TCSS333
C for System Programming
Programming Assignment 2
Image Manipulation

DUE: See Canvas Programming Assignment 2 link.

You are to submit a single .c file to Canvas and nothing else (do not zip anything, no Eclipse projects, etc.)

This assignment requires you to manipulate and combine the data that is stored in two image files and create two new image files based on the instructions seen below. There are several types of image file formats (bmp. Jpeg, tiff, gif, png). We will use the bmp (bitmap) format here. Bitmap files use the file extension of .bmp.

To keep things straight forward, you will be using 24-bit color with the width and height of the images as always multiples of 8. Several examples of such files will be posted on Canvas. Your program should also be able to work with other images of the same type. Your goal is to read two image files and from them create two new image files:

- A double explosure of the two input files or images (**you should name this output file “blend.bmp”**)
- An 8x8 checkerboard pattern of the two input files or images (**you should name this output file “checker.bmp”**)

In order to succeed in this assignment, you will need to learn some information on the data that is stored within an image file. The two input bitmap files you will use for the final results are **"in1.bmp"** and **"in2.bmp"** (hopefully you will recognize these two individuals).

You can find out how .bmp files are formatted by examining them with a hexadecimal editor and comparing what you see to the description at:

http://en.wikipedia.org/wiki/BMP_file_format

A very useful hexadecimal editor you may want to use is here:

http://www.chmaas.handshake.de/delphi/freeware/xvi32/xvi32.htm

Bitmap files are not all alike and Wiki article provides more information than you need. The most important parts are the two tables labeled "Bitmap file header" and "Windows BITMAPINFOHEADER". The first one explains what is contained in the first 14 bytes of the file and the second one explains the next 40 bytes. If the image is w pixels in width by h pixels in height, then there are a total of w*h pixels. A pixel in a .bmp file is recorded using three bytes standing for the red, green, and blue color components of the pixel. So an image file with image dimensions 100 by 200 will store 20,000 pixels. That same file will have a total size of:

14 bytes Bitmap file header
40 bytes Windows BITMAPINFOHEADER
<u>60,000 bytes</u> width * height * (bytes per pixel) = 100 * 200 * 3
60,054 bytes total

In the headers, you will find
size of the BMP file in bytes
bitmap width in pixels
bitmap height in pixels
the image size

The wikipedia article and a hex editor will help you figure out exactly where they're located. You will need to read some of those numbers into int variables so you can use them in your program. The rest of the header information should not be changed, so you can simply read those parts into char arrays where those bytes will be stored until you later write them out to your output files.

Once you have finished reading through the headers (using a number of fread's), you will be ready to read the pixel data that makes up most of the input files, again using fread. You'll put all that pixel data in two large 2D arrays, one for each input image.

You will read parts of the headers into 1D arrays and use 3-4 2D arrays for the image data read and results written.

The double exposure image is created by calculating the average red, green, and blue values for each pixel, that is the average between image 1's pixel at row r, column c and image 2's pixel at row r, column c. We assume the two images have the same dimensions! As you calculate these averages, keep in mind that a char is a lot like a small number. By default, the type char can contain either a positive or negative number (-128 to 127). But R,G,B are values from 0 to 255. Therefore instead of using char to store your pixel data, use "unsigned char".

For the checker board, you will be combining the two images to create an 8 by 8 checkerboard. A normal checkerboard is made of alternating black and white squares. The checkboard you create will be made up of alternating squares taken from the matching position in the two original input images. To make this process simpler, we assume the width and height of the input images are both multiples of 8 (see HW 2 Sample Images.jpg).

After creating pixel data for your new images, you need to write out two .bmp files by reversing the same process you used to read in the starter file, this time using fwrite instead of fread.

Be ready to use a hexadecimal editor with your generated images, especially if Paint (or similar program) can't open the image. That's likely because the headers have been damaged.

Your program will also be graded on good coding practices such as indentation, meaningful identifiers, proper spacing, etc.