UNIVERSITY OF ENGINEERING AND TECHNOLOGY MARDAN

ASSIGNMENT # 02

SUBMITTED TO :

DR. IRFAN KHAN

SUBMITTED BY:

M.AAZAIN HAMEED BADSHAH

REG NO : 21MDELE229

DEPARTMENT : ELECTRICAL ENGINEERING

BATCH : 4TH

COURSE: DIGITAL IMAGE PROCESSING

Task:

Take a black & White image of yourself. Write a MATLAB code to perform:

a. Load the image in MATLAB

b. Multiply the image by (−1)$x+y$

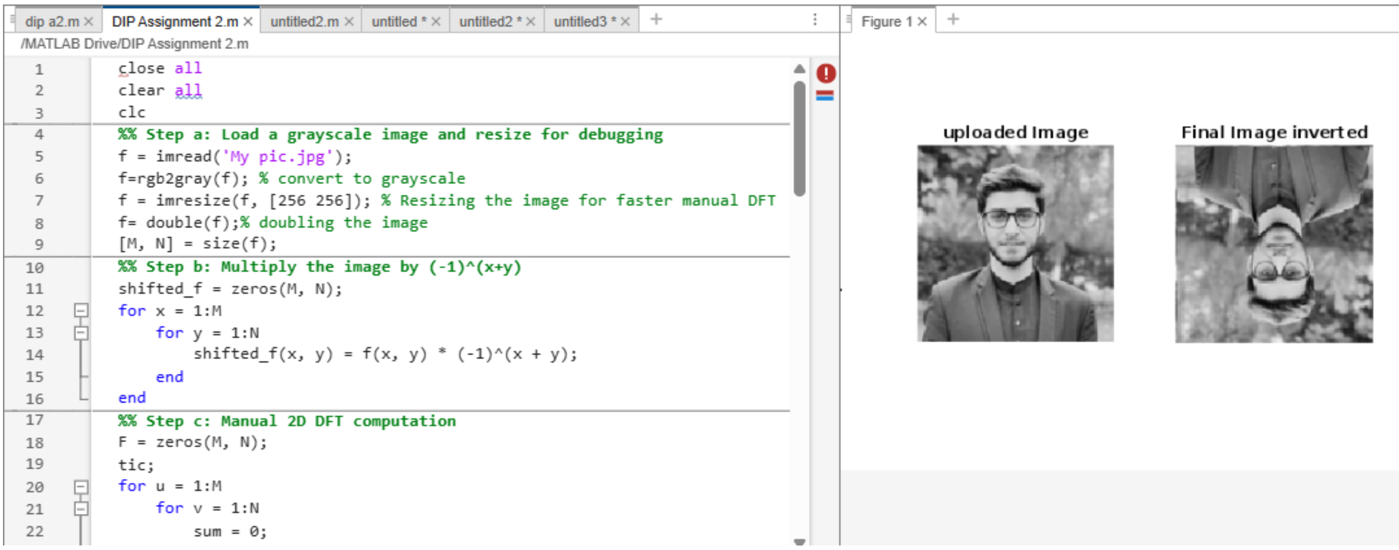
c. Computing the DFT using the equation shown below. Do not use the built-in Matlab function (FFT2)

d. Take the complex conjugate of the transform $F(u,v)$

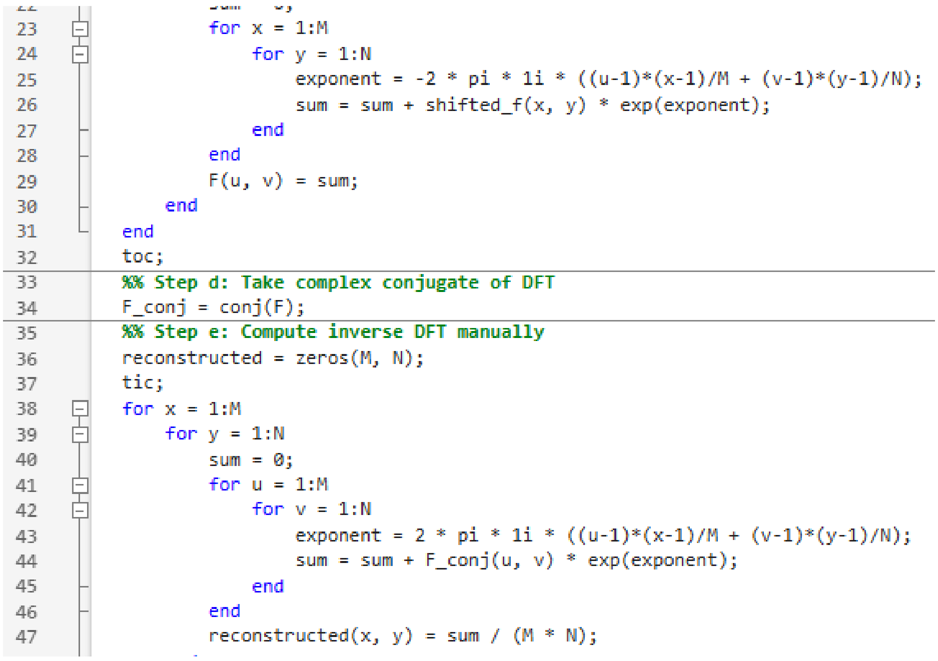
e. Compute the inverse DFT

f. Multiply the real part of the result by (−1)$x+y$ g. Display the original and the final images side by side.

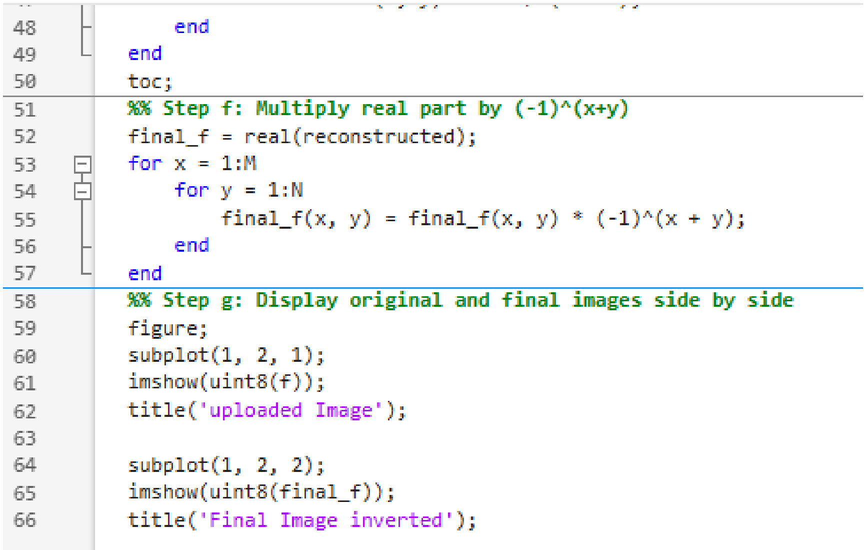
Code:



```
close all
clear all
clc
%% Step a: Load a grayscale image and resize for debugging
f = imread('My pic.jpg');
f=rgb2gray(f); % convert to grayscale
f = imresize(f, [256 256]); % Resizing the image for faster manual DFT
f= double(f);% doubling the image
[M, N] = size(f);
%% Step b: Multiply the image by (-1)^(x+y)
shifted_f = zeros(M, N);
for x = 1:M
    for y = 1:N
        shifted_f(x, y) = f(x, y) * (-1)^(x + y);
    end
end
%% Step c: Manual 2D DFT computation
F = zeros(M, N);
tic;
for u = 1:M
    for v = 1:N
        sum = 0;
```



```
        sum = 0;
        for x = 1:M
            for y = 1:N
                exponent = -2 * pi * 1i * ((u-1)*(x-1)/M + (v-1)*(y-1)/N);
                sum = sum + shifted_f(x, y) * exp(exponent);
            end
        end
        F(u, v) = sum;
    end
end
toc;
%% Step d: Take complex conjugate of DFT
F_conj = conj(F);
%% Step e: Compute inverse DFT manually
reconstructed = zeros(M, N);
tic;
for x = 1:M
    for y = 1:N
        sum = 0;
        for u = 1:M
            for v = 1:N
                exponent = 2 * pi * 1i * ((u-1)*(x-1)/M + (v-1)*(y-1)/N);
                sum = sum + F_conj(u, v) * exp(exponent);
            end
        end
        reconstructed(x, y) = sum / (M * N);
```



```
    end
end
toc;
%% Step f: Multiply real part by (-1)^(x+y)
final_f = real(reconstructed);
for x = 1:M
    for y = 1:N
        final_f(x, y) = final_f(x, y) * (-1)^(x + y);
    end
end
%% Step g: Display original and final images side by side
figure;
subplot(1, 2, 1);
imshow(uint8(f));
title('uploaded Image');

subplot(1, 2, 2);
imshow(uint8(final_f));
title('Final Image inverted');
```