CSC 101: Image Modifier
Due: 11 March, 2022
Points: 50

Write a program to modify images. The image format we will use is PPM. As part of writing this program, you'll know more about PPM images and you'll be able to modify images using your program.

You will be practicing the following concepts:

- variables
- integers
- if then/else
- loops
- functions
- lists
- files
- exception handling

PPM Images:

The PPM image format is encoded in human readable plain text. A PPM image has two parts:

1. Header
2. Body

PPM Header

The PPM header contains three undocumented lines at the beginning of the file (example shown below):

```
P3
4 4
```

```
255
```

The first line specifies the image encoding in the file. For this project, we will use the "P3" encoding specification. The second line specifies the number of pixel columns and rows present in this image. In this example, we have a 4 by 4 pixel image. The number on the third line indicates the maximum value for each red, green, and blue (RGB) element in a pixel. We will always use the max value of 255.

PPM Body

Below the header is the body of the PPM file. Each pixel has a red, green, and blue value as indicated by an integer number between 0 and 255. For example, the pixel with a RGB value of (255, 0, 0) represents the color Red.

Example PPM Image

Download the PPM image of the NY sky line. You can open the image with a PPM image reader software such as Ifranview. The image file can also be opened as plain text in a text editor, which allows us to edit the image.

Tasks:

Write a program to modify a PPM image in the following ways:
1. Negate the colors in the image (10 points)
2. Apply a high contrast to the image (10 points)
3. Apply a gray scale to the image (10 points)
4. Remove a primary color (red, green, blue) from the image (10 points)
5. Test the output of the above functions by **displaying the modified image** after each modification with a PPM image reader software such as Ifranview. (10 points)

Your program should follow the steps below:
1. Prompt the user to enter the input PPM file name, the output PPM file name, and one of the following commands to determine what modification should be applied to the input PPM image so that the modified image is stored in the output PPM file:
A. "negate"
B. "high contrast"
C. "gray scale"
D. "remove <color>" where <color> is "red", "green", or "blue"

2. Apply the image modification based on the user command. Your must define the functions mentioned below to perform the image Modification. Note that the original image file should not be changed; the modified image must be stored in the output PPM image file name. Also note that the output file should have the same header lines as the input file.

a. negate_image(): accepts the input image file object and the output image file object as arguments. For each pixel element, it takes the value, subtracts 255 from it and takes the absolute value of the result. This result is written to the output file.

b. high_contrast_image(): accepts the input image file object and the output image file object as arguments. For each pixel element, if the value is higher than 127, set it to 255 or else set it to zero. This result is written to the output file.

c. gray_scale_image(): accepts the input image file object and the output image file object as arguments. For each element in a RGB triplet (consecutive three elements), convert its value to the triplet's average. The result is written to the output file.

d. remove_red_image(): accepts the input image file object and the output image file object as arguments. This function sets all red values to 0 (red is the 1st value in an RGB triple). The result is written to the output file.

e. remove_green_image(): accepts the input image file object and the output image file object as arguments. This function sets all green values to 0 (red is the 2nd value in an RGB triple). The result is written to the output file.

f. remove_blue_image(): accepts the input image file object and the output image file object as arguments. This function sets all blue values to 0 (blue is the 3rd value in an RGB triple). The result is written to the output file.

3. Test the output of the above functions by displaying the modified image with a PPM image reader software such as Ifranview. (10 points)