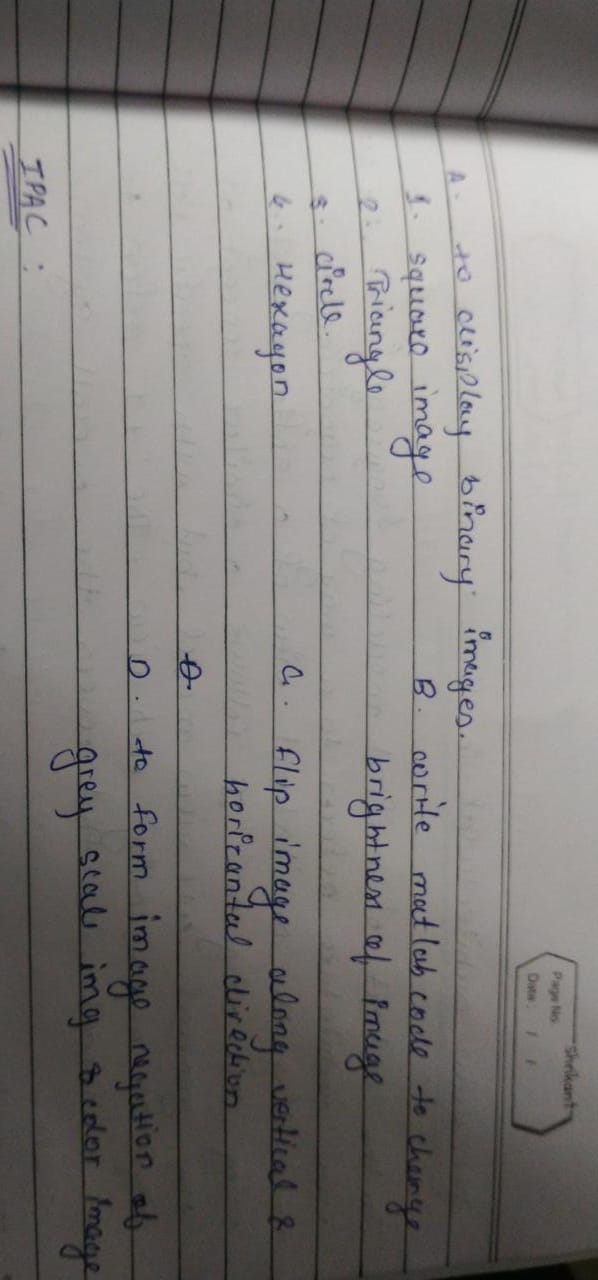
Experiment Number: 1

NAME: **Kartik Banshi Katkar** ROLLNO: 36

CLASS: TY IT A BATCH: 1

DATE OF PERFORMANCE: 06/07/2003

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Question:**

Answer:

A.  
1. Square:

A = zeros (512,512);

for i =150:350

for j=150:350

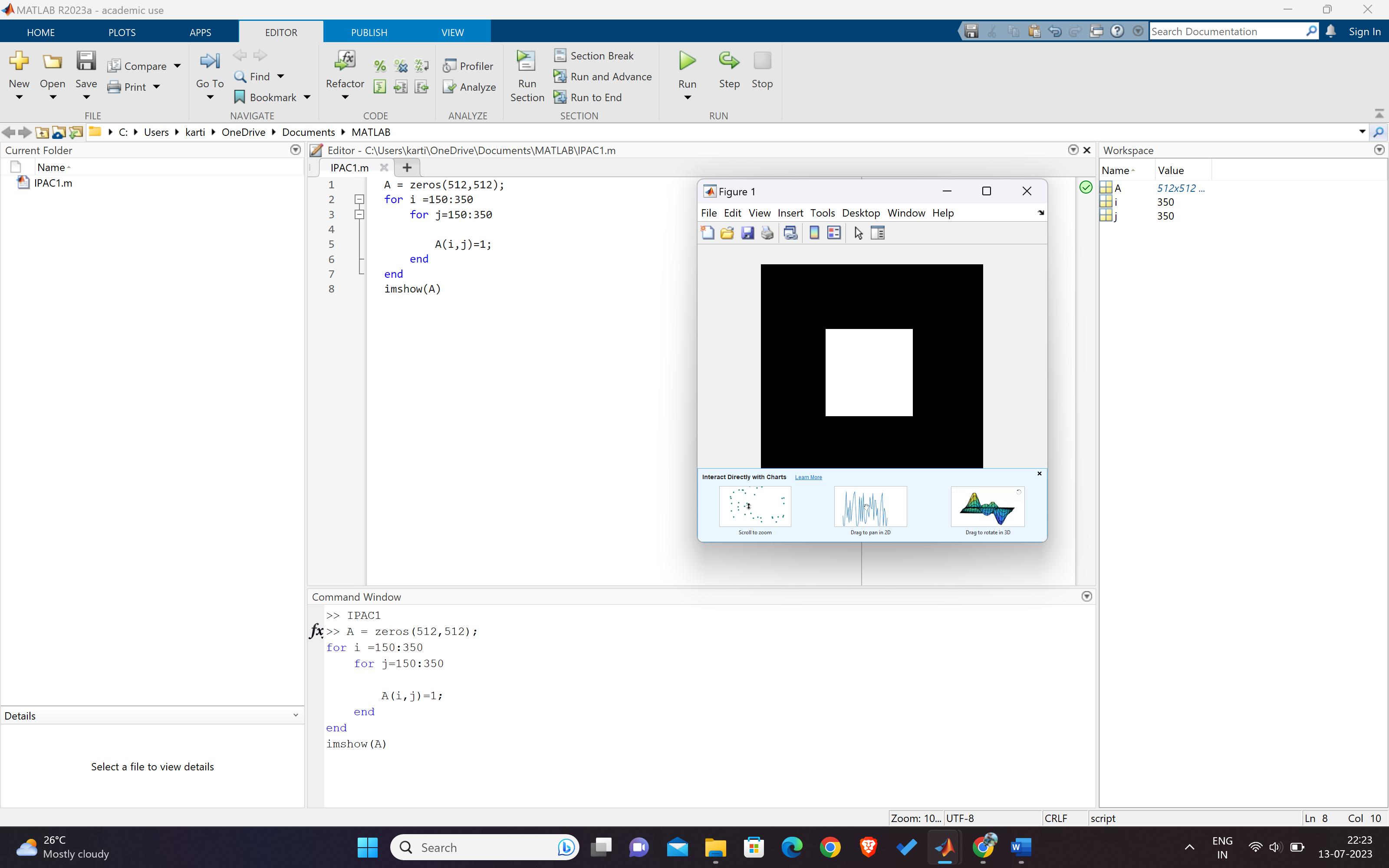
A(i,j)=1;

end

end

imshow(A)

Output:



2. Triangle

A = zeros(512,512);

for i =150:350

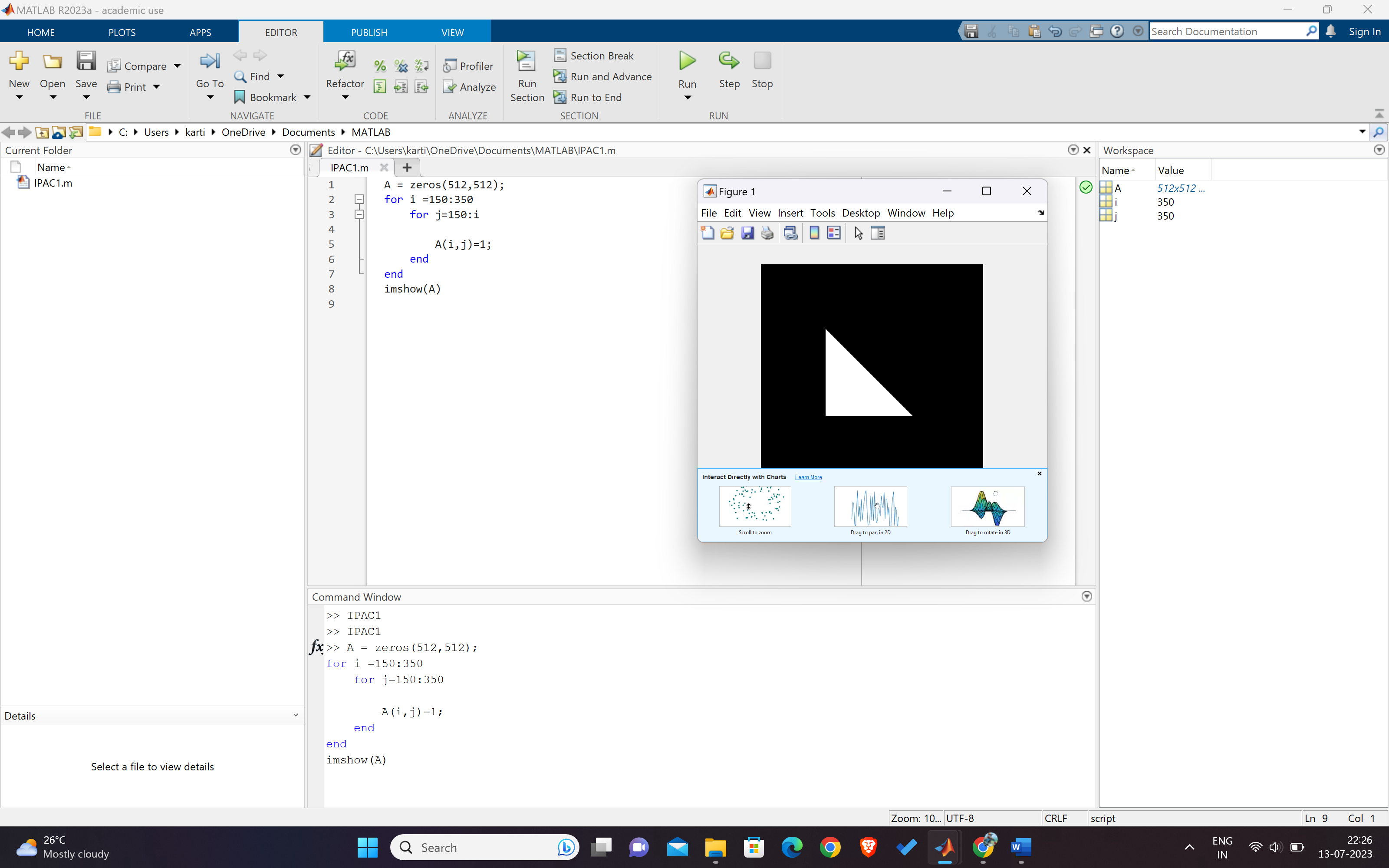
for j=150:i

A(i,j)=1;

end

end

imshow(A)  
  
  
Output:



3. Circle  
  
A = zeros(512, 512); % Create a 512x512 matrix filled with zeros

% Center and radius of the circle

center\_x = 256; % x-coordinate of the center

center\_y = 256; % y-coordinate of the center

radius = 200; % radius of the circle

% Loop through each pixel in the matrix

for i = 1:size(A, 1)

for j = 1:size(A, 2)

% Calculate the distance between the current pixel and the center of the circle

distance = sqrt((i - center\_x)^2 + (j - center\_y)^2);

% If the distance is less than or equal to the radius, set the pixel value to 1 (white)

if distance <= radius

A(i, j) = 1;

end

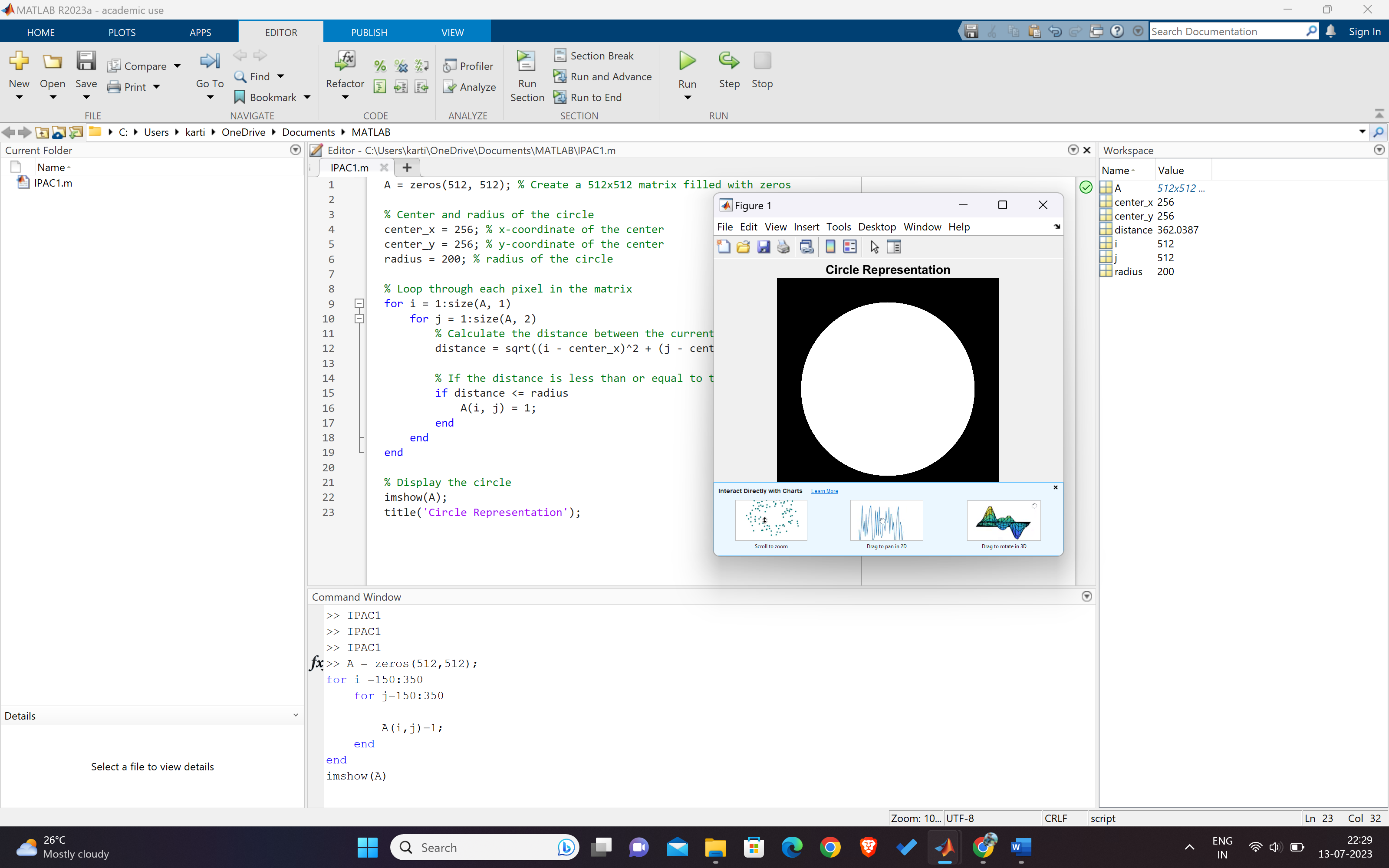
end

end

% Display the circle

imshow(A);

title('Circle Representation');



4. Hexagon  
  
%hexagon

scale = 5;

N\_sides = 6;

t=(1/(N\_sides\*2):1/N\_sides:1)'\*2\*pi;

x=sin(t);

y=cos(t);

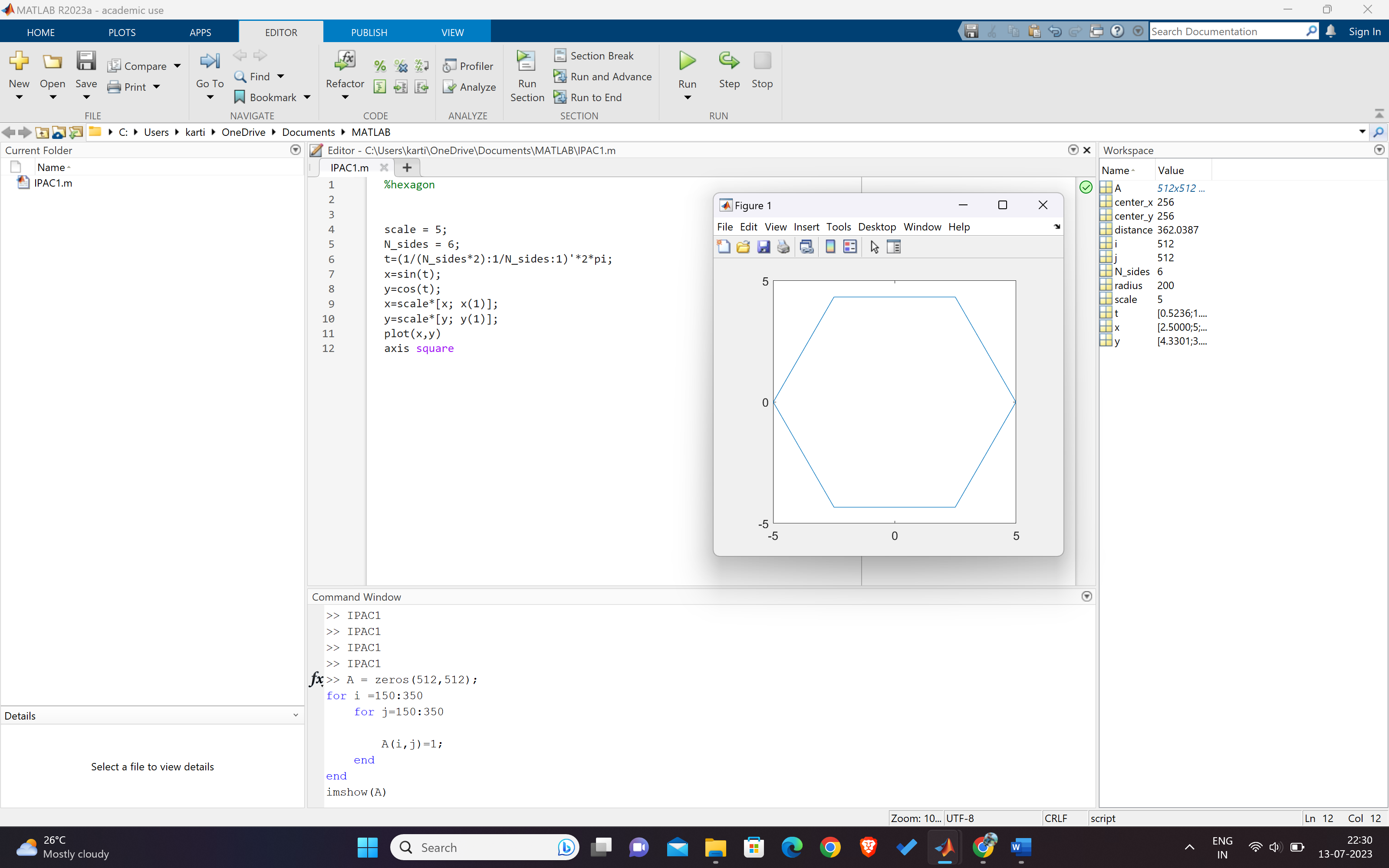
x=scale\*[x; x(1)];

y=scale\*[y; y(1)];

plot(x,y)

axis square

Output:



B. Brightness of Image

a = imread("image.jpeg");

a1 = a + 150;

a2 = a - 150;

subplot(1, 2, 1);

imshow(a1);

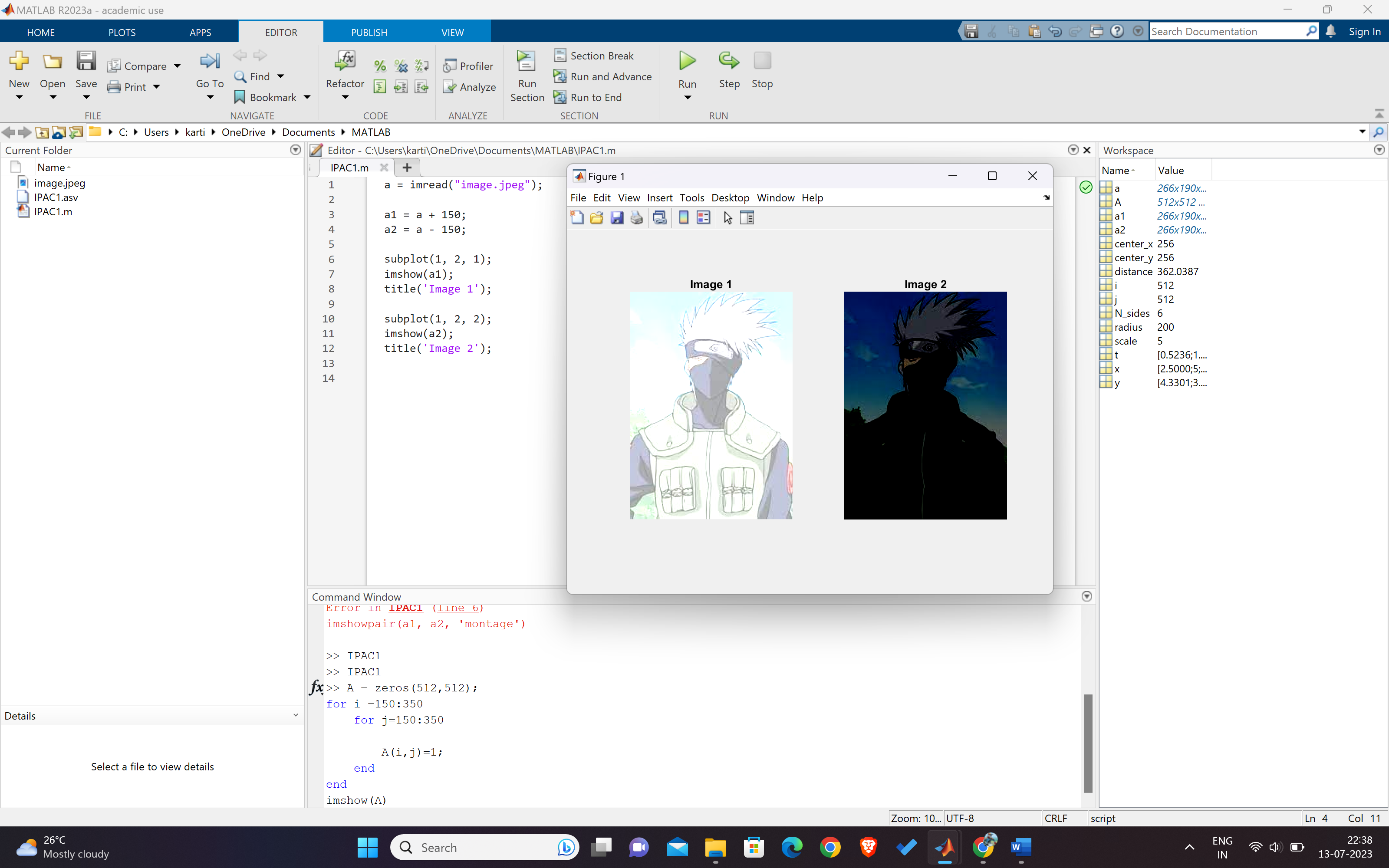
title('Image 1');

subplot(1, 2, 2);

imshow(a2);

title('Image 2');

Output:



C. Flipped Image along vertical and horizontal direction  
  
%function image\_operations(~)

% Load the image

image = imread('image.jpeg');

% Flip image along the horizontal direction

flipHorizontal = flip(image, 2);

% Flip image along the vertical direction

flipVertical = flip(image, 1);

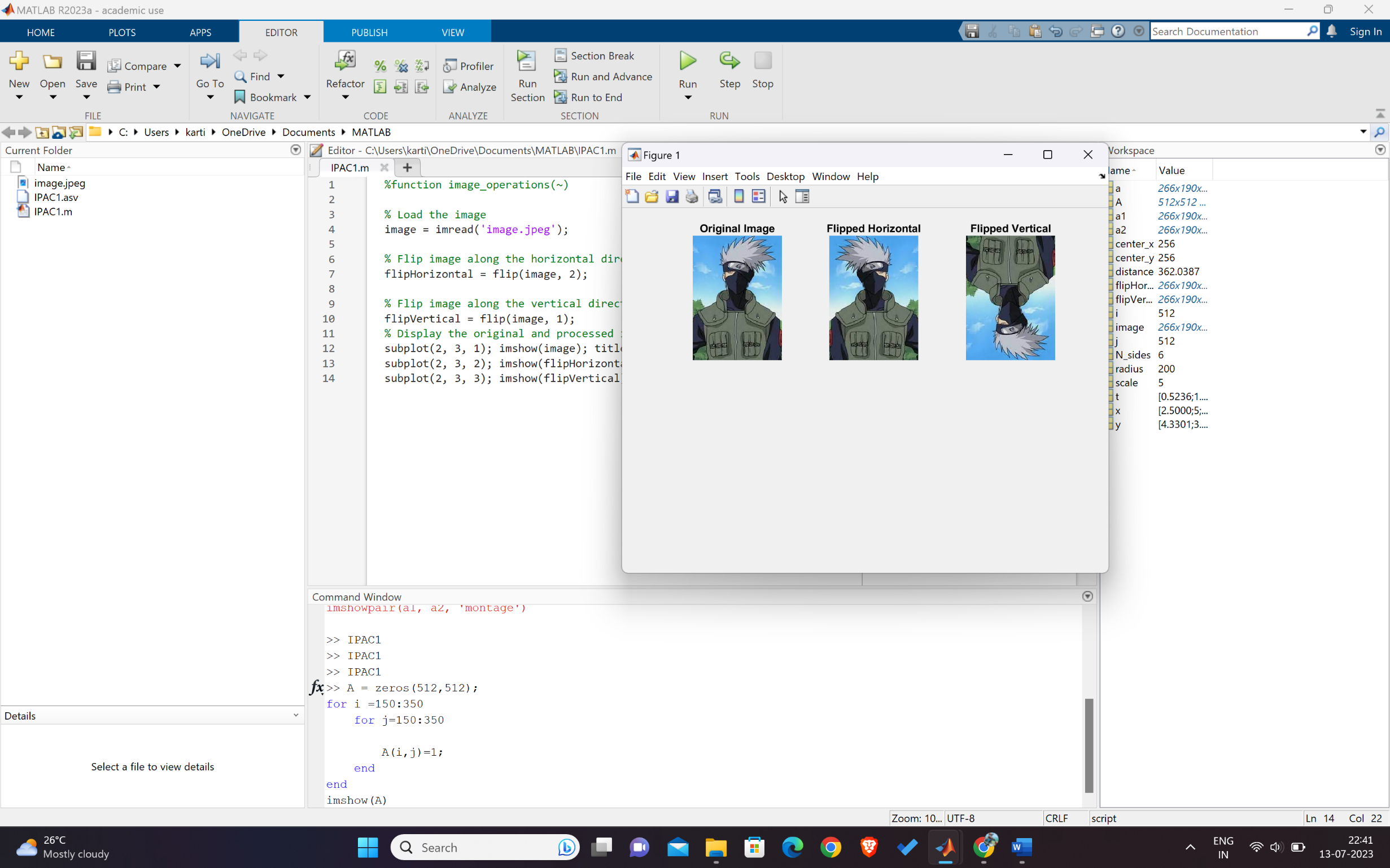
% Display the original and processed images

subplot(2, 3, 1); imshow(image); title('Original Image');

subplot(2, 3, 2); imshow(flipHorizontal); title('Flipped Horizontal');

subplot(2, 3, 3); imshow(flipVertical); title('Flipped Vertical');

Output:

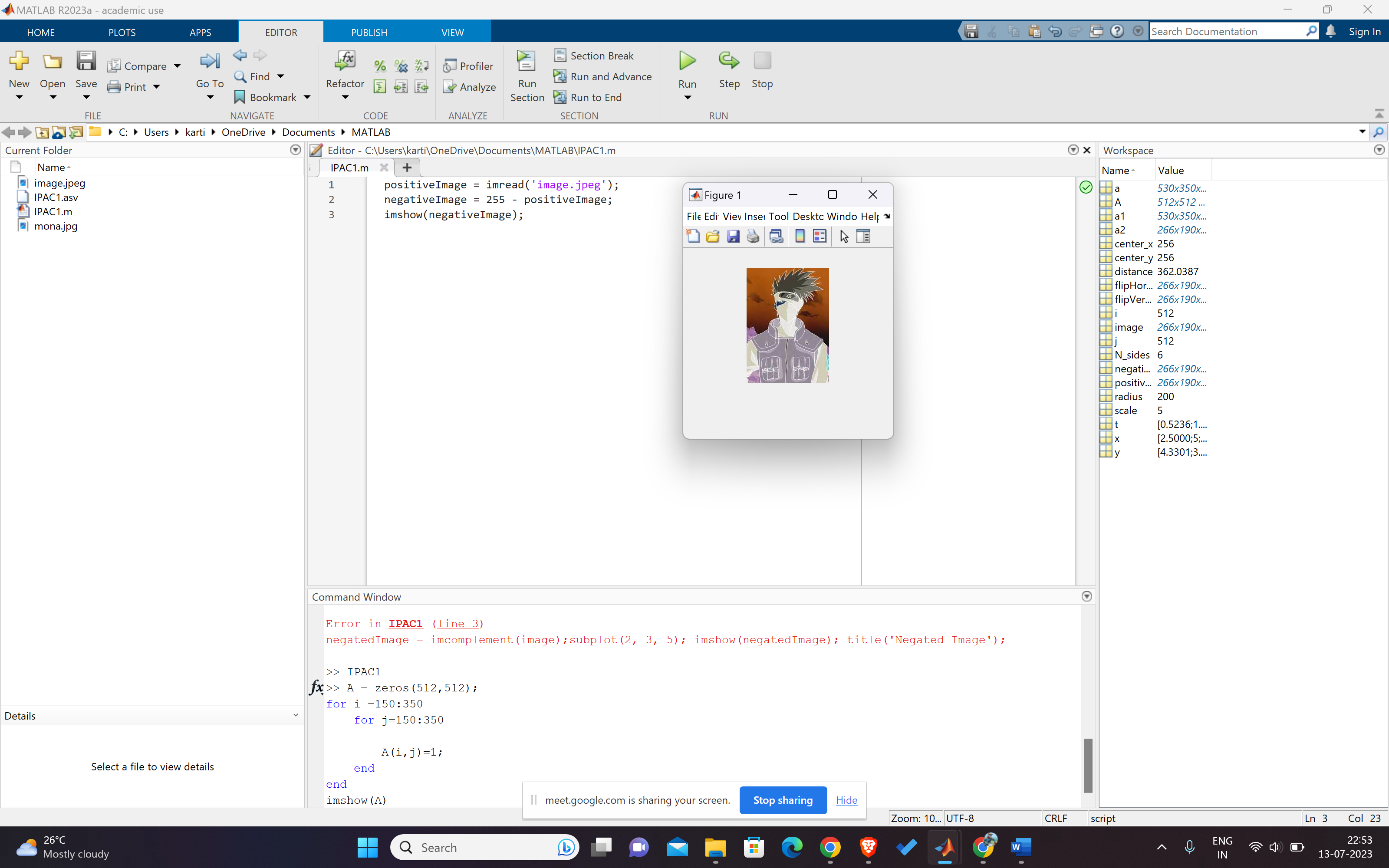


D. To form image negation of grayscale image to color image

positiveImage = imread('image.jpeg');

negativeImage = 255 - positiveImage;

imshow(negativeImage);



**E. Change contrast of given image:**

% Read the image

originalImage = imread('image.jpeg');

% Convert the image to grayscale if necessary

if size(originalImage, 3) == 3

grayImage = rgb2gray(originalImage);

else

grayImage = originalImage;

end

% Scale the intensity values to the range [0, 1]

normalizedImage = mat2gray(grayImage);

% Adjust the contrast

contrastAdjusted = imadjust(normalizedImage, [0.2 0.8], [0 1]);

% Display the original and contrast-adjusted images

figure;

subplot(1, 2, 1);

imshow(normalizedImage);

title('Original Image');

subplot(1, 2, 2);

imshow(contrastAdjusted);

title('Contrast Adjusted Image');

