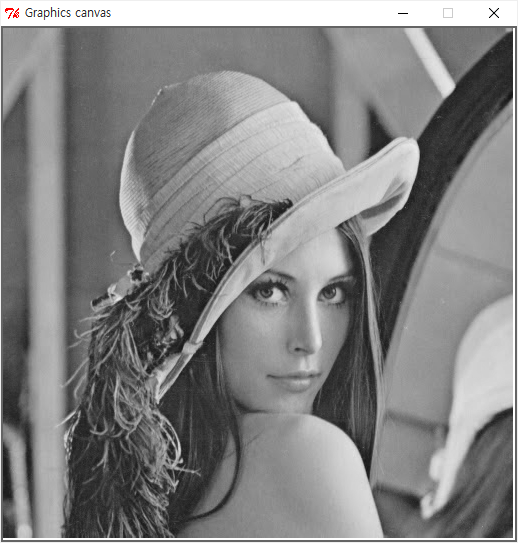
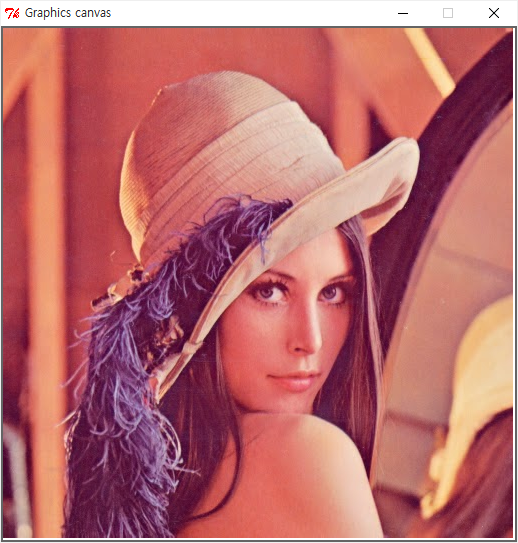
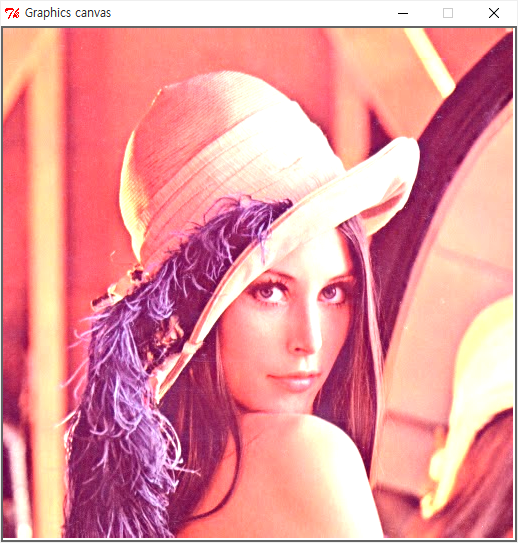
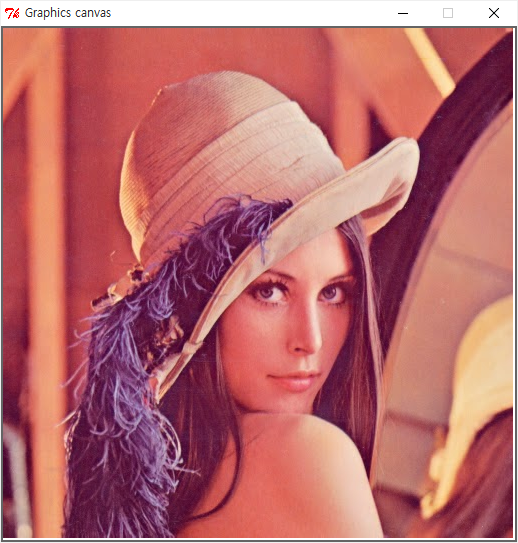
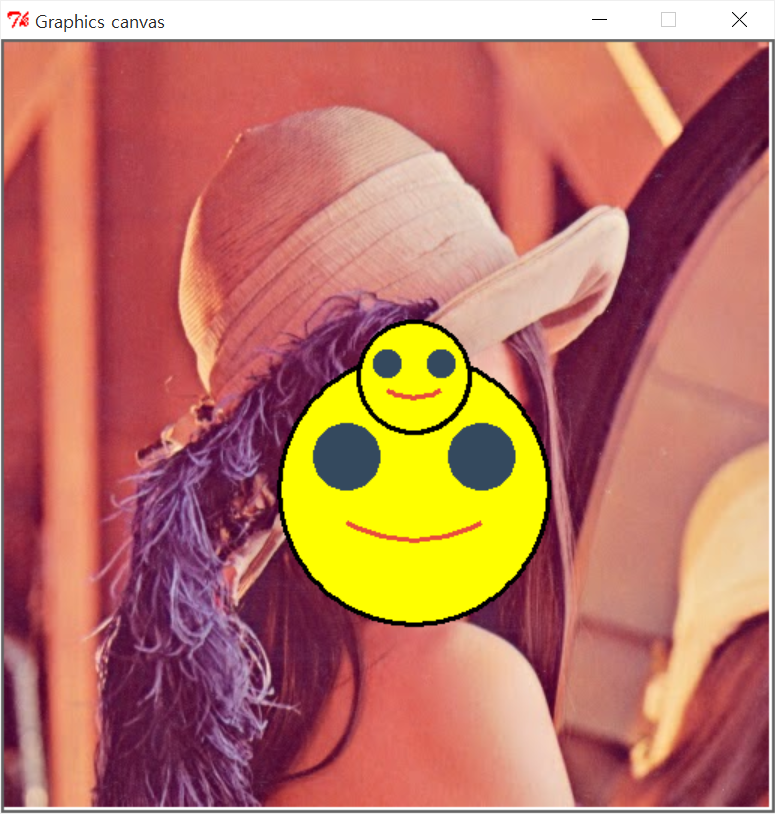
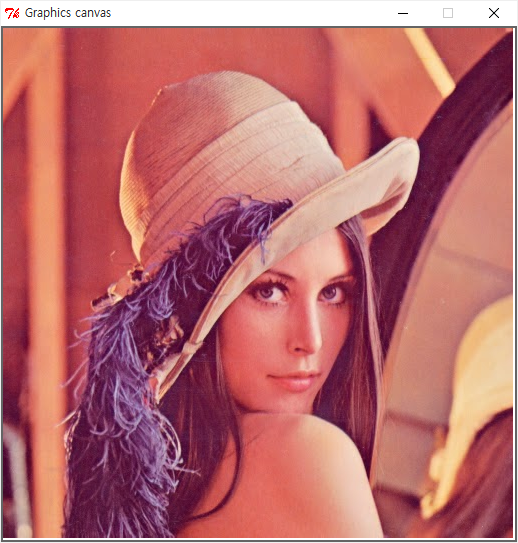
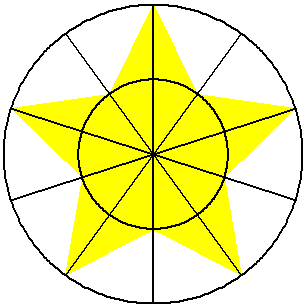
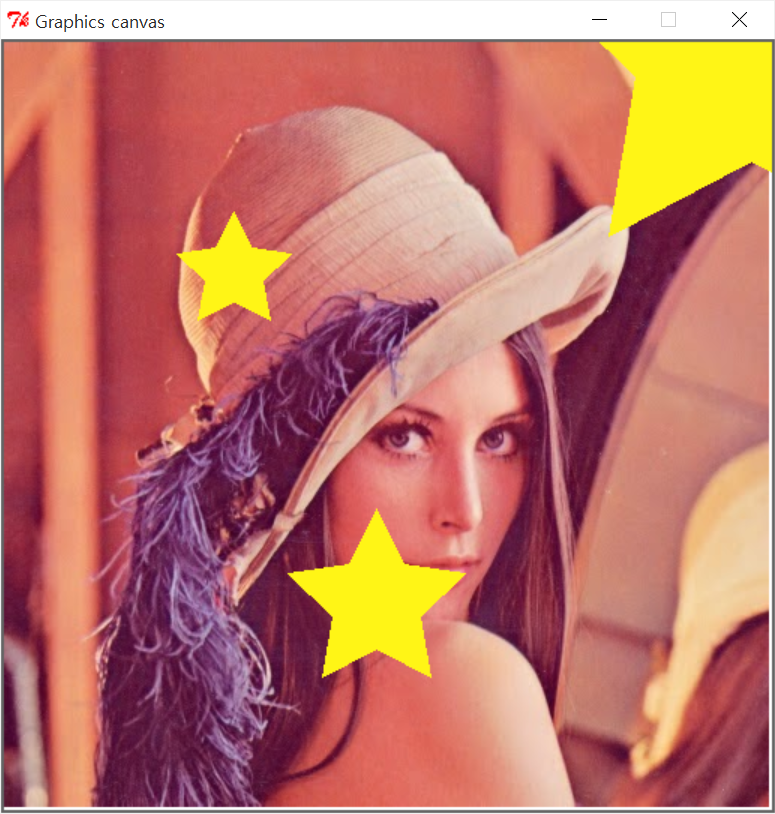
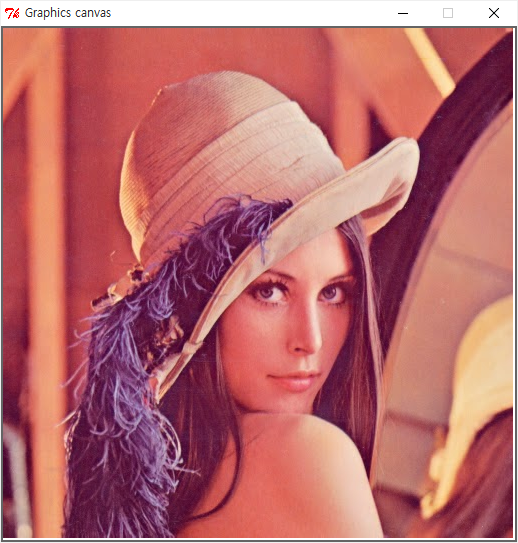
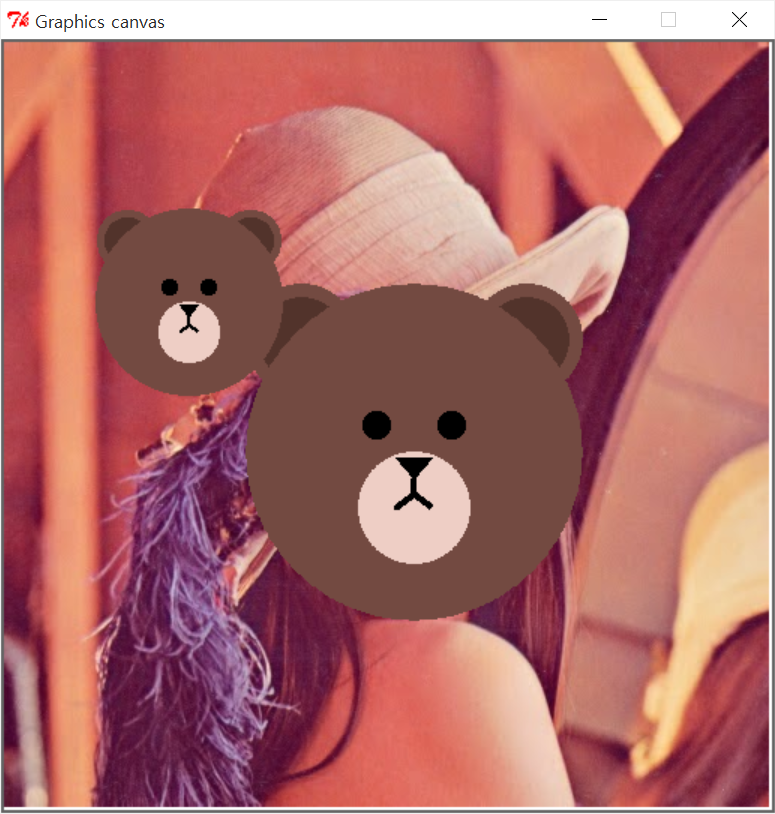
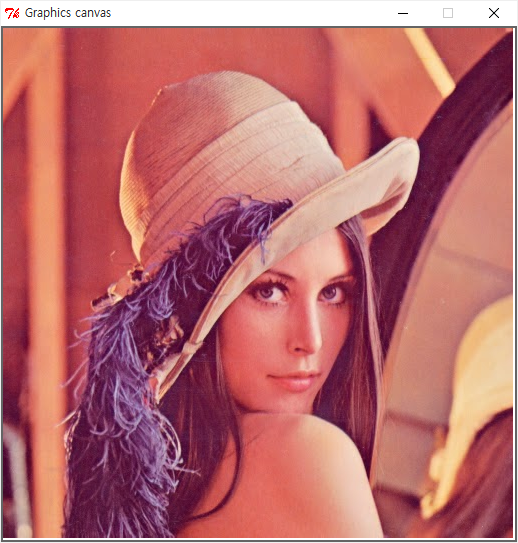
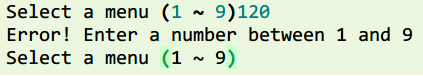
­­­­­CS101 HW#2 Report

지준섭

1. Introduction to the program  
   In this homework, I need to make simple photo editing program that has gray-scaling, add brightness, putting smile mark, star mark, and custom mark function. And here are the codes and algorithms what I made for function to be implemented.  
   **a. gray\_scale(img)**  
   To make image grayscale, we need to make each pixel’s RGB values same to the luminance value. From lecture note, I found out that computer should calculate to get luminance value.  
   So here is the code making all pixels’ RGB values same to the luminance values:  
   In this code, w and h saves the size of the img, then x and y can express all the pixels of the image with for. The variables R, G, B take the values of each pixel, and another variable avg gets the luminance value. Then we set the pixel’s value (avg, avg, avg) to change img into grayscale.  
     
     
   **b. make\_lighter(img, factor\_percent)**  
   This function should make img more brighter related to value factor\_percent. To implement this, just increasing each pixel’s RGB values related to factor\_percent is needed. Here is the code:  
     
   In this code, we get pixel’s RGB values and multiplies (1 + factor\_percent / 100.0) and set the pixel’s values new RGB values. Now the image will be brighter.  
     
     
   **c. put\_smile\_mark(x\_pos, y\_pos, size)**  
   We need to make circles and a spline to make a shape of smile mark. At First, make layer smile\_mark to add all the components of the mark at the future.  
   Then we’ll make face. Face’s radius must be the half of the value size and its center point should be (x\_pos, y\_pos). Also its border color will be black and its width equals to 3, and its color should be yellow.  
     
   Its depth is same as the global variable depth. And we added face to layer smile\_mark.  
   Now we’ll add eyes to the face. Making eyes is similar to making face.  
   You can see that the depth of the eyes are depth – 1. It means eyes are in front of the face.  
   Then we’ll make mouth. Using spline (curved path) will make things easier.  
   I picked three proper points that would make this spline looks like the smiling mouth. You can see the depth of the mouth is also depth – 1, just like depth of the eyes.  
   At last, we finished making smile\_mark, so we’ll add it to the canvas.  
   I set the depth of the smile\_mark as depth, and depth decreased – at last added it to canvas. By the decreasing of the depth, the mark we may add to the canvas in the future will be in front of the current mark.  
     
     
   **d. put\_star\_mark(x\_pos, y\_pos, size)**  
   To make star, we should divide the shape of star into some same pieces.  
     
   At first, we should make the triangle A and B. Then making rotations of the A and B will make the star. Here is the code:  
     
   theta means the degree , and the variable inner\_point\_one and inner\_point\_two mean the points that the inner circle and the star meets at. And the variable outer\_point is the point that the outer circle and the star meets at.  
   Now look at the for part. We’ll repeat making the two small triangles five times, so the code will make the star automatically. We can see the inner\_point\_one’s coordinates is decided as (x\_pos + rad \* sin((2 \* i - 1) \* theta), y\_pos - rad \* cos((2 \* i - 1) \* theta)), inner\_point\_two’s are (x\_pos + rad \* sin((2 \* i + 1) \* theta), y\_pos - rad \* cos((2 \* i + 1) \* theta)), outer\_point’s are (x\_pos + 2 \* rad \* sin(2 \* i \* theta), y\_pos - 2 \* rad \* cos(2 \* i \* theta)). Then we make small triangles decided by original point and two points, like (Point(x\_pos, y\_pos), inner\_point\_one, outer\_point) and (Point(x\_pos, y\_pos), inner\_point\_two, outer\_point).  
     
     
   **e. put\_my\_mark(x\_pos, y\_pos, size)**  
   I decided to make the mark of the bear. To make the mark of the bear, we need two circles to express outer ears, two smaller circles of inner ears, a big circle of face, two small circles to express eyes, one small triangle for nose, one path for mouth, and a little bigger circle to express the thing around the nose and mouth. Here is the result:  
     
     
   **f. range\_input(prompt, low, high)**  
   Limiting the range of input is important to prevent the errors of the program while running. In this function, the three arguments are given – prompt, low, high. So all we need to do is print the prompt message with the range (low ~ high), and check the input if it is inside the correct range. Here is the code:  
     
     
   And this is the result.  
   

w, h = img.size()

for y in range(h):

for x in range(w):

R, G, B = img.get(x, y)

avg = int(0.299 \* R + 0.587 \* G + 0.114 \* B)

#take luminance value of the RGBs to make grayscale

img.set(x, y, (avg, avg, avg))

img.commit()

w, h = img.size()

for y in range(h):

for x in range(w):

R, G, B = img.get(x, y)

R = (1 + factor\_percent / 100.0) \* R

G = (1 + factor\_percent / 100.0) \* G

B = (1 + factor\_percent / 100.0) \* B

#change RGB more brighter

R = int(R)

G = int(G)

B = int(B)

img.set(x, y, (R, G, B))

img.commit()

smile\_mark = Layer()

face = Circle(size/2, Point(x\_pos, y\_pos))

face.setBorderColor('black')

face.setBorderWidth(3)

face.setFillColor('yellow')

face.setDepth(depth)

smile\_mark.add(face) #make face

left\_eye = Circle(size/8, Point(x\_pos - size / 4, y\_pos - size / 8))

left\_eye.setBorderColor((52, 73, 94))

left\_eye.setFillColor((52, 73, 94))

left\_eye.setDepth(depth - 1)

smile\_mark.add(left\_eye) #make left eye

right\_eye = Circle(size/8, Point(x\_pos + size / 4, y\_pos - size / 8))

right\_eye.setBorderColor((52, 73, 94))

right\_eye.setFillColor((52, 73, 94))

right\_eye.setDepth(depth - 1)

smile\_mark.add(right\_eye) #make right eye

mouth = Spline(Point(x\_pos - size / 4, y\_pos + size / 8), Point(x\_pos, y\_pos + size / 4), Point(x\_pos + size / 4, y\_pos + size / 8))

mouth.setBorderColor((231, 76, 60))

mouth.setBorderWidth(3)

mouth.setDepth(depth - 1)

smile\_mark.add(mouth)

smile\_mark.setDepth(depth)

depth = depth - 1

canvas.add(smile\_mark)

A

B

pi = math.pi

theta = 2 \* pi / 10 #degree 36

sin = math.sin

cos = math.cos

rad = size / 4

star\_mark = Layer()

for i in range(5):

inner\_point\_one = Point(x\_pos + rad \* sin((2 \* i - 1) \* theta), y\_pos - rad \* cos((2 \* i - 1) \* theta))

inner\_point\_two = Point(x\_pos + rad \* sin((2 \* i + 1) \* theta), y\_pos - rad \* cos((2 \* i + 1) \* theta))

outer\_point = Point(x\_pos + 2 \* rad \* sin(2 \* i \* theta), y\_pos - 2 \* rad \* cos(2 \* i \* theta))

star\_tmp\_one = Polygon(Point(x\_pos, y\_pos), inner\_point\_one, outer\_point)

star\_tmp\_two = Polygon(Point(x\_pos, y\_pos), inner\_point\_two, outer\_point)

star\_tmp\_one.setBorderColor((255, 245, 22))

star\_tmp\_two.setBorderColor((255, 245, 22))

star\_tmp\_one.setFillColor((255, 245, 22))

star\_tmp\_two.setFillColor((255, 245, 22))

star\_mark.add(star\_tmp\_one)

star\_mark.add(star\_tmp\_two)

star\_mark.setDepth(depth)

depth = depth - 1

canvas.add(star\_mark)

check = True

while check == True:

in\_put = raw\_input(prompt + ' (' + str(low) + ' ~ ' + str(high) + ')')

int\_put = int(in\_put)

if int\_put >= low and int\_put <= high:

check = False

else:

print 'Error! Enter a number between ' + str(low) + ' and ' + str(high)

return int\_put

1. Explanations about additional functions that I want to add  
   I want to add the red / green / blue filter function. If we see things through colored lens, we see the world filled with the lens color. It’ll be great if I add this function to the program. To implement this function, I should take the input color that the user want to filter, from one of the Red, Green, and Blue. If the input was ‘red’, I should set all the pixels RGB value as (R, 0, 0). Then the image would be seen red.
2. What I learned and felt  
   I have studied a lot of image processing, I realized that the image editing program, like photoshop, are made like this. From this homework, now I am able to edit the values of pixels and change the whole image whatever I want. Maybe I can use this skill in researching – like counting the number of something in the image, or etc. It was very intersting and fun that I programmed the image editing program on my own.