Homework 2, TCSS 333 Autumn 2015
Due: Monday, November 2, 9:00 a.m.

OBJECTIVE

The objective of this assignment is to give you more practice with Unix, with using 2D arrays, and with binary representation of data.

ASSIGNMENT SUBMISSION

To get credit for this assignment, you must

- ✓ complete Unix tutorial (25%)
- ✓ write a program in C (75%)
- ✓ submit your files through Canvas exactly as instructed (naming, compatibility, etc.)
- ✓ submit your assignment on time

UNIX TUTORIAL

Go to Code Academy website (https://www.codecademy.com/courses/learn-the-command-line) and complete the other two modules from *Learn the Command Line* tutorial: *Redirecting Input and Output* and *Configuring the Environment*. After you complete both tutorials, take a snapshot that shows your name and either the completed course or the two tutorials - upload to Canvas as `jpg, gif, png` or `pdf`. Your file is to be named `<yournetid>_badges.<extension>`. If you can find some other way of showing completed lessons that includes your name, you may take a snapshot of some other screen.

C PROGRAM

Problem Statement

Create a program that reads 24-bit bmp files and generates new versions of the original image. The new versions to be handled are:

- a copy of original with all red hue removed
- a copy of the original in grayscale or sepia
- a copy of the original flipped horizontally or vertically
- a copy of the original mirrored horizontally or vertically

Your program is to prompt for the name of the input file and its pixel dimensions. Assume valid input and that the filename entered should be followed by a .bmp extension.
This is a sample run of the program:

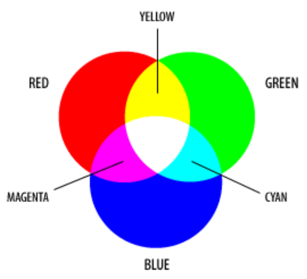
```
---
Enter the filename: test1
Enter height and width (in pixels): 160 240
Done. Check the generated images.
---
```

Note that the generated files should be named as

```
copy1.bmp
copy2.bmp
copy3.bmp
copy4.bmp
```

BMP files

A 24-bit bmp file is capable of storing 2d images. A file consists of a 54 byte header and the rest of the file is pixel information. Each pixel is represented by 3 bytes, where each byte contains the information for blue, green, and red hues – in this order. Each hue can be of value 0..255, where 0 signifies the complete lack of that color, while 255 signifies its full saturation. New colors are generated by mixing different intensities of this basic color palette. All together 256^3 = 16,777,216 colors can be represented using this scheme.



Note that the image height and width in pixels correspond to how we store information in a 2D array.

width

height

	0	1	2	3	4	5	.....				N
0	Pixel 1 B	Pixel 1 G	Pixel 1 R	Pixel 2 B	Pixel 2 G	Pixel 2 R	.....		Pixel X B	Pixel X G	Pixel X R
1											
....											
M											

A sample file `bmpchange.c` is provided to show how to read and write such files, and how blue and green components of each picture could be switched. You are to use this file as the basis of your program.

Several small bmp test files are provided with this homework.

Pixel Manipulation

Grayscale Algorithm
For each pixel calculate its average value of BGR:
(blue_value + green_value + red_value) / 3 = average
Replace original BGR values with average, average, average

Sepia Algorithm
For each pixel, recalculate its BGR values as:
newB = (R * 0.272 + G * 0.534 + B * 0.131)
newG = (R * 0.349 + G * 0.686 + B * 0.168)
newR = (R * 0.393 + G * 0.769 + B * 0.189)

Vertical Flip

Before

a	b	c
d	e	f
g	h	i
j	k	l

After

c	b	a
f	e	d
i	h	g
l	k	j

Before

a	b	c	m
d	e	f	n
g	h	i	o
j	k	l	p

After

m	c	b	a
n	f	e	d
o	i	h	g
p	l	k	j

For each row in old image

- Copy the row in reverse

Vertical Mirror

Before

a	b	c
d	e	f
g	h	i
j	k	l

After

a	b	a
d	e	d
g	h	g
j	k	j

Before

a	b	c	m
d	e	f	n
g	h	i	o
j	k	l	p

After

a	b	b	a
d	e	e	d
g	h	h	g
j	k	k	j

For each row in old image

- Copy the first half of the row as is
- Copy the first half of the row reversed

Other Specs

- Your program is to be contained in a single c file named `<netid>_hw2.c`
- Your program is to be compatible with the `gcc` version discussed in class
- Your program has to follow basic stylistic features, such as proper indentation (use whitespaces, not tabs), meaningful variable names, etc.
- Use memcpy to copy the matrix
- **Your program must compile in gcc gnu 90 – programs that do not compile will receive a grade of 0**
- Your program should include the following comments:
    - Your name at the top
    - Comments explaining your logic
    - If your program does not run exactly as shown above, explain at the top how to run your program – you will not receive full credit but at least you will receive some credit rather than none

Extra Credit (15%)

Provide some other interesting manipulation of the original image – explain in comments at the top of your code.

Program Submission

On or before the due date, use the link posted in Canvas next to Homework 2 to submit your tutorial snapshot and your C code. Make sure you know how to do that before the due date since late assignments will not be accepted. Valid documentation file formats are: pdf, jpg, gif, png. Valid program format: a single file .c