```
% read the image
>> image = imread('image_RGB.jpg');

>> whos
Name          Size                Bytes  Class    Attributes

image       260x390x3            304200  uint8

% show the image readed
% Output Photo_1
>> imshow(image);

%  See the content of a selected part of the image matrix
>> image(11:15,70:72,:);

% Change the color of the above image rectangular region to green
% (i.e., R=0, G=255, and B=0) and display the modified image
>> image(11:15,70:72,1)=0; image(11:15,70:72,2)=255;image(11:15,70:72,3)=0;
% Output Photo_2
>>imshow(image);

% read a gray image
>> gray_image = imread('einstein.jpg');
% Output Photo_3
>> imshow(gray_image);
% Image histogram
% Output Photo_4
>> imhist(gray_image);

% improve the image
>> improve_image = histeq(gray_image);
% Image histogram
% Output Photo_6
>> imhist(improve_image);
% Output Photo_5
>> imshow(improve_image);

>> rgbimg = imread("original_3.jpg");
% Output Photo_7
>> imshow(rgbimg);

% let the colored image a gray image
>> grayimg = rgb2gray(rgbimg);
% Output Photo_8
>> imshow(grayimg);

% minimize the size of the photo
>> resized_bigger = imresize(grayimg,1.5);
>> resized_smaller = imresize(grayimg,0.5);
% Output Photo_9
>> subplot(121); imshow(resized_smaller);
>> subplot(122);imshow(resized_bigger);

% Flip the image horizontally
```

```
>> flipped_horiz = fliplr(rgbimg);
% Flip the image vertically
>> flipped_vert = flipud(rgbimg);
% Rotate the image 90 degrees counterclockwise
>> rotated_90 = rot90(rgbimg, 1);
% Rotate the image 180 degrees
>> rotated_180 = rot90(rgbimg, 2);
% Rotate the image 270 degrees
>> rotated_270 = rot90(rgbimg, 3);

% Output Photo_10
>> figure;
>> subplot(231), imshow(rgbimg), title('Original');
>> subplot(232), imshow(flipped_horiz), title('Flipped Horizontally');
>> subplot(233), imshow(flipped_vert), title('Flipped Vertically');
>> subplot(234), imshow(rotated_90), title('Rotated 90°');
>> subplot(235), imshow(rotated_180), title('Rotated 180°');
>> subplot(236), imshow(rotated_270), title('Rotated 270°');

% Read the grayscale image
>> y_Gray = imread('einstein.jpg');
% Output Photo_11
>> figure
>> subplot(121)
% Display the original grayscale image
>> imshow(y_Gray)
>> subplot(122)
% Display the negative image
>> imshow(255 - y_Gray)
% Output Photo_12
%  Display 6 Copies of the Grayscale Image with Increasing Brightness
>> subplot(231), imshow(y_Gray)
>> subplot(232), imshow(min(y_Gray + 20, 255))
>> subplot(233), imshow(min(y_Gray + 40, 255))
>> subplot(234), imshow(min(y_Gray + 60, 255))
>> subplot(235), imshow(min(y_Gray + 80, 255))
>> subplot(236), imshow(min(y_Gray + 100, 255))

% Read and Analyze an Indexed Image
% Read indexed image and colormap
% t_Indexed stores the image matrix (integer values representing color indices).
% cmap is a colormap matrix (each row is an RGB triplet).
>> [t_Indexed, cmap] = imread('original_4.png');
>> whos

 cmap                  0x0                    0  double
  t_Indexed          256x256              65536  uint8
  y_Gray             182x186x3           101556  uint8

  % Get the color index at row 50, column 66
  >> t_Indexed(50,66)

ans =
```

```
  uint8

   176

  % Output Photo_13
  >> figure,
  >> subplot (131);
  % Displays the indexed image incorrectly (grayscale appearance)
  >> imshow(t_Indexed);

  >> subplot (132);
  % Scales intensity values for better visibility
  >> imshow(t_Indexed, []);

  >> subplot (133);
  % Displays the image with the correct colormap
  >> imshow(t_Indexed, cmap);


% imshow(t_Indexed):
% displays the index values as grayscale intensities.

% imshow(t_Indexed, []):
% normalizes the index values between 0 and 255.

% imshow(t_Indexed, cmap):
% correctly maps index values to RGB colors using the colormap.
```