

Goal:**Manipulation of BMP image files:**

- (1) Convert a color image to greyscale.
- (2) Convert a color image to sepia.

Objective:

A second gentle refresher on C using loops and function calls.

Background:

A *greyscale* image is an image in which the value of each pixel is a single sample, that is, it carries only intensity information. Thus greyscale images are composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest.

A *sepia* image is reddish-brown in color, named after the rich brown pigment derived from the ink sac of the common cuttlefish. Sepia resembles the effect of aging in old photographs, and of older photographs chemically treated either for visual effect or for archival purposes.

Download:

Download and unpack file lab2.zip from Camino. It contains the same pre-compiled library file (libbmp.a) and include file (bmp.h) that was used for lab1, but a different partially completed main program (lab2.c).

Assignment:

Complete the source code for each the following two functions that are located within the provided main program (lab2.c):

```
IMAGE *ColorToGreyscale (IMAGE *image) ;
```

Replaces the color of each pixel in the image by a grey color whose intensity is the same as the perceived brightness of the original pixel. Returns a pointer to the greyscale image.

Note: Use the brightness function created for lab 1.

When done correctly, the result should look similar to the example below:

Original Image**Greyscale Image**

```
IMAGE *ColorToSepia (IMAGE *image) ;
```

Replaces the RGB components of each pixel in the image by a corresponding set of sepia RGB components and returns a pointer to the image. The new RGB components are computed as:

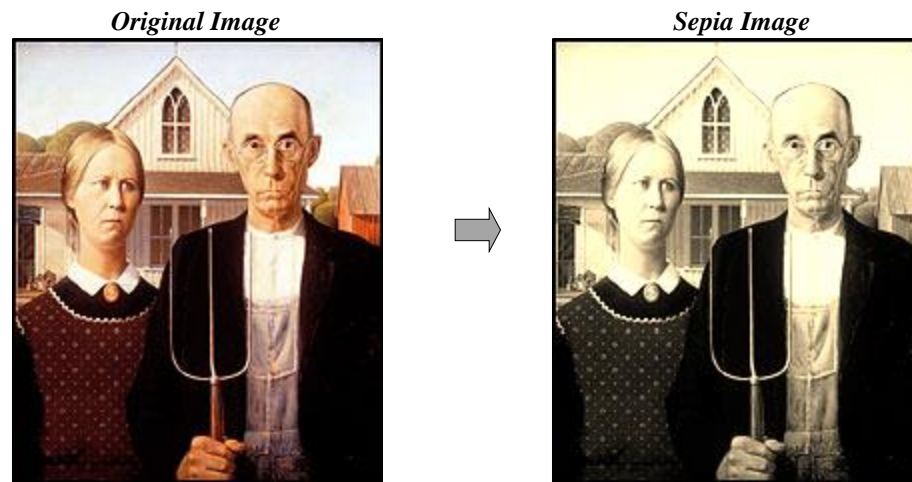
$$newred = 0.393 \times oldred + 0.769 \times oldgrn + 0.189 \times oldblu$$

$$newgrn = 0.349 \times oldred + 0.686 \times oldgrn + 0.168 \times oldblu$$

$$newblu = 0.272 \times oldred + 0.534 \times oldgrn + 0.131 \times oldblu$$

Note: The values produced by these formulas must be limited to a maximum of 255.

When done correctly, the result should look similar to the example below:



Compilation: Compile and link your program using the following command line:

```
gcc -o lab2 lab2.c -L. -lbmp
```

Execution: Execute your program using the following command syntax:

```
./lab2 src-file dst-file {option#}
```

When Done: Demonstrate proper operation of your program to the teaching assistant and upload the completed source code for file lab2.c to the lab drop box on Camino. Do not upload any other files.