EE5175:IMAGE SIGNAL PROCESSING

LAB-7

AQIL K H

EE20S049

1. **Implement 2D DFT using row-column decomposition.**

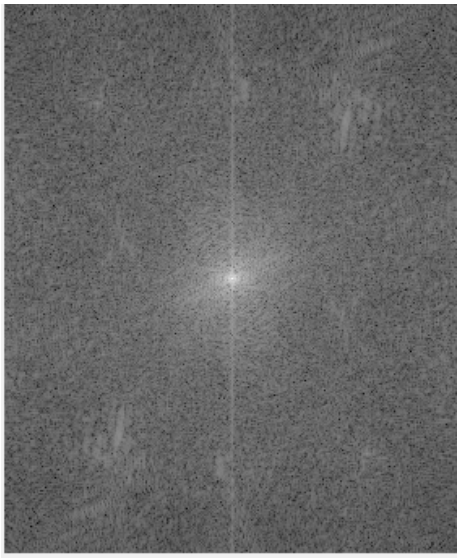
```
%%%Function to find  2D DFT using row-column decomposition.
function DFT=findDFT(Image)
[rows,colms]=size(Image);                %Finding the rows and column size
Intermediate=zeros(size(Image));         %Initializing the Intermediate matrix
DFT=zeros(size(Image));                  %Initializing the DFT matrix
%Taking Rows and finding DFT of each row
for i=1:rows
    Intermediate(i,:)=fft(Image(i,:));
end
%Taking Columns of intermediate and finding DFT of each column
for i=1:colms
    DFT(:,i)=fft(Intermediate(:,i));
end
end
```

2. **Compute DFTs F1(k, l) = |F1(k, l)|exp(jφ1(k,l)) and F2(k, l) = |F2(k, l)| exp(jφ2(k,l)) of I1(fourier.png) and I2 (fourier transform.png) respectively. Arrive at two new images I3 and I4 such that their DFTs are, respectively, F3(k, l) = |F1(k, l)|exp(jφ2(k,l)) and F4(k, l) = |F2(k, l)|exp(jφ1(k,l)) .**

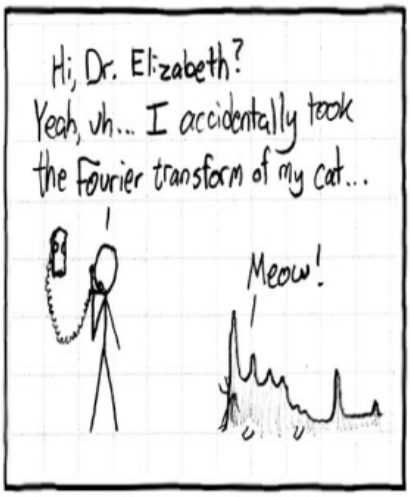
Outputs:

Fourier.png:

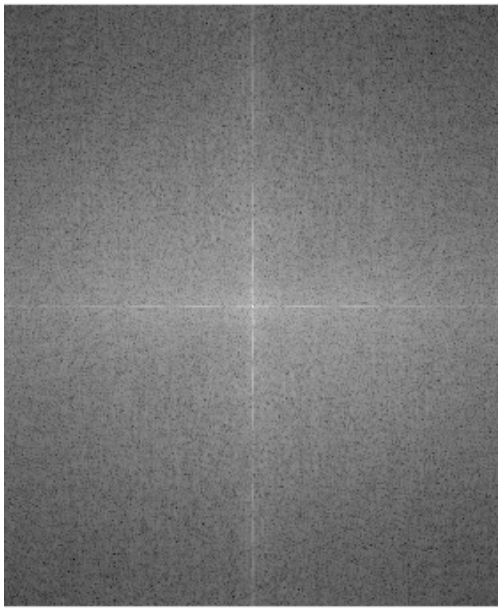
2D-DFT:



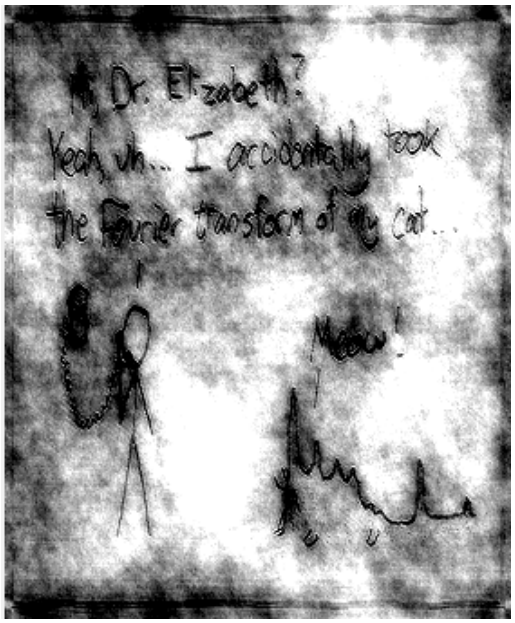
Fourier transform.png:



2D DFT:



I3:

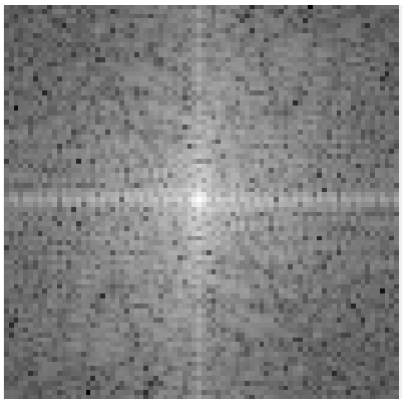

Dr. Elizabeth?
Yeah, uh... I accidentally took
the Fourier transform of my cat...
Meow!

I4:

3. Verify the rotation property of 2D DFT using peppers small.pgm.

Input:

90 degree Rotated DFT:



2D IDFT of Rotated DFT:

Image Rotated by 90 degree:

Observation: Verified the rotation property of 2D DFT