CIS 212: Project #2G
Assigned: October 27th, 2018
Due November 4th, 2018
(which means submitted by 6am on November 5th, 2018)
Worth 6% of your grade

Please read this entire prompt!

Assignment: You will begin manipulation of images

1) Write a struct to store an image.
2) Write a function called ReadImage that reads an image from a file
3) Write a function called YellowDiagonal, which puts a yellow diagonal across an image.
4) Write a function called LeftRightCombine, which combines two images.
5) Write a function called WriteImage that writes an image to a file.

Note: there is starter code, proj2G.c. My YouTube video will walk the starter code.

We will test it by check_2G that can be found on the class website. Check_2G has some built-in smarts about seeing if you have gotten the right images.

The rest of this document describes (1) PNM files, (2) the five steps enumerated above, and (3) submission instructions.

== 1. PNM files ==

PNM is a format that was used in Unix for quite a while. PNM utilities are regularly deployed on Linux machines, and are easily installable on Mac. They are available on ix as well.

To be able to view PNM files on VirtualBox, issue this command:
sudo pacman –Sy imagemagick

Then you can view an image using "display <file>.pnm"
(Example: display 2G_input.pnm)

We will only be supporting the "P6" format. You can learn more about P6 here:
http://netpbm.sourceforge.net/doc/ppm.html

But you should be able to get by without looking at the website. My YouTube video should help.

== 2.1 Image struct ==

Your Image struct will need a width, a height, and an array to store pixel data. As we discussed on the YouTube, image data is a 2D array of pixel data. A pixel contains 3 unsigned chars: one for red, one for green, and one for blue. There are multiple correct ways to store this data. We will be doing this by creating a Pixel struct, and having a one dimensional array. (See YouTube video.)

== 2.2 ReadImage ==

You will read a PNM/P6 file. The start of the read function is in the .c file, since I haven't lectured enough on string parsing.

== 2.3 YellowDiagonal ==

You will modify every pixel along the diagonal to be yellow. What does it mean to be on the diagonal? An image is a 2D array, so every pixel has an (i, j) index. A pixel is on the diagonal if i == j.

This function should not modify the input image. Instead, it should assign colors to the output. The output image should be the same as the input image, except on the diagonal. (This means you will have to assign values to every pixel in the output – the same value if it is off the diagonal, and "yellow" if it is on the pixel.)

The calling function is responsible for freeing the image that is returned by this function.

== 2.4 LeftRightCombine ==

This function will take two input images and one output image. It will combine them, left-to-right. The example image on the course website should be helpful in explaining the concept.

Important: when copying data from each input to the output, please keep in mind that the indices are different. Meaning, if you are copying a Pixel at index1 from input1, you are copying into a Pixel in output that is almost certainly at a different index, index2 (!= index1). Be sure to use GetPixelIndex, and be sure to make sure you are sending in the correct width and height.

== 2.5 WriteImage ==

You will write a PNM/P6 file. See the section on PNM files for information on how to view this file.

== 3. Turnin ==

Before you submit, make sure to test your code with "./check_2G". You should actually be calling "check_2G" every time you implement a function.

Turn in: only your proj2G.c source code file.