```
close all; % closes all figures

% read images and convert to single format
im1 = im2single(imread('./moon.jpg'));
im2 = im2single(imread('./bridge.jpg'));

% convert to grayscale
im1 = rgb2gray(im1); %low frequency
im2 = rgb2gray(im2); %high frequency

% uncomment this when debugging hybridImage so that you don't have to keep aligning
%keyboard;

%% Choose the cutoff frequencies and compute the hybrid image (you supply
%% this code)
cutoff_low = 4;
cutoff_high = 1.9;
n = 3;
[im2,im1] = hybridImage(im1, im2, cutoff_low, cutoff_high, n);

%sum the aligned images
hybrid = im2 + im1;
hybrid = imresize(hybrid, 8);

%% Crop resulting image (optional)
figure(1), hold off, imagesc(hybrid), axis image, colormap gray
disp('input crop points');
[x, y] = ginput(2);  x = round(x); y = round(y);
hybrid = hybrid(min(y):max(y), min(x):max(x), :);
figure(1), hold off, imagesc(hybrid), axis image, colormap gray

imshow(hybrid);

%% Compute and display Gaussian and Laplacian Pyramids (you need to supply this function)
function [im4, im3] = hybridImage(im1, im2, cutoff_low, cutoff_high, n)

        %Blur image 1 - low frequency
        for c = 1:n
            Z = imgaussfilt(im2,cutoff_low); %2D Gaussian
            im3 = imresize(Z, 0.5);
        end

        %Hig hfrequency of image two
        for l = 1:n
            V = locallapfilt(im1, .2, cutoff_high);
            im4 = imresize(double(V),0.5);
        end

        % use this if you want to align the two images (e.g., by the eyes) and crop
        % them to be of same size
        [im4,im3] = align_images(im3, im4);
end
```