CIS 330: Project #3A
Assigned: April 24th, 2017
Due May 3rd, 2017
(which means submitted by 6am on May 4th, 2017)
Worth 4% 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 WriteImage that writes an image to a file.

Note: I gave you a file to start with that has interfaces for the functions

Note: your program should be run as:
./proj3A <input image file> <output image file>

We will test it by running it against an image file that can be found on the class website. We will test that it makes the image on the class website. So you can check that we have the same output before submitting.

Turn in: all of your source code & your Makefile. The Makefile should all have relative paths. Meaning that we can just run "make" and it won't be pointing at any funny directories that are specific to you.

The rest of this document describes (1) PNM files and (2) the four steps enumerated above.

== 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.

pnmtopng < file.pnm > file.png
is a useful utility, since it takes a PNM input and makes a PNG output.
The image processing utility "gimp" recognizes PNM, as do most version of the "xv" image viewer on Linux machines.

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

== 2.1 Image struct ==

Your image struct will need a width, a height, and a buffer to store image data. As we discussed in class, 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 ways to store this data, and they are all correct.

== 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.

This functions returns an Image. The calling function is responsible for freeing its memory.

== 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. It is on the diagonal if i == j.

This function should not modify the input image. Instead, it should create a new image. The new image should be the same as the input image, except on the diagonal.

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

== 2.4 WriteImage ==

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

== 3. Turnin ==

The grader will run your program as:

```
% tar xvf Proj3A.tar
% cd Proj3A
% make
% ,/proj3A /path/to/input/image/file /path/to/output/image/file
```

(and the paths to the images may or may not be relative)
If your program does not fit the above pattern, you will be docked.
Also, important: you should not include any images in your tarball submission.