Name: SUN RUI
Student ID: 18083229g

Task1, Gray Image Processing:

1. Basic image read, write and display, the below is main code snippets, annotations show the functions of each line code:

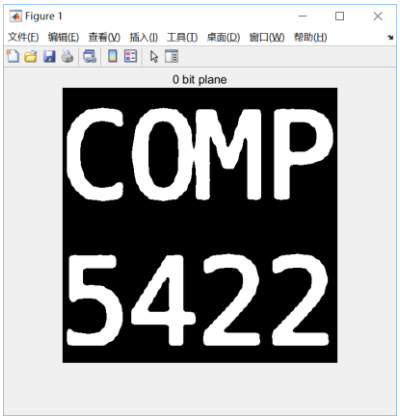
```
grayLena = imread('lena.bmp');    % 1.1 read 'lena.bmp' by imread.
grayLena_half = grayLena*0.5;     % 1.2 reduce all pixel values by half.
figure(9);
imshow(grayLena_half),title("half pixel value");  % 1.3display it by imshow.
imwrite(grayLena_half, 'lena2.bmp')    % 1.4 save to 'lena2.bmp' by imwrite.
```

1.5 What is the visual difference between 'lena.bmp' and 'lena2.bmp':
'lena2.bmp' is darker than 'lena.bmp' in human vision.

2. Bit-plane, the below is main code snippets, annotations show the functions of each line code:

```
for index = 1:8
    figure(index);    % named the figure
    eachPlaneLena=bitget(grayLena,index);    %  get bit at specified position
    imshow(logical(eachPlaneLena)),title((index-1) + " bit plane");    %
show and titled by bit-plane every picture
    if index == 1
        imwrite(logical(eachPlaneLena), "message.bmp")    % save picture of 0
bit-plane and named message.bmp
```

Through traveling and showing every bit-plane of the image "lena.bmp", we can find the picture of 0 bit-plane has the secret message which is "COMP 5422":



Question: Can you perceive this message directly from the image without any image processing? Why?
Answer: I cannot perceive this message directly, because lower order planes add fine (often imperceptible) details to the image, higher order bit planes of an image carry a significant amount of visually relevant details, and the message is encoded by 0 bit-plane of the image, 0 bit-plane is lowest bit plane of the image.

3. Lossy Compression by Discarding Lower Bits, the below is main code snippets, annotations show the functions of each line code:

```
plane8 = bitget(grayLena,8);    % 3.1 Extract 8 bit-planes from 'lena.bmp'
plane7 = bitget(grayLena,7);    % 3.1 Extract 7 bit-planes from 'lena.bmp'
plane6 = bitget(grayLena,6);    % 3.1 Extract 6 bit-planes from 'lena.bmp'
plane5 = bitget(grayLena,5);    % 3.1 Extract 5 bit-planes from 'lena.bmp'

save('compressed.mat', 'plane5', 'plane6', 'plane7', 'plane8');    %3.2 Save the
four bit-planes to 'compressed.mat' by 'save'
S=load('compressed.mat');    % 3.4 load the four bit-planes from 'compressed.mat'
by load('compressed.mat')
concat = S.plane5*2^4 + S.plane6*2^5 + S.plane7*2^6 + S.plane8*2^7;    % 3.5 Reconst
ruct the image from the highest four bit-planes.

imwrite(concat, 'compressed.bmp')    % 3.6 Save it to 'compressed.bmp'.
```

3.2 How large is the image file 'lena.bmp'?
How large is the file 'compressed.mat'?
The size of 'lena.bmp' is 258KB.
The size of 'compressed.mat' is 91KB.

3.6 Could you see the difference after lossy compression? What is the difference?
Yes, I can. The image “compressed.bmp” has many noisy points, and details of “compressed.bmp” are not very good, in a word, it is less clear than “lena.bmp”.

Task2, Color Image Processing:

1. Complement Image, the below is main code snippets, annotations show the functions of each line code:

```
flowers = imread('Flowers.jpg');
invFlowers=255-flowers;    % 1.1 compute complement image of Flowers.
imwrite(invFlowers, 'negativeFlowers.jpg')    % 1.2 save it to 'negativeFlowers.jpg'
```

2. Modify image saturation, the below is main code snippets, annotations show the functions of each line code:

```
hsvFlowers = rgb2hsv(flowers);    % 2.1 convert flowers to HSV format
hsvFlowers(:,:,2) = hsvFlowers(:,:,2)*0.5;    % 2.2 reduce Saturation of all pixels
to half
rgbFlowers = hsv2rgb(hsvFlowers);    % 2.3 convert back to RGB format.
imwrite(rgbFlowers, 'LessColorfulFlowers.jpg')    % 2.4 save it to 'LessColorfulFlowe
rs.jpg'
```

3. Image Segmentation, the below is main code snippets, annotations show the functions of each line code:

```
redChannel = flowers(:,:,1);    % extract red channel
greenChannel = flowers(:,:,2);  % extract green channel
blueChannel = flowers(:,:,3);   % extract blue channel
BWR=redChannel>165;    % set red channel booling values by low thresold value 165
BWG=greenChannel>115;  % set red channel booling values by low thresold value 115
BWB=blueChannel<70;    % set red channel booling values by top thresold value 70
combine_bool =BWR&BWG&BWB;    % use '&' to combine three channels booling values
flowers(:,:,1) = flowers(:,:,1).*uint8(combine_bool);    % convert combine_bool to
unit8 and filter red channel
flowers(:,:,2) = flowers(:,:,2).*uint8(combine_bool);    % convert combine_bool to
unit8 and filter green channel
flowers(:,:,3) = flowers(:,:,3).*uint8(combine_bool);    % convert combine_bool to
unit8 and filter blue channel
figure(1)
imshow(flowers);    % show the new image
imwrite(flowers, 'segFlowers.jpg')    % save image
```