CS111 Spring 2019 Project

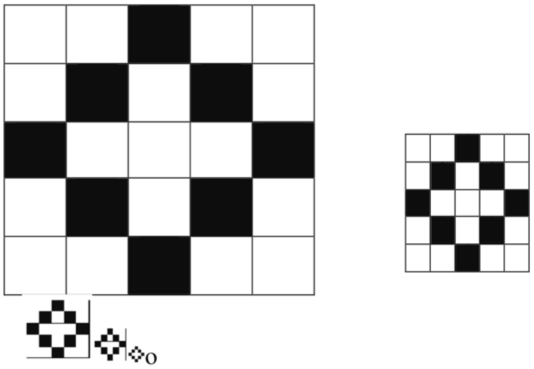
In this project you will be using the knowledge you gained from lectures and assignments and applying them to a lengthier programming assignment. The answers to some of your questions do not strictly reside in this document. You are encouraged to look at other sources for inspiration. This semester's project will involve the image manipulation. This is an *individual* project, and as such you will be held to the University's academic integrity policies. You are welcome to discuss strategies and ideas with other students (and will be encouraged to in recitation), but you must code the project on your own.

Good luck!

Graphics Background

Images and image manipulation programs have long been a staple of software projects. Programs like Photoshop ® and Gimp have become common tools for artists and creators. While much of what those programs do is outside the scope of this project and class, we're going to delve into some of the basics of images and how they can be manipulated.

Digital images are represented as a series of pixels. Pixels are like dots that make up an image. You may have heard the term "megapixel" describing resolutions of pictures. 1 megapixel refers to an image that is made up of a million dots (or pixels). In a given area, the more dots, the finer and more clear pictures will appear. The following diagram shows an image of the letter "O" comprised of 25 pixels in different sized images.



In a black and white image as above, the pixels are made of a binary value where black pixels have the value 1 and white pixels have the value 0.

Color images are slightly more complex. Rather than having a binary value, color pixels are actually comprised of 3 different values – Red, Green, and Blue (R,G,B). These three colors combined at different intensities are what our eyes see as the visible light spectrum. The values range from 0 to 255 (0 -> 2^8-1).

[You can explore the color wheel and it's RGB values using various web tools like: http://htmlcolorcodes.com/color-picker/]

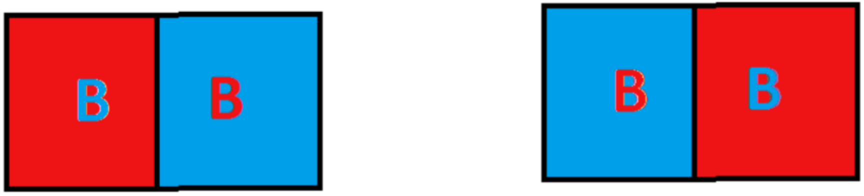
Part 1

Imagine that you are given an abstraction for an image that includes a set of individual colored pixels. The pixels are presented to you as an indexable set (like a 2D array). Every pixel has 3 values – red, green, and blue associated with it.

You will devise and implement an algorithm that can:

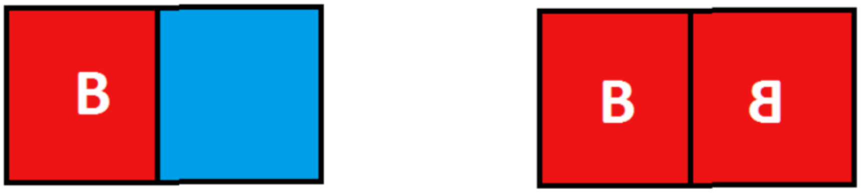
1. Count how many red pixels there are in an image.
2. Change all the red pixels in an image to blue pixels and vice versa.

Eg:



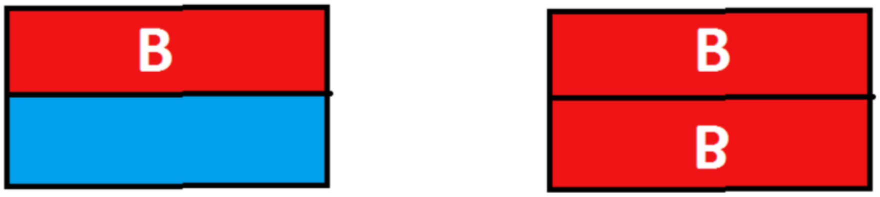
3. Take the left half of an image and reflect it over an imaginary vertical line going across the middle of the image.

Eg:



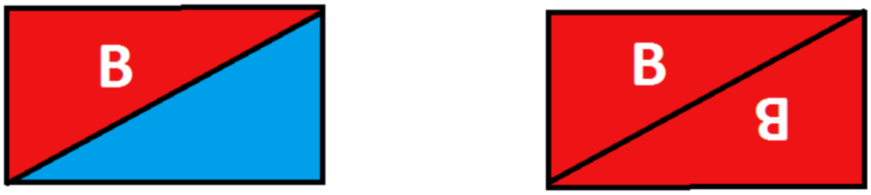
4. Take the top half of an image and reflect it over an imaginary horizontal line going through the image.

   Eg:



5. Take the upper half of an image and reflect it over an imaginary diagonal line going from the upper left corner of the image to the lower right corner.

   Eg:



The MediaComp Libraries

We provide you with a library and corresponding documentation that abstracts images and pixels into Java code. This opens up a world of possibilities for you to explore your creative and analytical sides.

To get started look at the API documentation for the library. Pay extra attention to the API for Picture/SimplePicture and Pixel. All methods you see in SimplePicture are also available for you to use in Picture objects.

As you learned in class, you can create an object by using the provided constructors. Specifically:

- Windows the backslashes in the path to the picture file MUST be escaped
  `Picture p = new Picture(“C:\\anapaula\\pics\\mypic.jpg”);`

- MacOS/Linux
  `Picture p = new Picture(“/home/anapaula/pics/mypic.jpg”);`

This creates a new Picture object from the file mypic.jpg. You may then call any of the methods below and pass in `p`. The object referenced by `p` will be manipulated by your methods.

To write your modified picture call the instance method write() on `p`.
`p.write(“C:\\anapaula\\pics\\mypic-modified.jpg”);` will write the changes to a new file called `mypic-modified.jpg`.

IMPORTANT

You MUST use the full path when writing your new image. Relative paths (e.g. ./foo.jpg) or simple filenames (e.g. foo.jpg) will not work correctly.

You can then view your modified image.