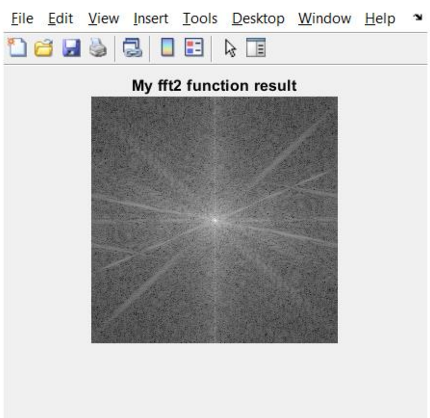
$n \leftarrow length[a]$ ▷ n is a power of 2.
if $n = 1$
  then return a
$\omega_n \leftarrow e^{2\pi i/n}$
$\omega \leftarrow 1$
$a^{[0]} \leftarrow (a_0, a_2, \ldots, a_{n-2})$
$a^{[1]} \leftarrow (a_1, a_3, \ldots, a_{n-1})$
$y^{[0]} \leftarrow$ RECURSIVE-FFT($a^{[0]}$)
$y^{[1]} \leftarrow$ RECURSIVE-FFT($a^{[1]}$)
for $k \leftarrow 0$ **to** $n/2 - 1$
  do $y_k \leftarrow y_k^{[0]} + \omega y_k^{[1]}$
    $y_{k+(n/2)} \leftarrow y_k^{[0]} - \omega y_k^{[1]}$
    $\omega \leftarrow \omega \omega_n$
return y ▷ y is assumed to be column vector.

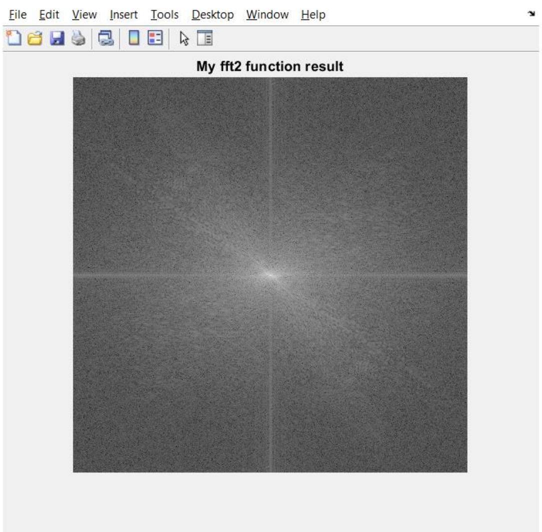


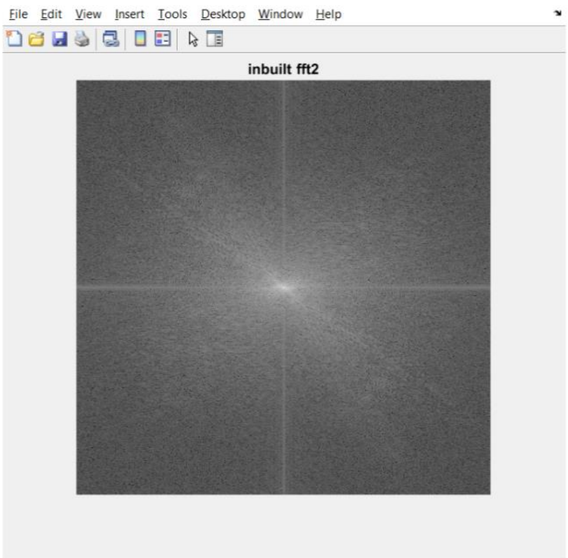





File Edit View Insert Tools Desktop Window Help
inbuilt fft2


File Edit View Insert Tools Desktop Window Help
My fft2 function result





On length of 1024

Inbuilt Elapsed time is 0.004953 seconds.

Fft Elapsed time is 0.116509 seconds.

Dft Elapsed time is 0.287425 seconds

On length of 2^14

Inbuilt Elapsed time is 0.001033 seconds.

Fft Elapsed time is 1.581128 seconds.

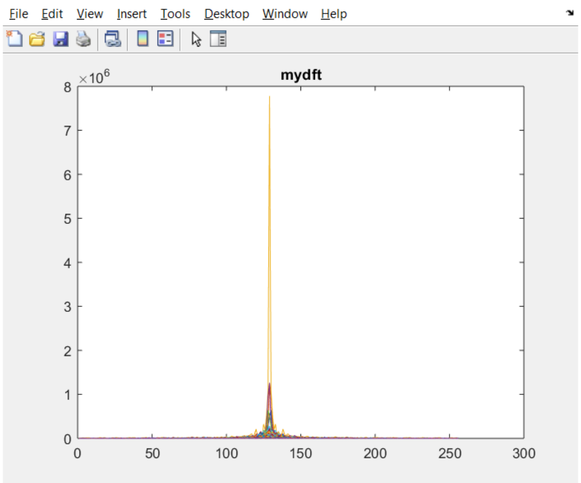
Dft Elapsed time is 155.222063 seconds.

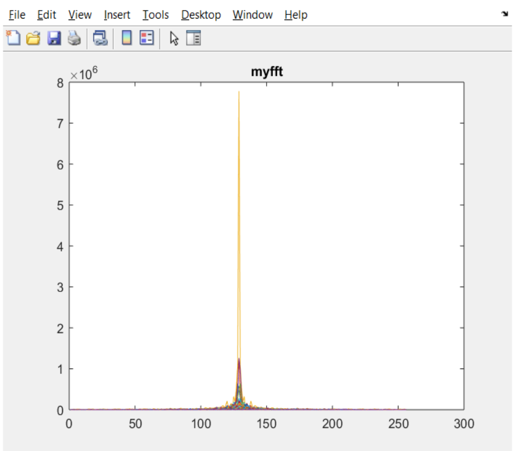
On cameraman.tif

Inbuilt Elapsed time is 0.000694 seconds.

Fft Elapsed time is 1.040118 seconds.

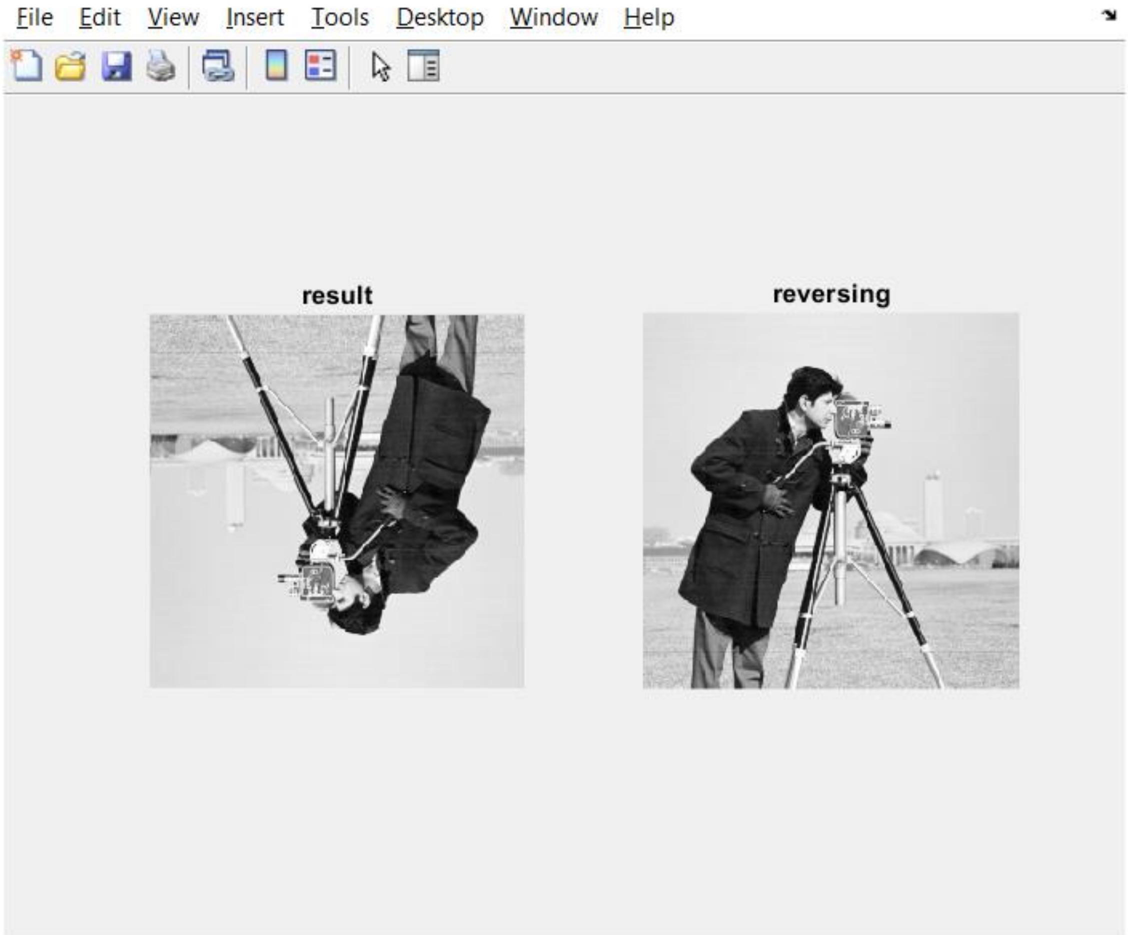
Dft Elapsed time is 6.000722 seconds.





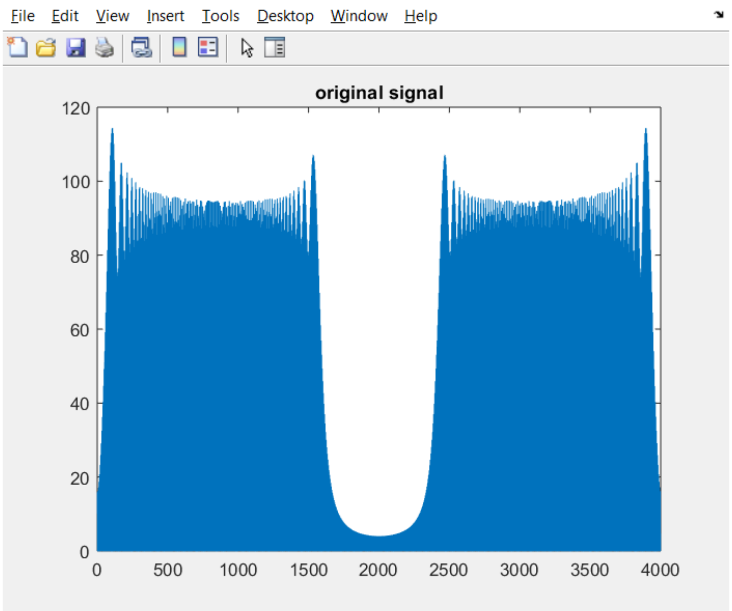
Q6)

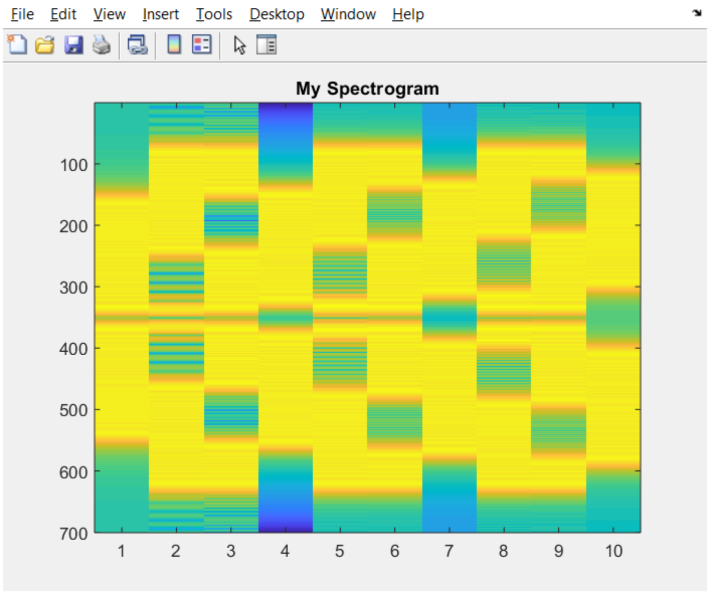
When we do fft(fft(img)) then we get inverted image.

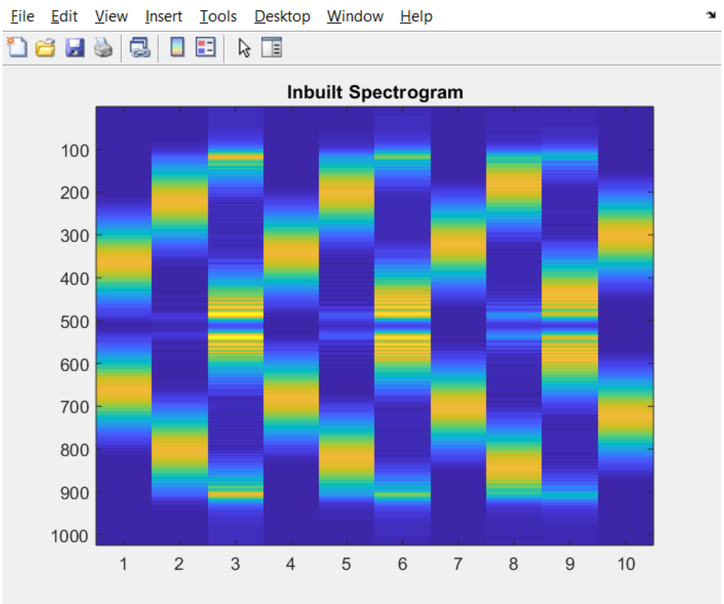


The result of applying fft two times is reversed as fft and ifft are conjugates of each other and when we apply duality property we get x(-t) = fft(fft(x(t)));

Q7)

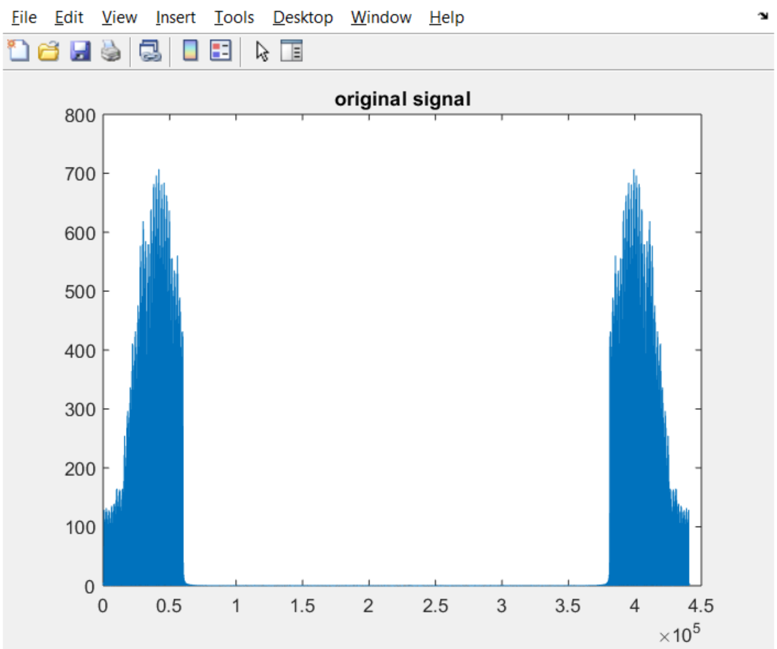

File Edit View Insert Tools Desktop Window Help
original signal
120
100
80
60
40
20
0
0 500 1000 1500 2000 2500 3000 3500 4000

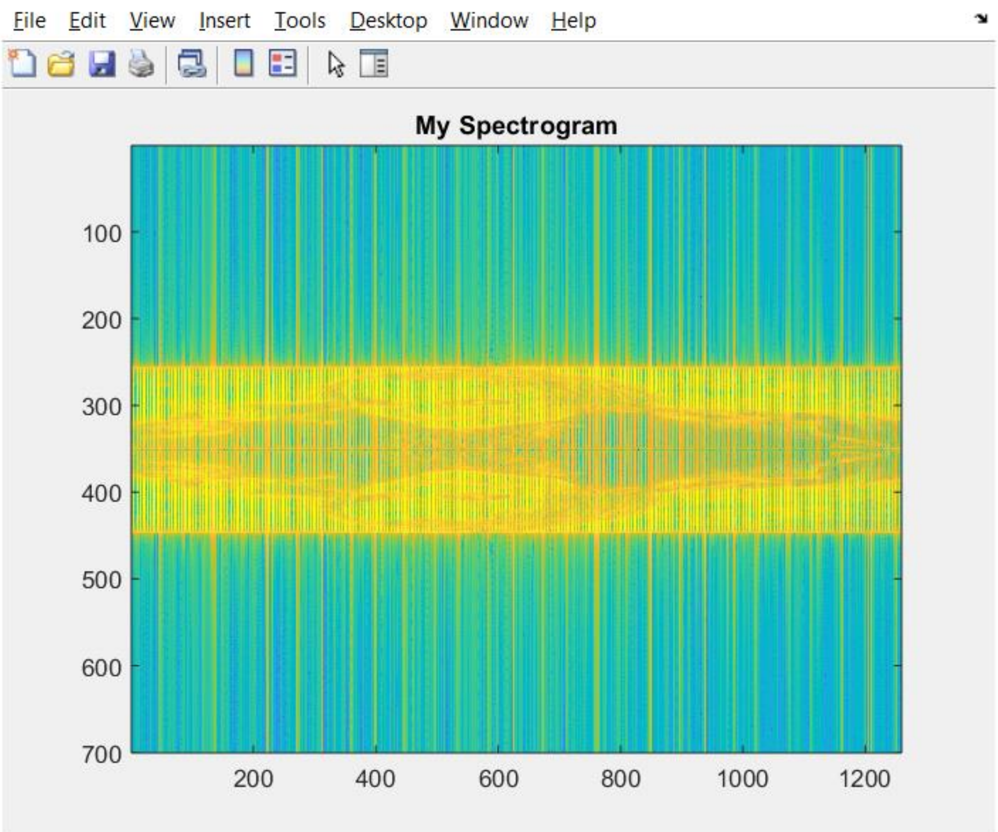


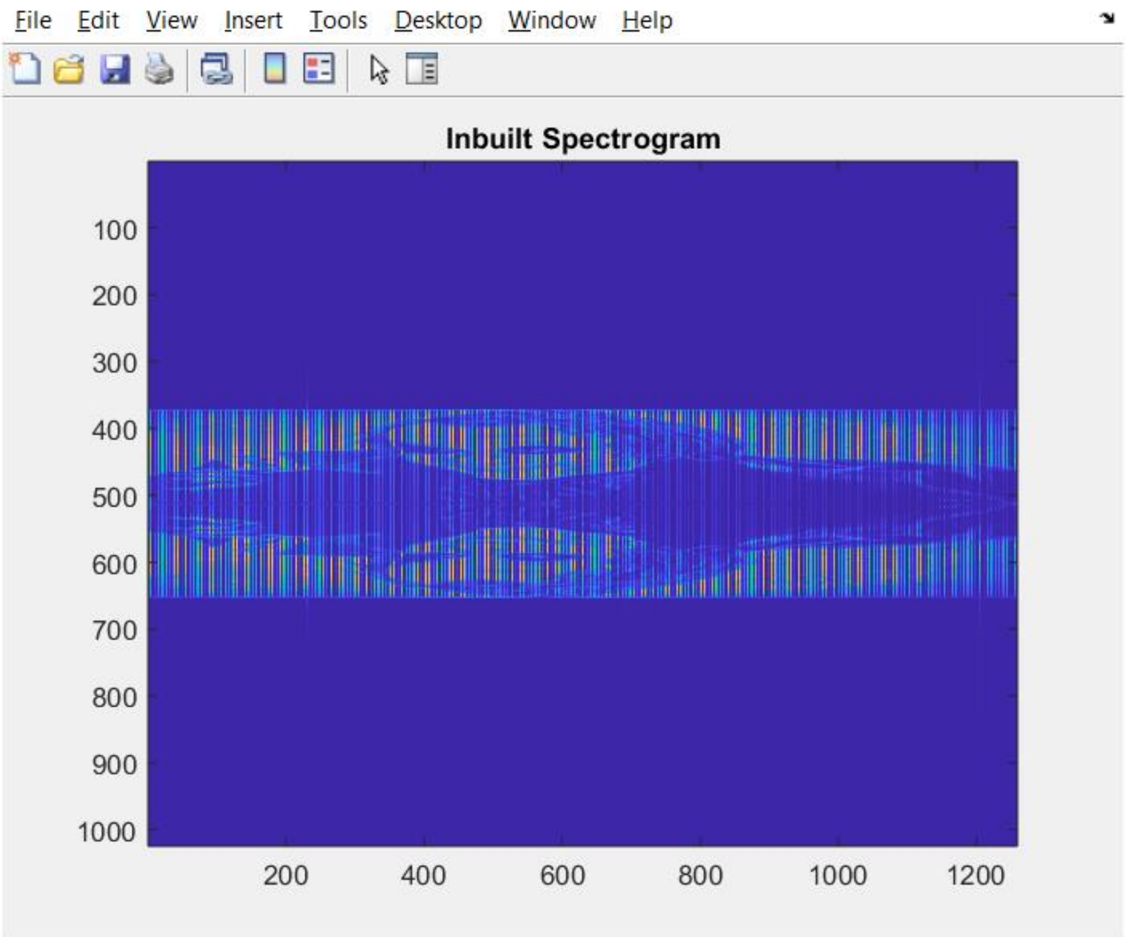


Q7,2)

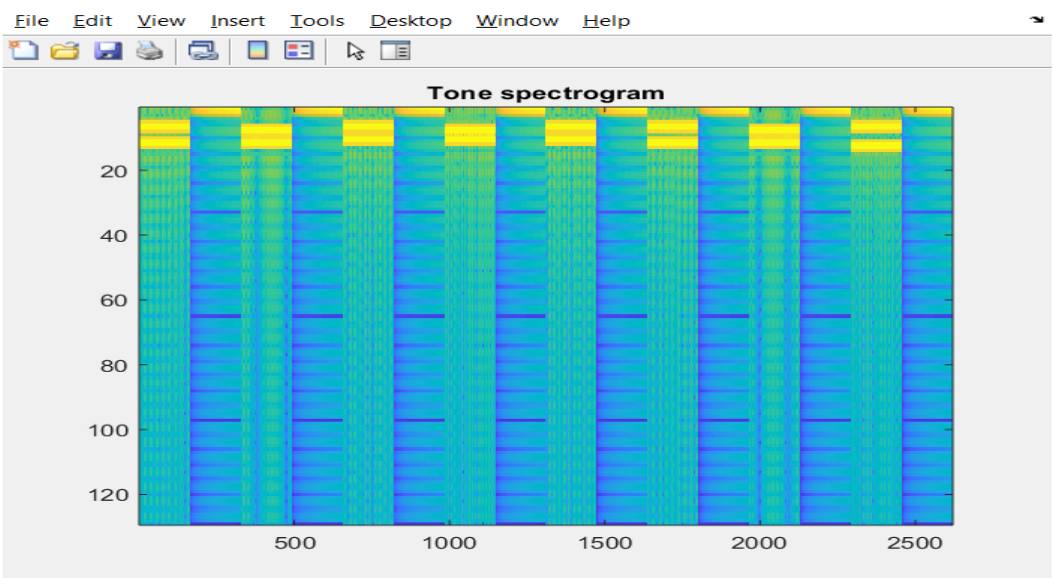
The password must be “Joker” as the per the spectrogram.


File Edit View Insert Tools Desktop Window Help
original signal
800
700
600
500
400
300
200
100
0
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5
×10^5


File Edit View Insert Tools Desktop Window Help
My Spectrogram
100
200
300
400
500
600
700
200 400 600 800 1000 1200


File Edit View Insert Tools Desktop Window Help
Inbuilt Spectrogram
100
200
300
400
500
600
700
800
900
1000
200
400
600
800
1000
1200

Q7,3)


File Edit View Insert Tools Desktop Window Help
Tone spectrogram
20
40
60
80
100
120
500
1000
1500
2000
2500