MATLAB Assignment 2

Quetion 1

The portable graymap format (PGM) grayscale images used in this assignment were retrieved from http://sun.aei.polsl.pl/~rstaros/mednat/index.html. This is a dataset of medical grayscale images for evaluating image compression algorithms. The set was prepared as a part of the research project Nr 4 T11C 032 24, which was fully supported by the Grant of the Polish National Research Committee (KBN).

This function performs multiplicative brightness corretion on a monochrome image. It takes as arguments: a monochrome image, a number between 0 and 100 and a parameter indicating whether the correction is intended to brighten or darken the image

```
function [y] = brightnessFunction(I, amount, b)
y = (1 + b * (amount/100)) * I;
```

Published with MATLAB® R2015b

Quetion 2

MATLAB script that reads an image, performs brightness correction using the brightnessFunction, and displays a windows with theimage and its histogram, before and after the brightness

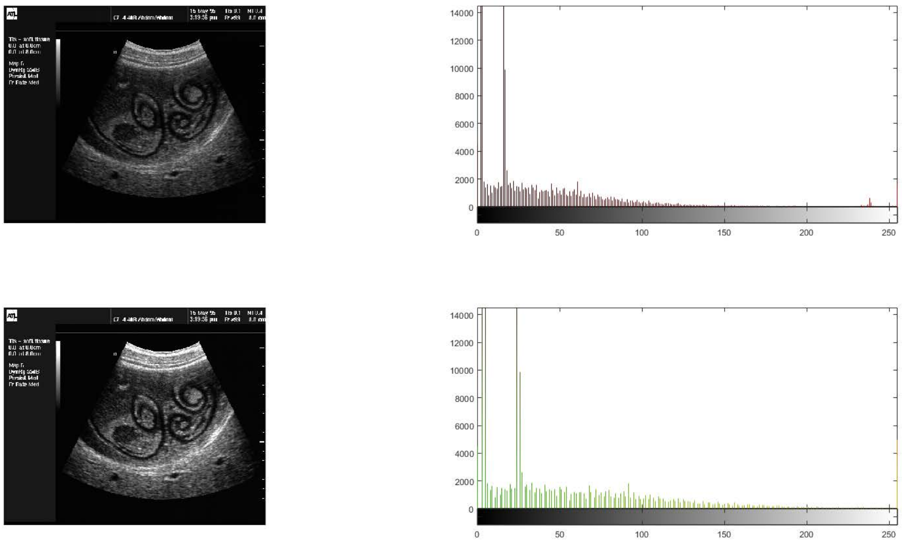
```
close all; clear; clc;
I = imread('us_3403.pgm');

brightInput = input('Enter the letter b to brighten or d to darken):
 ', 's');
if (brightInput == 'b')
    brt = 1;
elseif (brightInput == 'd');
    brt = -1;
end;

amount = input('Enter a number from 0 to 100 for the amount of
 brightness correction: ');
J = brightnessFunction(I, amount, brt);

subplot(2,2,1), imshow(I)
subplot(2,2,2), imhist(I)
subplot(2,2,3), imshow(J)
subplot(2,2,4), imhist(J)
```

```
Enter the letter b to brighten or d to darken): b
Enter a number from 0 to 100 for the amount of brightness correction:
50
```



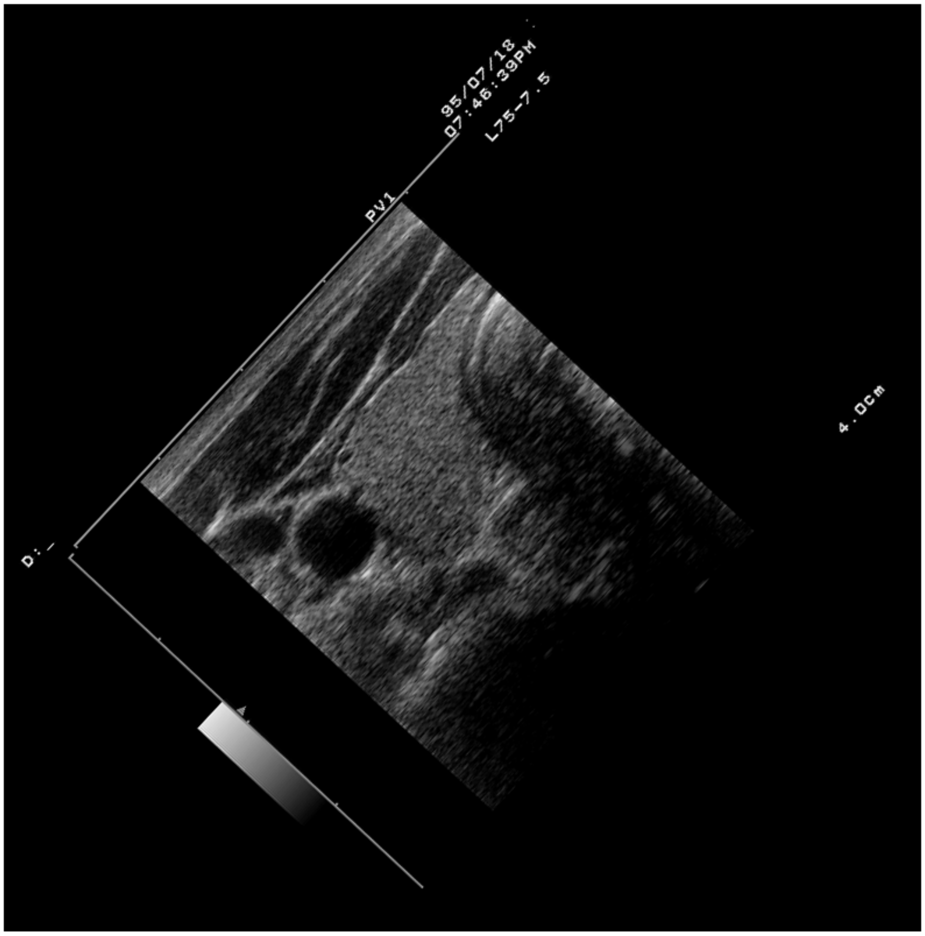
Published with MATLAB® R2015b

Quetion 3

*Uses imrotate to write a MATLAB script that reads an image, rotates it by an arbitrary angle using bilinear interpolation. The script di8splays the amount of rotation in degrees, the name of the interpolation method, and associated processing time using the tic and toc functions.

```
close all; clear; clc;
tic
fName = 'us_19773.pgm';
I = imread(fName);
deg = 47;
method = 'bilinear';
I_rot = imrotate(I,deg, method);
imshow(I_rot);
disp(['Output: ', fName, ', ', int2str(deg),', ', method])
t = toc;
elapsed = sprintf(['The elapsed time for rotation of the image '
 fName ', at ' int2str(deg) ' degrees, using ' method, ' interpolation
 is ', num2str(t), ' seconds']);
disp(elapsed)
```

```
Output: us_19773.pgm, 47, bilinear
The elapsed time for rotation of the image us_19773.pgm, at 47
 degrees, using bilinear interpolation is 2.0058 seconds
```



Published with MATLAB® R2015b

Quetion 4 function

A MATLAB function that performs a piecewise linear brightness and contrast adjustment on a monochrome image. The function takes aa arguments: a monochrome image, the c coefficient (slope), and the b coefficient(offset).

```
function [y] = piecewiseFunction(I, c, b)
temp = immultiply(I,c);
y = imadd(temp, b);
```

Published with MATLAB® R2015b

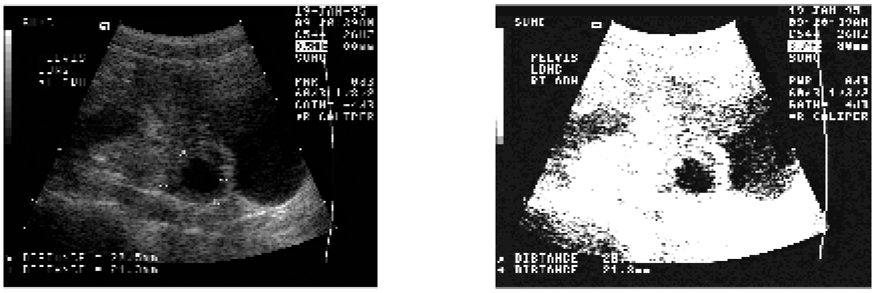
Quetion 4

Implements the piecewiseFunction in Question 4, to perform a brightness and contrast on a monochrom image

```
close all; clear; clc;
I = imread('us_27743.pgm');
c = 10;
b = 15;
```

piecewiseFuction call

```
J = piecewiseFunction(I, c, b);
subplot(1,2,1), imshow(I)
subplot(1,2,2), imshow(J)
```



Published with MATLAB® R2015b

Quetion 5

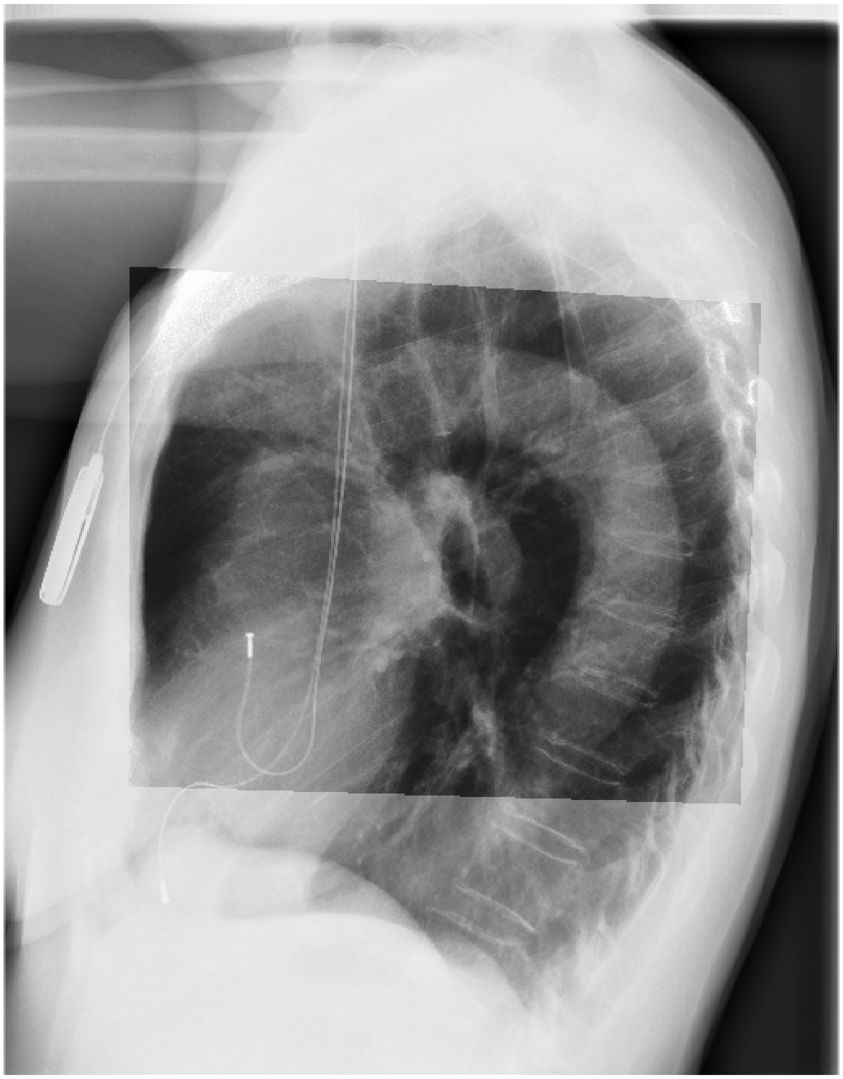
A MATLAB script that implements region-based histogram equalization on a monochrome image. The script allows the users to specify an image, interactively select a region of interest to which the histogram equalization operation will be applied, and disppays the results of the operation.

```
close all; clear; clc;
[file, canceled] = imgetfile();
if canceled
    return
end
info = imfinfo(file);
if not(strcmp(info.ColorType, 'grayscale' ))
    fprintf('input image must be grayscale!\n');
    return
end

img = imread(file);
mask = roipoly(img);

img2 = histeq(img, 256);
img2 = immultiply(not(mask), img) + immultiply(mask, img2);
imshow(img2);
```

```
Warning: Image is too big to fit on screen; displaying at 33%
Warning: Image is too big to fit on screen; displaying at 33%
```



Published with MATLAB® R2015b

Quetion 6 Function

A MATLAB function that creates an 8-bit random image with a Gaussian distribution of pixel values. The function takes 4 parameters; height width, mean value,and standard deviation

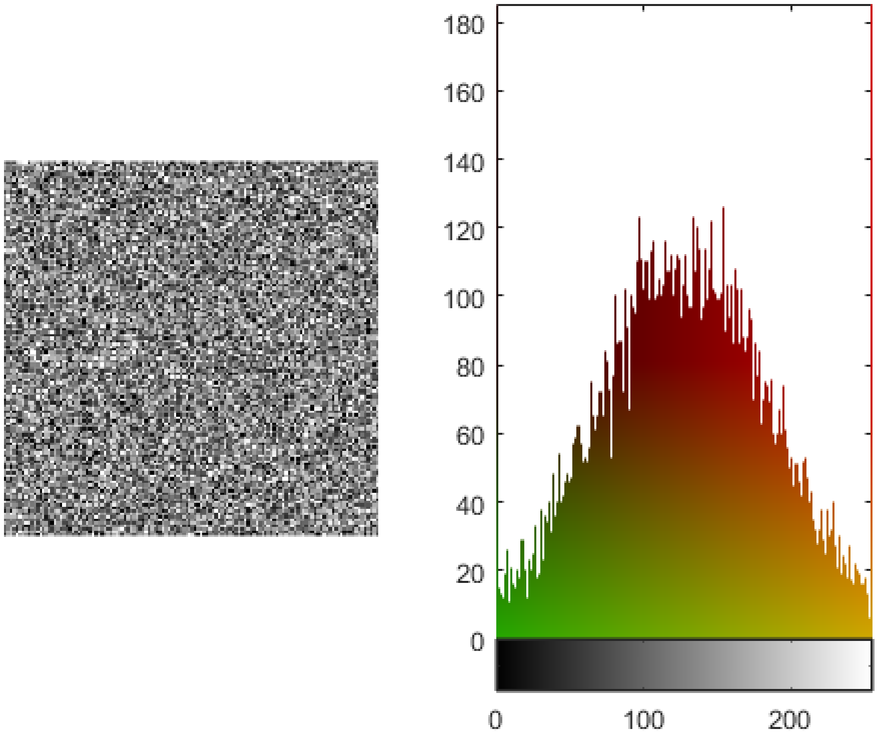
```
function [y] = gaussianFunction(h, w, mean, std)
y = mean + std * randn(h, w);
```

Published with MATLAB® R2015b

Quetion 7

A MATLAB script that implements the gaussianFunction to create a 128 X 128 random image with Gaussian distribution of pixel values, with mean value of 128 and standard deviation of 60. The image's histogram is also displayed to confirm its Gauss Curve.

```
h = 128;
w = 128;
mean = 128;
std = 60;
J = gaussianFunction(h, w, mean, std);
J = uint8(J);
subplot(1,2,1), imshow(J)
subplot(1,2,2), imhist(J)
```



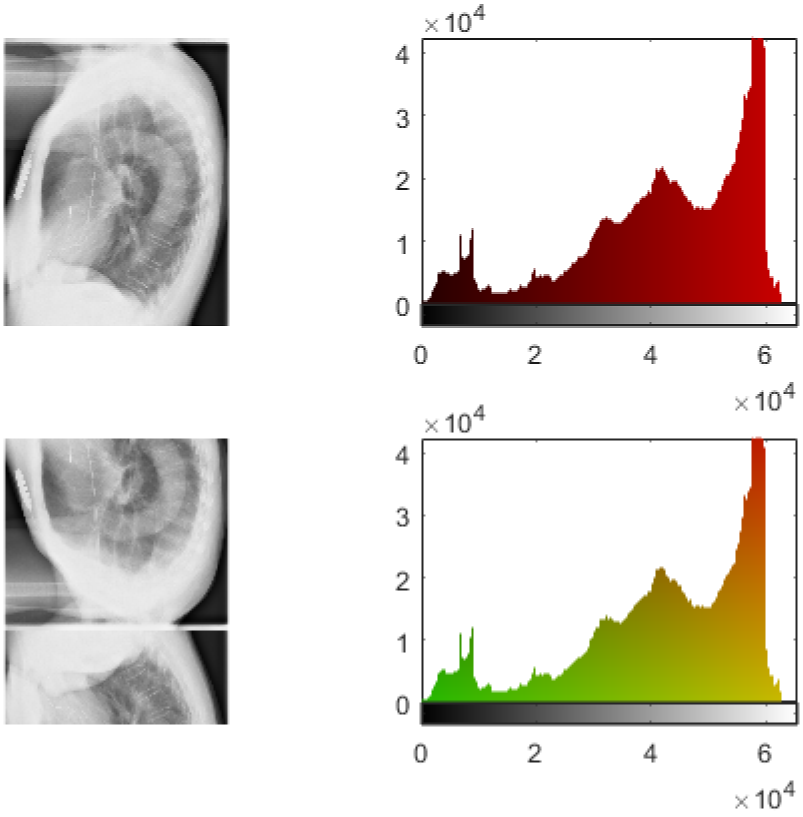
Published with MATLAB® R2015b

Quetion 8

A MATLAB script that reads a monochrome image, dispalys its histogram, and generates another gray-level image, different from the original, but hwise histogram is identical to the original's histogram

```
close all; clear; clc;
I = imread('cr_pacem_2.pgm');
I2 = circshift(I,650);
I3 = flip(I2);

subplot(2,2,1), imshow(I);
subplot(2,2,2), imhist(I);
subplot(2,2,3), imshow(I3);
subplot(2,2,4), imhist(I3);
```



Published with MATLAB® R2015b

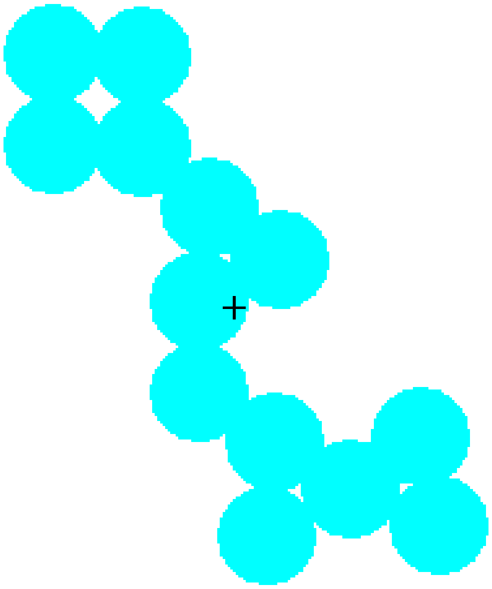
Quetion 9

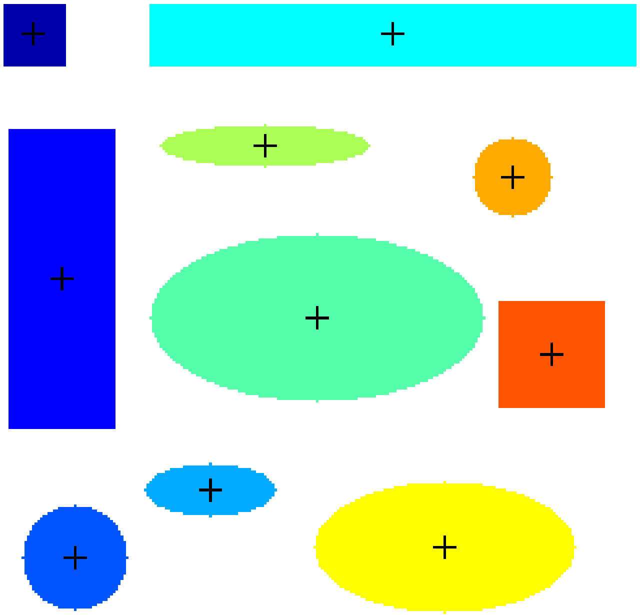
A MATLAB script that extracts the connected components of a binary image, diplays the results using different colors for each component and overlays a cros-shaped symbol on top of each component's center of gravity, whose coordinates can be obtained by the Centroid property

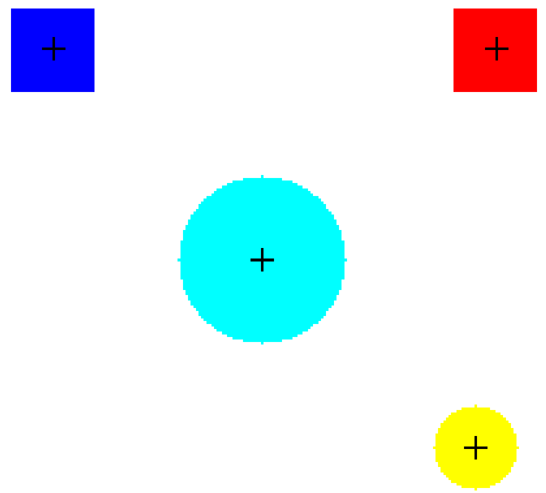
```
close all; clear; clc;
I = imread('circles.png');
Ign = im2double(I);
Ibw = im2bw(Ign,.5);
L = bwlabel(Ibw);
stat = regionprops(L,'centroid');
figure; imshow(label2rgb(L)); hold on;
for x =1: numel(stat)
    plot(stat(x).Centroid(1),stat(x).Centroid(2),'k+');
end

clear;
I2 = imread('Test3.png');
Ig2 = rgb2gray(I2);
Ign2 = im2double(Ig2);
Ibw = im2bw(Ign2,.5);
L2 = bwlabel(Ibw);
stat = regionprops(L2,'centroid'); %#ok<MRPBW>
figure; imshow(label2rgb(L2)); hold on;
for x =1: numel(stat)
    plot(stat(x).Centroid(1),stat(x).Centroid(2),'k+');
end

clear;
I3 = imread('TPTest1.png');
Ign3 = im2double(I3);
Ibw3 = im2bw(Ign3,.5);
L3 = bwlabel(Ibw3);
stat = regionprops(L3,'centroid');
figure; imshow(label2rgb(L3)); hold on;
for x =1: numel(stat)
    plot(stat(x).Centroid(1),stat(x).Centroid(2),'k+');
end
```







Published with MATLAB® R2015b