Bài tập 4

Họ và tên: Trần Gia Long

MSSV: N20DCCN035

Lớp: D20CQCNPM01-N

Bài làm

Câu 1:

Khai báo thư viện cần thiết:

import numpy as np

import cv2

import matplotlib.pyplot as plt

import math

Tạo mảng từ 0 đến 7

arr=np.array(list(float(y) for y in range(0,8)))

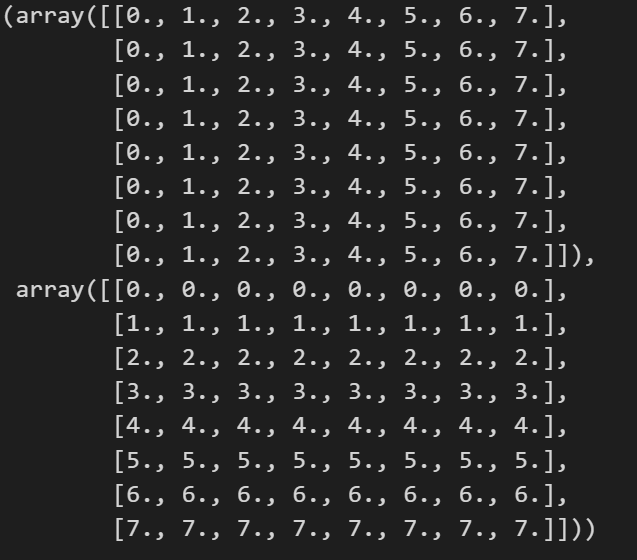
arr



Tạo ma trận COLUMN(m) và ROW(n)

[COLUMN,ROW]=np.meshgrid(arr,arr)

COLUMN,ROW



Tạo I1 theo công thức

u0=2.0

v0=2.0

def create\_I1():

    I1=0.5 \* np.exp(2j\*np.pi/8\*(u0\*COLUMN + v0\*ROW))

    return I1

I1=create\_I1()

Hàm tạo ảnh căn giữa, ảnh real, ảnh imaginary

def create\_imaginary(I1):

    return np.round(np.imag(I1) \* (10\*\*4))\*(10\*\*(-4))

def create\_real(I1):

    return np.round(np.real(I1) \* (10\*\*4))\*(10\*\*(-4))

def create\_center(I1):

    return (np.fft.fftshift(np.fft.fft2(I1)))

Hàm tạo cân bằng tương phản

def create\_full\_contrast(I):

    min=np.min(I)

    max=np.max(I)

    if(min==max):# các phần tử giống nhau

        return np.zeros(I.shape)

    scale=255/(max-min)

    return np.round((I-min)\*scale)

In hình real và imag của I1

real\_I1=create\_real(I1)

imag\_I1=create\_imaginary(I1)

plt.figure(figsize=(10, 10))

plt.subplot(2, 2, 1)

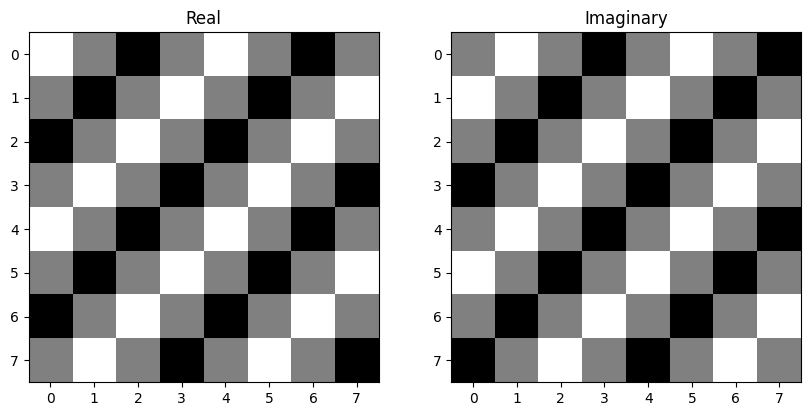
plt.title("Real")

plt.imshow(create\_full\_contrast(real\_I1),cmap='gray')

plt.subplot(2, 2, 2)

plt.title("Imaginary")

plt.imshow(create\_full\_contrast(imag\_I1),cmap='gray')



Tạo căn giữa DFT và in ma trận real và imag của nó

Itilde1=create\_center(I1)

Itilde1\_real=create\_real(Itilde1)

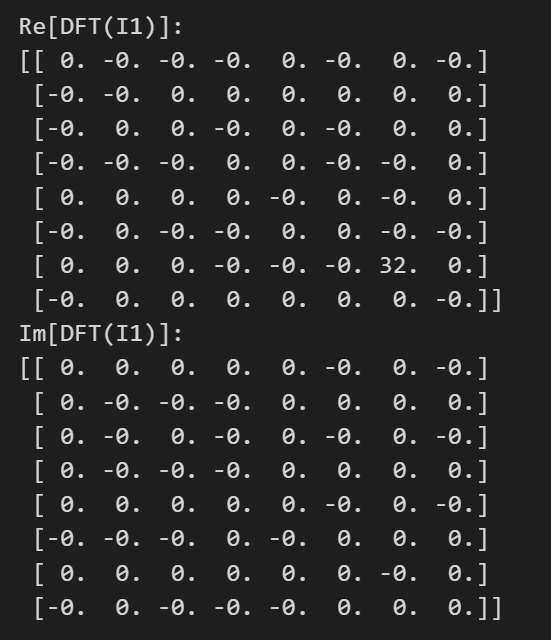
Itilde1\_imag=create\_imaginary(Itilde1)

print('Re[DFT(I1)]:')

print(Itilde1\_real)

print('Im[DFT(I1)]:')

print(Itilde1\_imag)



Câu 2:

Tạo I2

def create\_I2():

    I2=0.5 \* np.exp((-1)\*2j\*np.pi/8\*(u0\*COLUMN + v0\*ROW))

    return I2

I2=create\_I2()

real\_I2=create\_real(I2)

imag\_I2=create\_imaginary(I2)

In hình

plt.figure(figsize=(10, 10))

plt.subplot(2, 2, 1)

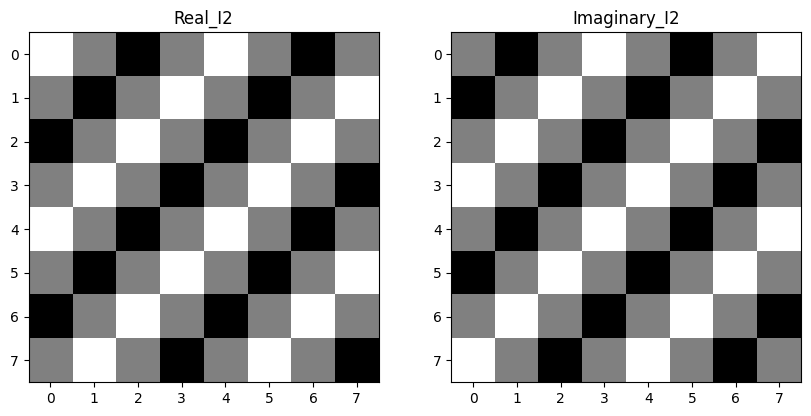
plt.title("Real\_I2")

plt.imshow(create\_full\_contrast(real\_I2),cmap='gray')

plt.subplot(2, 2, 2)

plt.title("Imaginary\_I2")

plt.imshow(create\_full\_contrast(imag\_I2),cmap='gray')



Tạo căn giữa DFT và real, imag của nó

Itilde2=create\_center(I2)

Itilde2\_real=create\_real(Itilde2)

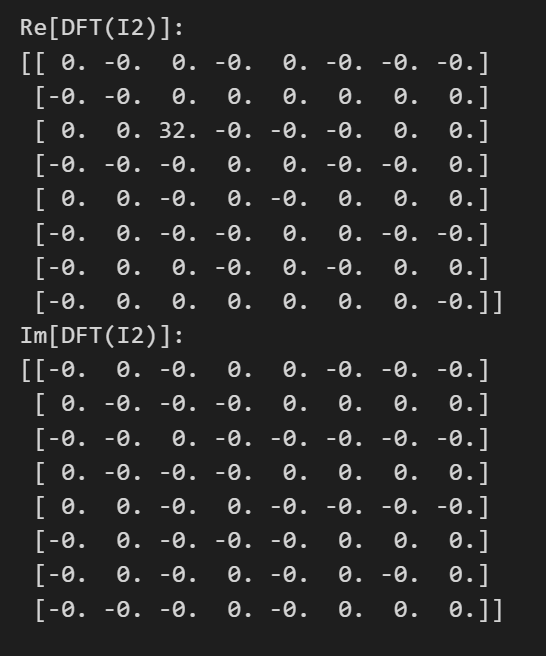
Itilde2\_imag=create\_imaginary(Itilde2)

print('Re[DFT(I2)]:')

print(Itilde2\_real)

print('Im[DFT(I2)]:')

print(Itilde2\_imag)



Câu 3:

Tạo I3

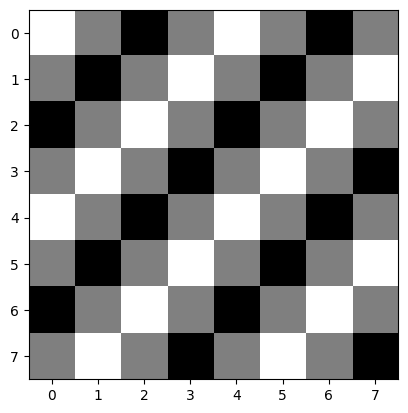
def create\_I3():

    I3=np.cos((2\*np.pi/8)\*(u0\*COLUMN+v0\*ROW))

    return I3

I3=create\_I3()

plt.imshow(create\_full\_contrast(I3),cmap='gray')



Kiểm tra công thức

False in (I3==I1+I2)



I3 đúng bằng I1+I2

Tạo căn giữa DFT

Itilde3=create\_center(I3)

Itilde3\_real=create\_real(Itilde3)

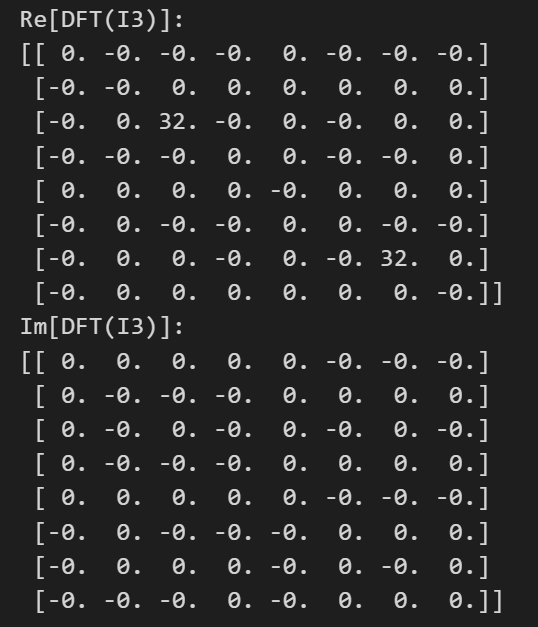
Itilde3\_imag=create\_imaginary(Itilde3)

print('Re[DFT(I3)]:')

print(Itilde3\_real)

print('Im[DFT(I3)]:')

print(Itilde3\_imag)



Câu 4:

Tạo I4

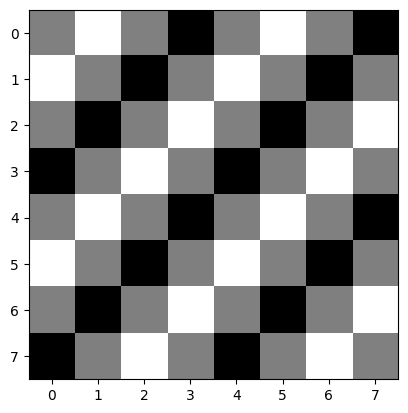
def create\_I4():

    I4=np.sin((2\*np.pi/8)\*(u0\*COLUMN+v0\*ROW))

    return I4

I4=create\_I4()

plt.imshow(create\_full\_contrast(I4),cmap='gray')



Kiểm tra công thức

False in (I4==-1j\*(I1-I2))



I4 đúng bằng –j \*(I1-I2)

Tạo căn giữa DFT và real, imag của nó

Itilde4=create\_center(I4)

Itilde4\_real=create\_real(Itilde4)

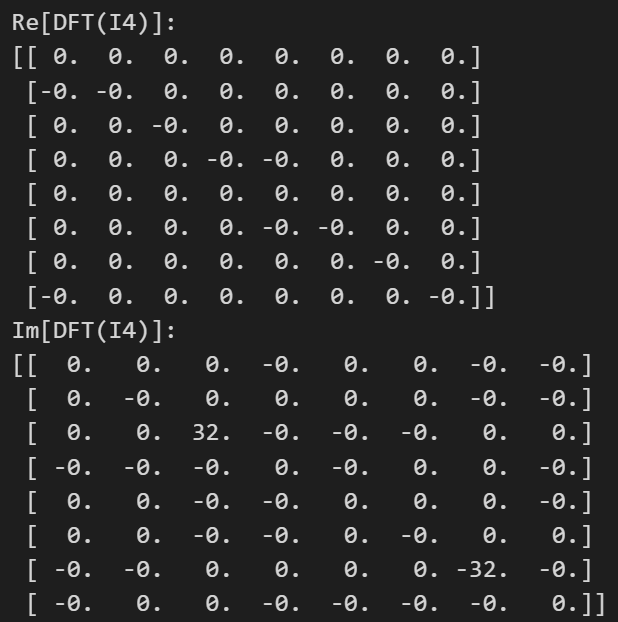
Itilde4\_imag=create\_imaginary(Itilde4)

print('Re[DFT(I4)]:')

print(Itilde4\_real)

print('Im[DFT(I4)]:')

print(Itilde4\_imag)



Câu 5:

Tạo I5

u1=1.5

v1=1.5

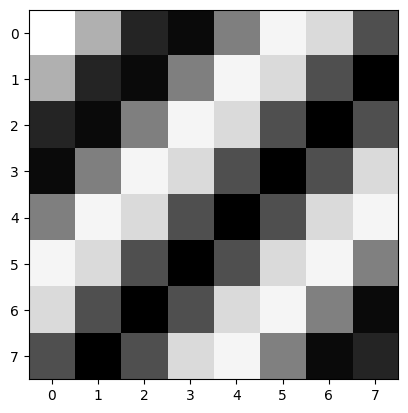
def create\_I5():

    I5=np.cos((2\*np.pi/8)\*(u1\*COLUMN+v1\*ROW))

    return I5

I5=create\_I5()

plt.imshow(create\_full\_contrast(I5),cmap='gray')



Tạo real và imag của I5

I5\_real=create\_real(I5)

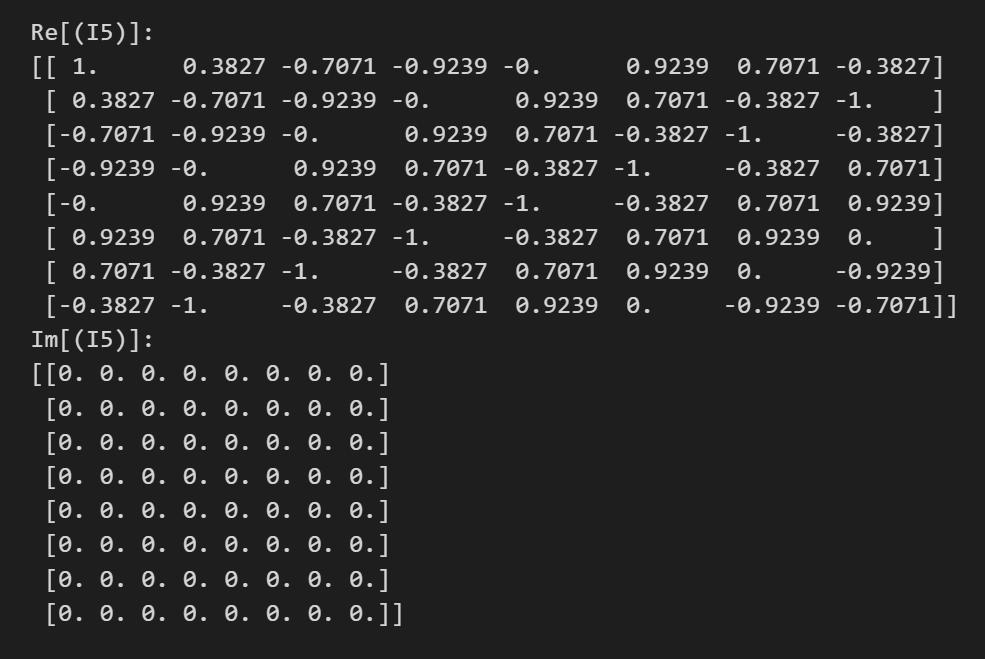
I5\_imag=create\_imaginary(I5)

print('Re[(I5)]:')

print(I5\_real)

print('Im[(I5)]:')

print(I5\_imag)



Tạo căn giữa DFT và real, imag của nó

Itilde5=create\_center(I5)

Itilde5\_real=create\_real(Itilde5)

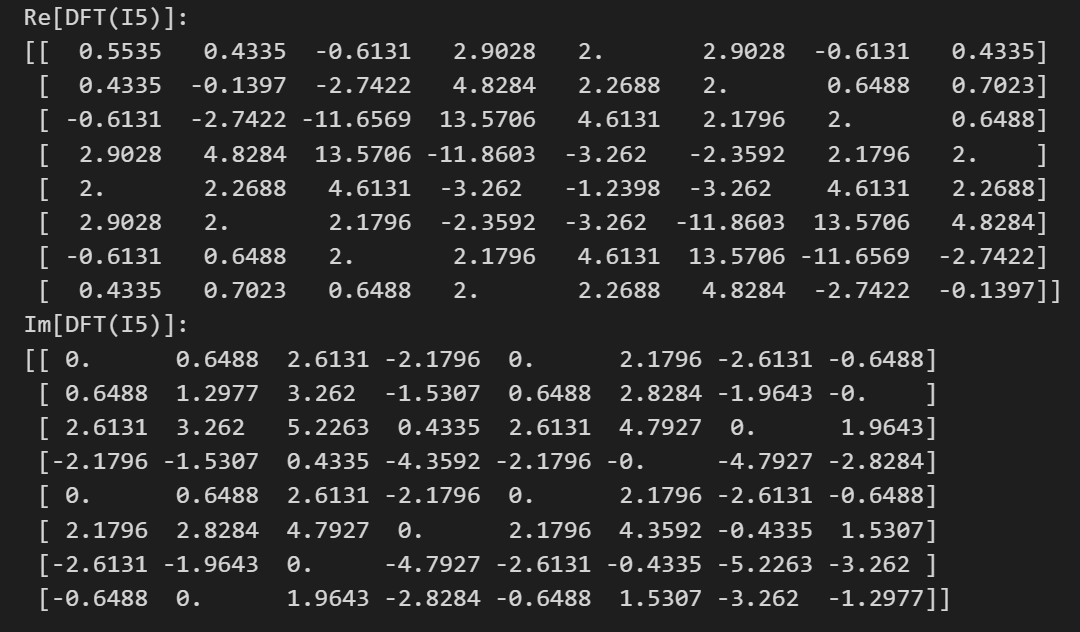
Itilde5\_imag=create\_imaginary(Itilde5)

print('Re[DFT(I5)]:')

print(Itilde5\_real)

print('Im[DFT(I5)]:')

print(Itilde5\_imag)



Nhận xét:

Đối với câu 1 và 2: Mỗi ảnh I1 và I2 đều được tính bằng hàm cơ sở DFT(1). Nên DFT I1 và I2 đều chỉ có một số khác không.

(1) 

Đối với câu 3 và 4: Mỗi ảnh I3 và I4 là tổng của hai hàm cơ sở DFT. Nên mỗi DFT có 2 số khác không.

Tuy nhiên với I5 có tần số khác các ảnh còn lại. Thế nên cần nhiều số khác không hơn để tạo thành.

Câu 6:

Tải hình

cam\_img=np.fromfile('camera.bin',dtype=np.uint8).reshape(256,256)

head\_img=np.fromfile('head.bin',dtype=np.uint8).reshape(256,256)

eyeR\_img=np.fromfile('eyeR.bin',dtype=np.uint8).reshape(256,256)

sale\_img=np.fromfile('salesman.bin',dtype=np.uint8).reshape(256,256)

Hiển thị lên

plt.figure(figsize=(10, 10))

plt.subplot(2, 2, 1)

plt.title("Camera")

plt.imshow(cam\_img,cmap='gray')

plt.subplot(2, 2, 2)

plt.title("Head")

plt.imshow(head\_img,cmap='gray')

plt.subplot(2, 2, 3)

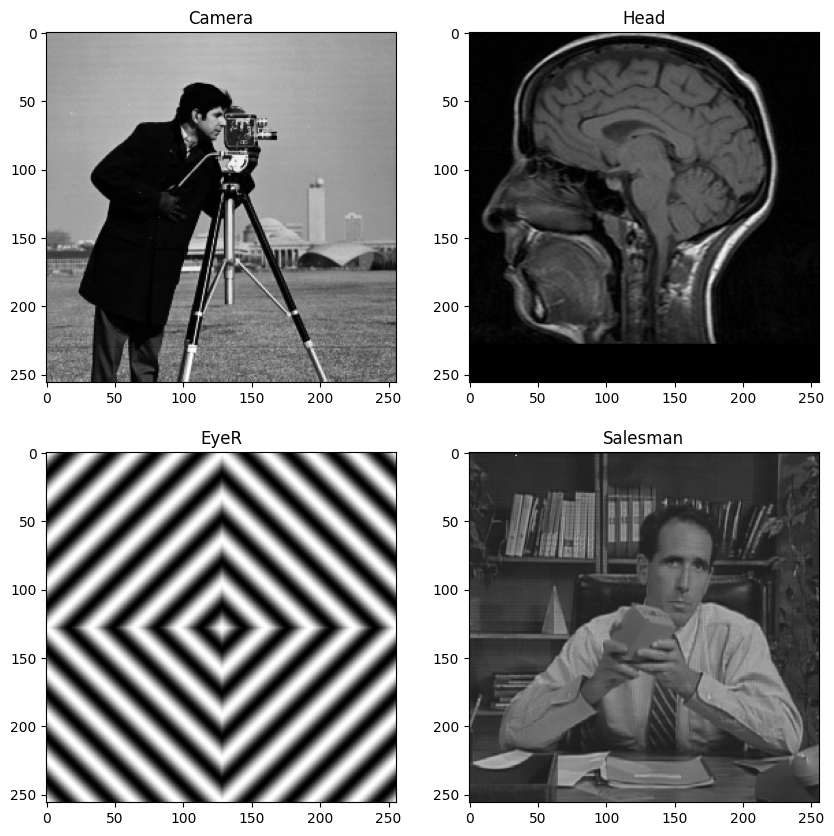
plt.title("EyeR")

plt.imshow(eyeR\_img,cmap='gray')

plt.subplot(2, 2, 4)

plt.title("Salesman")

plt.imshow(sale\_img,cmap='gray')



Vẽ 5 hình với mỗi đối tượng hình ảnh gồm: gốc, real, imaginary, log-magnitude spectrum, phase của căn giữa DFT

def draw\_5\_image(img):

    plt.figure(figsize=(10, 10))

    plt.subplot(1, 5, 1)

    plt.title("Origin")

    plt.imshow(create\_full\_contrast(img),cmap='gray')

    center\_img=create\_center(img)

    real\_img=create\_real(center\_img)

    imag\_img=create\_imaginary(center\_img)

    plt.subplot(1, 5, 2)

    plt.title("Real")

    plt.imshow(create\_full\_contrast(real\_img),cmap='gray')

    plt.subplot(1, 5, 3)

    plt.title("Imaginary")

    plt.imshow(create\_full\_contrast(imag\_img),cmap='gray')

    log\_magnitude\_spectrum=np.log(np.abs(center\_img)+1)

    phase\_spectrum=np.angle(center\_img)

    plt.subplot(1, 5, 4)

    plt.title("Log-magnitude")

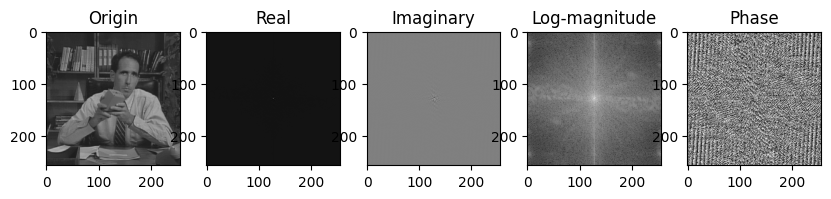
    plt.imshow(create\_full\_contrast(log\_magnitude\_spectrum),cmap='gray')

    plt.subplot(1, 5, 5)

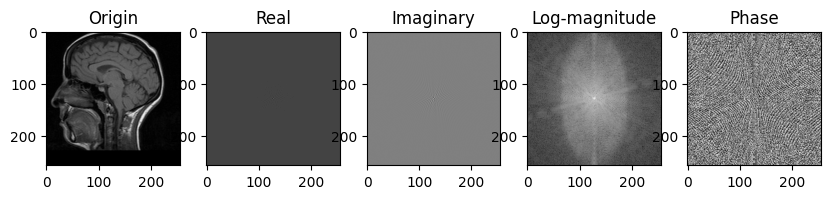
    plt.title("Phase")

    plt.imshow(create\_full\_contrast(phase\_spectrum),cmap='gray')

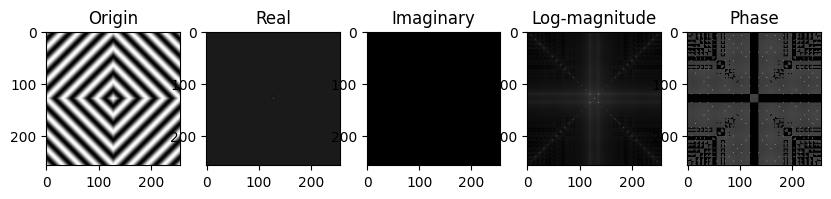
Salesman



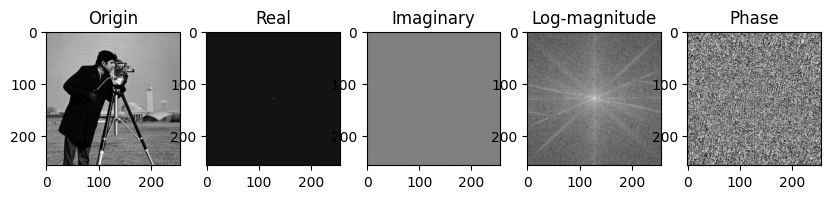
Head



EyeR



Camera



Câu 7:

Tạo I6, J^(J1) và J2

I6=cam\_img

DFT\_I6=np.fft.fft2(I6)

mag=np.abs(DFT\_I6)

phase=np.angle(DFT\_I6)

real\_J1=create\_real(np.fft.ifft2(mag))

real\_J2=create\_real(np.fft.ifft2(np.exp(1j\*phase)))

plt.figure(figsize=(10,10))

plt.subplot(1,2,1)

plt.title('JJ1')

JJ1=np.log(real\_J1)

plt.imshow(create\_full\_contrast(JJ1),cmap='gray')

plt.subplot(1,2,2)

plt.title('J2')

plt.imshow(create\_full\_contrast(real\_J2),cmap='gray')

