EXP 7 – HOMOMORPHIC FILTER

REG NO:21BEC1041
NAME: P Hari Krishna Reddy

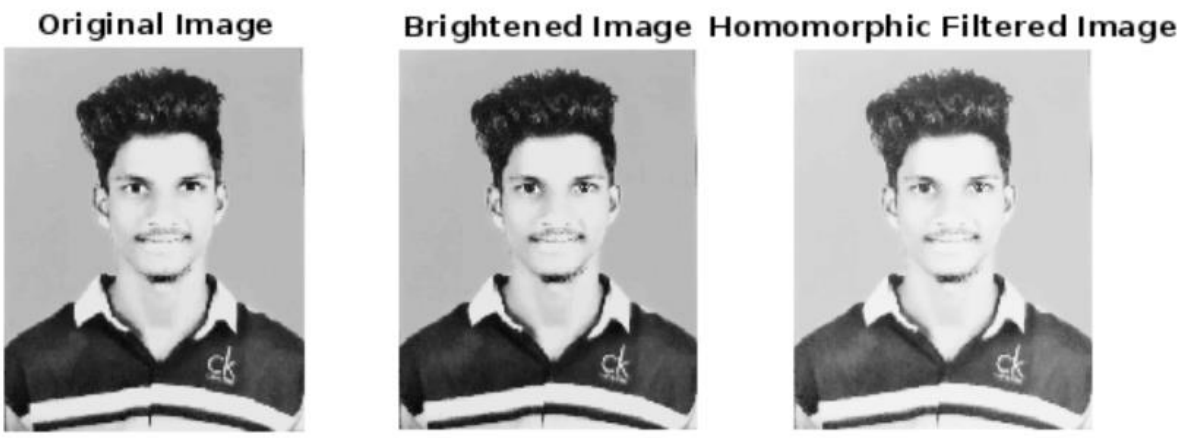
CODE:

```
clc;
close all
clear all
% Set parameters
d = 10;         % Cutoff frequency for the Gaussian filter
d2 = d^2;       % Square of the cutoff frequency

% Read and preprocess the image
f1 = double(rgb2gray(imread("/MATLAB Drive/passphoto.jpg"))); % Convert image to
grayscale and double precision
f = f1 + 100;   % Brighten the image

% Logarithmic transformation
l = log(1 + f); % Apply log transformation for dynamic range compression

% Fourier transform of the log-transformed image
z = fft2(l);    % Compute the 2D Fourier transform of the image

% Get image size
[m, n] = size(f);  % Dimensions of the image

% Initialize filter variables
b = zeros(m, n);   % Distance matrix (for Euclidean distance)
h = zeros(m, n);   % Gaussian filter matrix

% Create the Gaussian filter in frequency domain
for i = 1:m
    for j = 1:n
        % Compute the Euclidean distance from the center
        b(i, j) = sqrt((i - m / 2)^2 + (j - n / 2)^2);

        % Gaussian low-pass filter equation
        h(i, j) = exp(-b(i, j)^2 / (2 * d2));
    end
end

% Gamma values for homomorphic filtering
L = 0.5;   % Gamma low value
H = 1.5;   % Gamma high value

% Create the homomorphic filter
filter = L + (H - L) * h;

% Apply the filter in the frequency domain
```

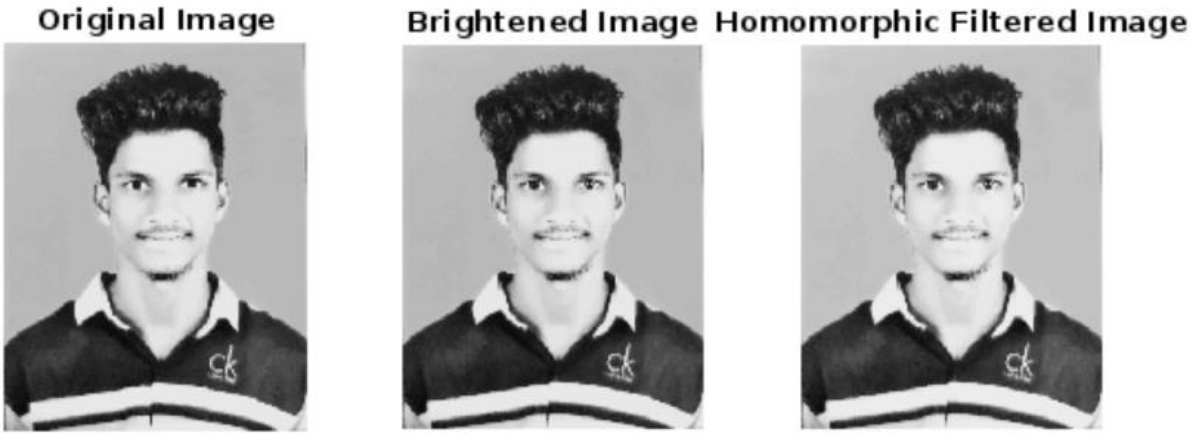
```
s = z .* filter;

% Perform inverse Fourier transform
g = abs(ifft2(s));  % Inverse FFT to get the filtered image back

% Inverse the logarithmic transformation
e = exp(g) - 1;

% Display results
figure;
subplot(1, 3, 1);   % Display the original image
imshow(f1, []);
title("Original Image");

subplot(1, 3, 2);   % Display the brightened image
imshow(f, []);
title('Brightened Image');

subplot(1, 3, 3);   % Display the homomorphic filtered image
imshow(e, []);
title('Homomorphic Filtered Image');
```

OUTPUT:

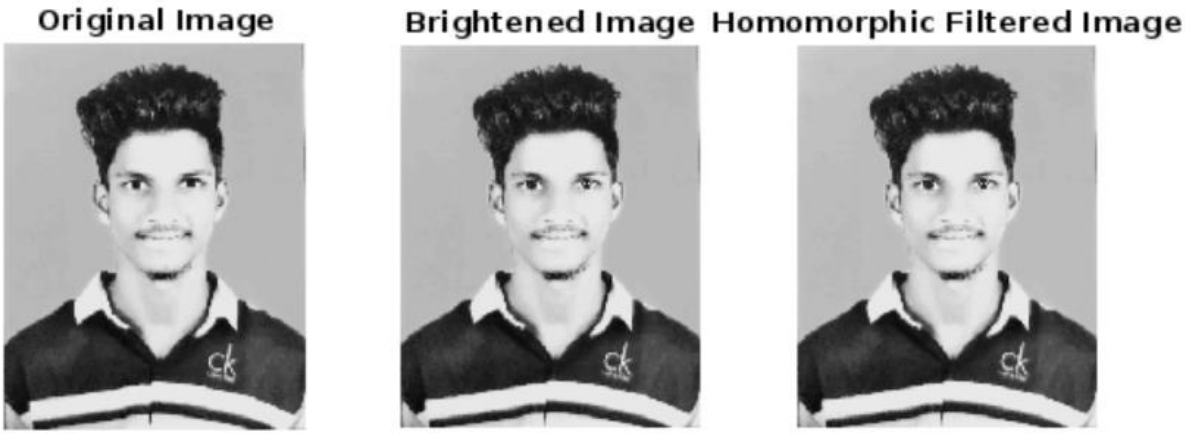
BRIGHTNESS:+0



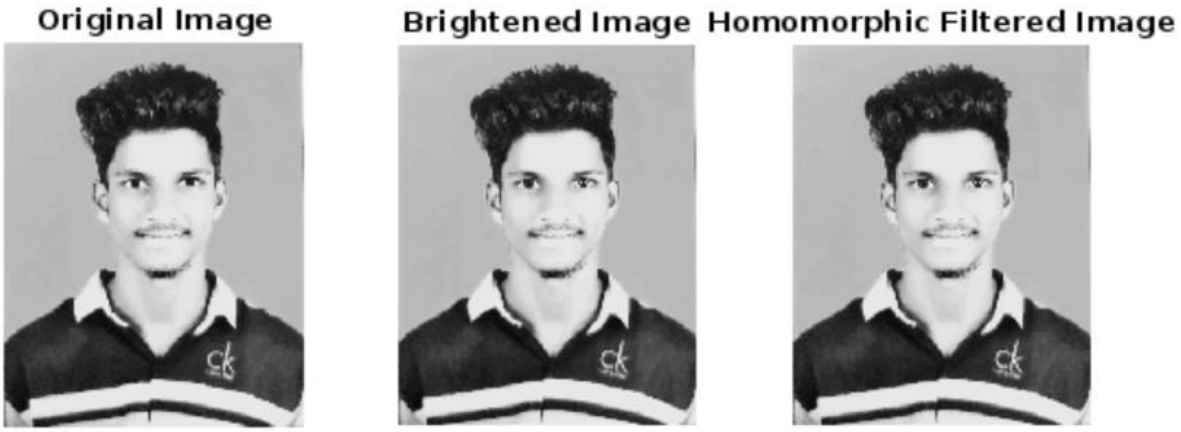
BRIGHTNESS+50:



BRIGHTNESS:+200



BRIGHTNESS:+1600



BRIGHTNESS:+30000