**Experiment-1**

Q1.

clc;

clear all;

close all;

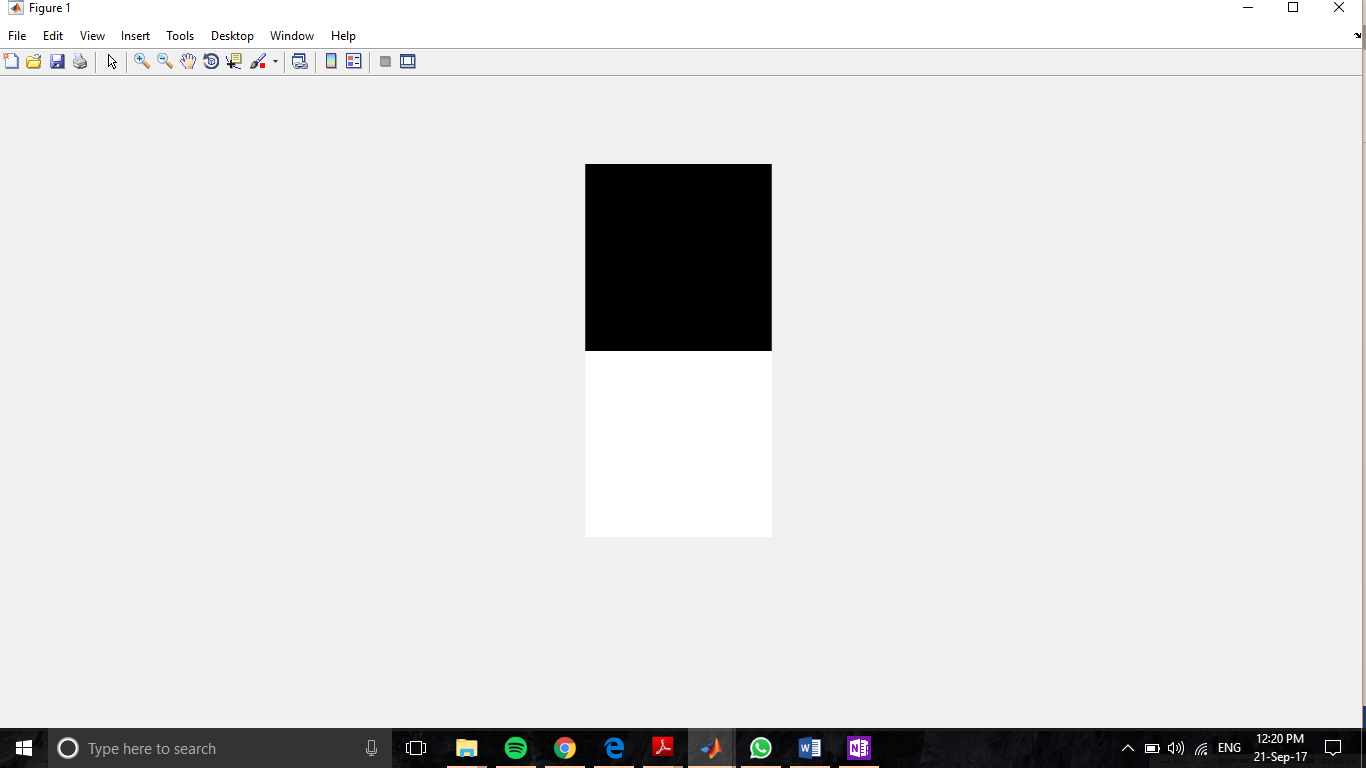
A = zeros(64, 128); % 64-by-64 matrix of black

B = ones(64,128)\*255; % 64-by-64 matrix of white

C = [A; B]; % Vertically concatenate A and B

imshow(C); %im2uint8 performs necessary scaling to recognize data as the valid image data

Simulation Results:



Q2.

clc;

clear all;

close all;

A = zeros(64, 128); % 64-by-64 matrix of black

B = ones(64,128)\*255; % 64-by-64 matrix of white

im = [A; B]; % Vertically concatenate A and B

figure

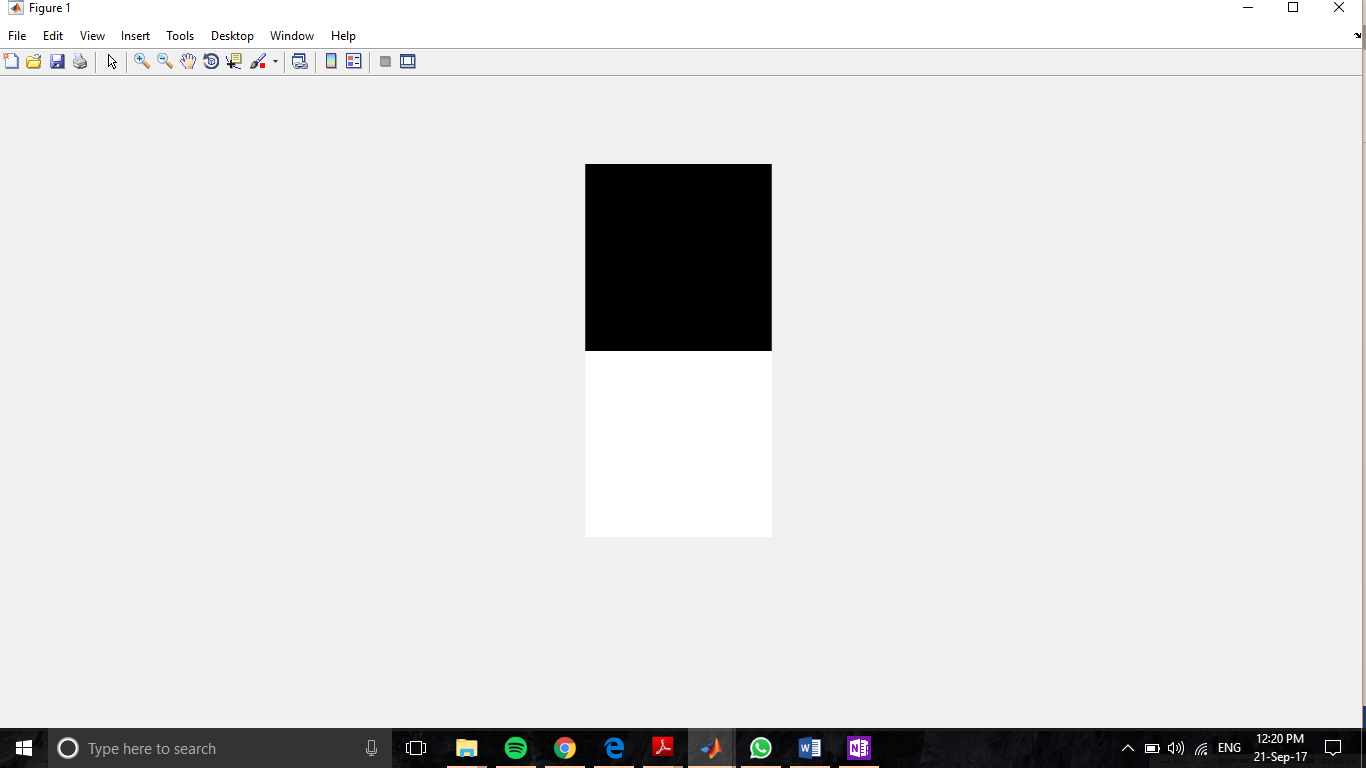
imshow(im); %im2uint8 performs necessary scaling to recognize data as the valid image data

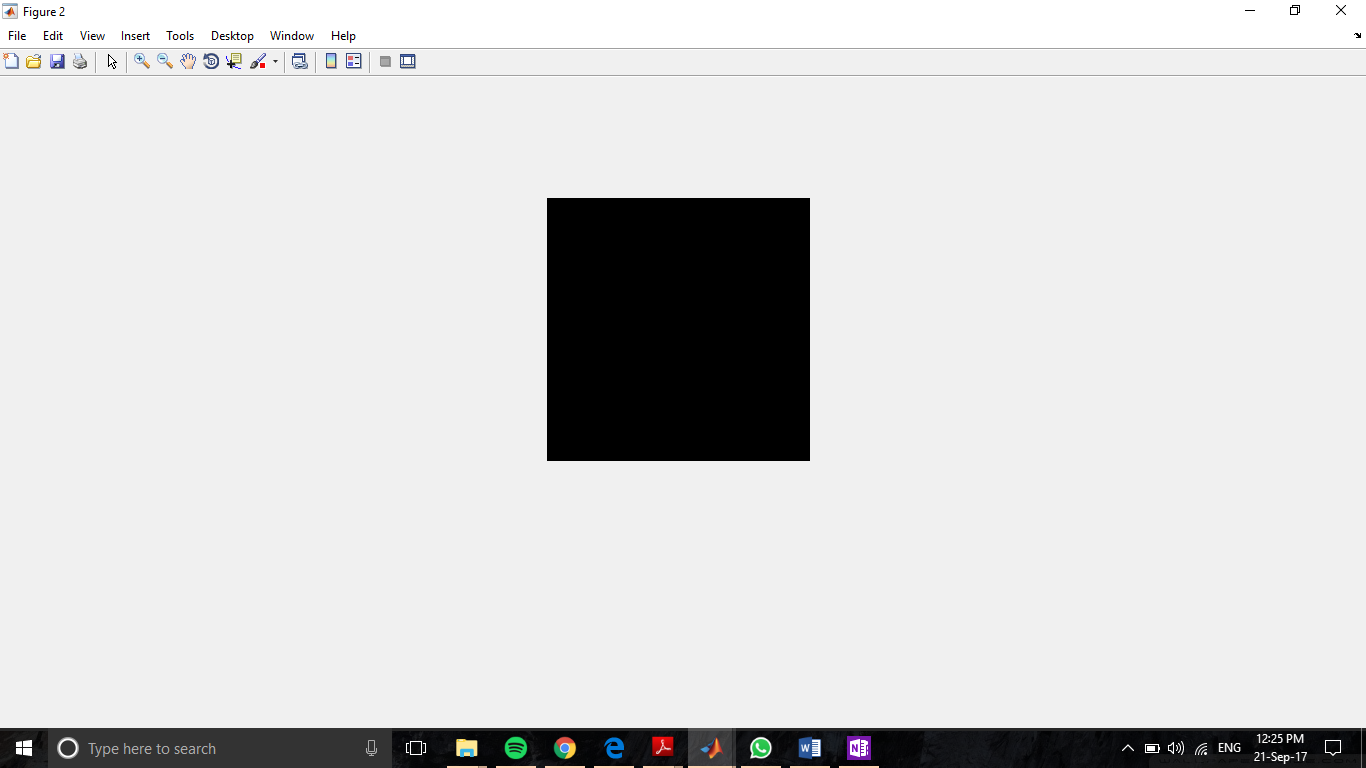
im2 = imcrop(im,[0,0,128,64]); %imcrop(image,[xmin,ymin,width,height](four element vector or a rectangular window)

figure

imshow(im2);

Simulation Results:





Q3

clc;

clear all;

close all;

A = zeros(64, 128); % 64-by-64 matrix of black

B = ones(64, 128)\*255; % 64-by-64 matrix of white

im = [A; B]; % Vertically concatenate A and B

figure

imshow(im); %im2uint8 performs necessary scaling to recognize data as the valid image data

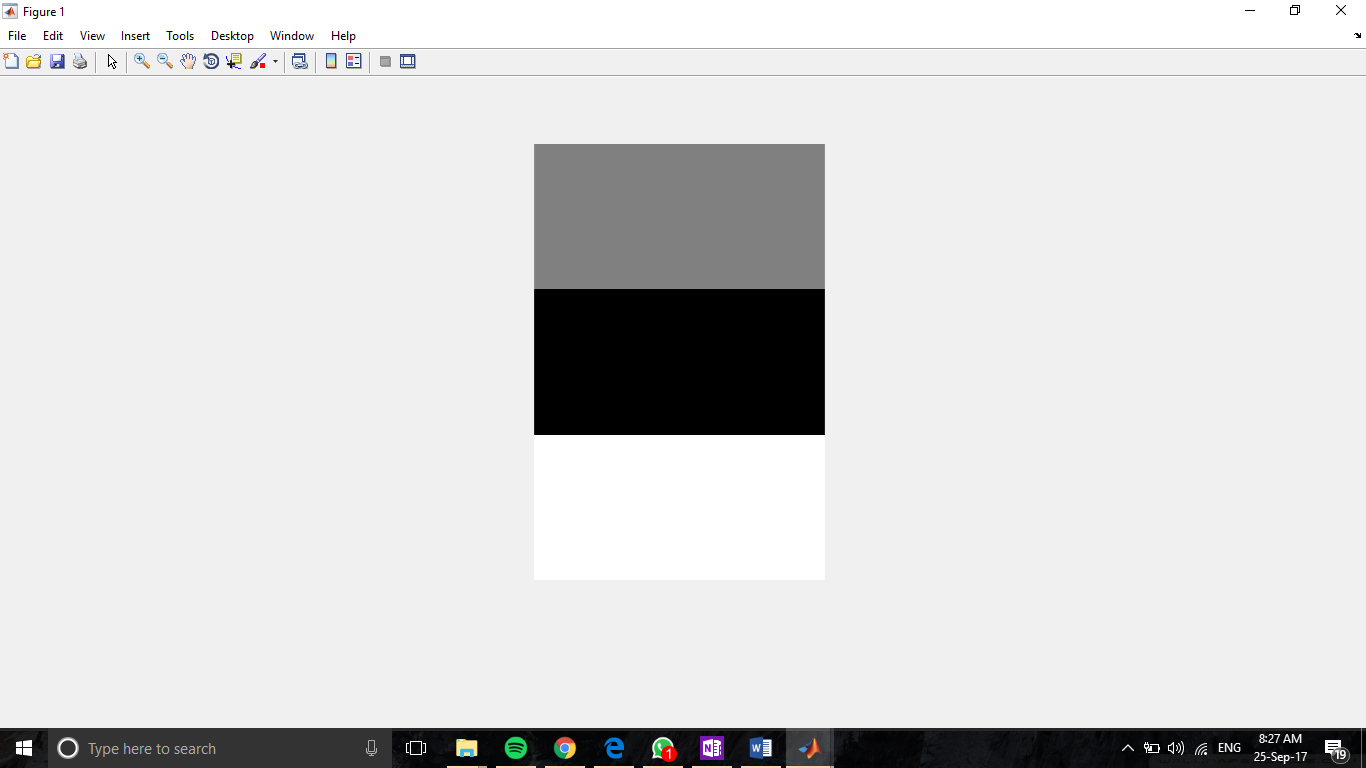
P = ones(64,128)\*128; % 64-by-128 matrix of grey

R = uint8(P); %change from double to uint 8

im2 = [R;im];

figure

imshow(im2);



Q4.

clc;

clear all;

close all;

im=logical(zeros(256,256));

im(128,128)=1; %make the centre pixel white

figure

imshow(im);

