

CIS 330: Project #3A  
Assigned: April 23<sup>rd</sup>, 2018  
Due May 1<sup>st</sup>, 2018  
(which means submitted by 6am on May 2<sup>nd</sup>, 2018)  
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 (3A\_c.c) 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 input image file (3A\_input.pnm) that can be found on the class website. We will test that it makes the output image (3A\_output.pnm) from the class website. So you can check that we have the same output before submitting.

The rest of this document describes (1) PNM files, (2) the four 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.

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 function returns an Image struct. 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. A pixel 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 ==

Before you submit, make sure to test your code on ix-dev. You should execute the provided grader program script (grader.sh) prior to submitting. It should be called within your project directory on ix-dev as follows:

```
./grader.sh 3A_c.c
```

If you pass all tests, that only assures that your code compiles properly with the correct input and output images, and follows the requirements of this prompt. However, your actual source code file will still be graded for good programming practices. The project will be graded on ix-dev.

Turn in: only your 3A\_c.c source code file.