Programming Assignment 6:

Image rewriting for limited color palette

Due Tuesday, October 17th at 11:59 pm

(Note: this is a hard deadline)

120 points

1. Overview

The purpose of this assignment is to give you experience computing with structs, arrays and functions. It will also give you an opportunity to employ file input/output functions.

For this assignment, you are provided with "skeleton code" (mostly constructed code that you need to complete). The file is heavily commented to indicate your task, but the main points are repeated here.

You will be provided with a zip file containing the p6a and p6b sub-directories and files.

p6a: defining a struct, creating an array of structs, and accessing the members of the struct within an array

The purpose of this program is to read in a file containing the text of a color palette into an array of structures, and to then display the contents of that array.

- **p6a/** contains
  - palette.h - a header file ; your task #1 is in here (define the color structure)
  - p6a.c - starter code for your implementation of p6a - you will add 3 lines of code (tasks 2-4)
  - palette.c - partial implementations of the get_palette and show_palette functions (tasks 5 & 6)
  - palette.txt - contains a plain text definition of the colors in the palette, with each row describing one color: a single-letter color code, then the RGB values of the color, then a color name. This palette contains 24 color definitions.
  - p6.output.txt - sample output from a run of **./p6a palette.txt 24**
  - Makefile

- First watch the demo video of how p6 should work and the overview of your tasks.
- Then review the provided code.
- Your tasks:
  1. in palette.h , provide the definition for the color structure used in this program
  2. in p6a.c -- add the 3 lines of code as described in the comments in the file
  3. in palette.c - complete the print statements in get_palette and show_palette
- Compile and test your program to see if you get the desired output as seen in the video and found in p6.output.txt

p6b: map a given image to a new image using colors from a specified palette

The purpose of this program is to read in an image file, map each pixel in that image to the closest color in a limited palette of colors, and write out an output image file with the new colors from the selected palette.

- **p6b/** contains
  - palette.h - a header file
  - p6b.c - starter code for your implementation of p6b
  - palette.c -
  - palette.txt
  - Makefile
  - example input and corresponding output files:
    - boxes.ppm and boxes.out.ppm
    - image1.ppm and image2.ppm
    - vert_stripes.ppm and vert_stripes.out.ppm

Watch the p6b demo video.

Read through p6_b.c , palette.h and palette.c

First, you will want to copy-paste in your code from p6a as indicated in comments in the files palette.h (the definition of the color structure), in p6b.c (the lines of code to create the array and read the palette in from the file) and in palette.c (the completed scanf and printf statements in get_palette and show_palette). Note that p6b does not invoke show_palette.

Your remaining tasks are:

- in p6b, add a line of code to invoke the **map_to_palette** function, as indicated in the comments

- in palette.c, implement the function
    - **struct color choose_closest(struct color pal[], struct color orig_color);**
    - from the calling code, the method should receive an array of **struct color** structures (the palette info that was read from the file by get_palette) and a single **struct color** for a pixel from the original image
    - it should return a **struct color** containing the updated pixel color (the closest color in the palette)
    - the code is heavily commented .. this is similar to Proj2, but much more streamlined with an array of structures

- in palette.c, implement the function
    - void **map_to_palette**(char input_image_fname[80], char output_image_fname[80], struct color palette[], int palette_size);
    - this function is partially/ largely implemented
    - you need only insert code as indicated in the comments

2. What to submit

Via gradescope, submit a zip file containing the updated p6a and p6b subdirectories.

The p6a subdirectory should contain ONLY:
- p6a.c
- palette.c
- palette.h

The p6b subdirectory should contain ONLY:
- p6b.c
- palette.c
- palette.h

The code should compile without warnings.

Do **not** submit any extraneous files (no executables, no image files).

3. Notes on Collaboration

You are required to work individually on this assignment. **Please do not consult anyone other than your instructor on any aspect of this assignment**. If you submit questions via the class piazza page, please don't include significant source code.

Review the section on "Academic Integrity" on the Course Syllabus for the policy on what happens when copying or sharing code.

For any of the programming assignments in lecture and/or lab, it is considered cheating to do any of the following:

- discuss in detail the C code in your program with another student
- show or share the C code in your program with another student
- use C code obtained from another student, or any other unauthorized source, either modified or unmodified (each student is responsible for protecting his or her files from access by others)
- use re-engineering tools
- submit work of others, from the Internet or any other source, even if attributing the work to others

4. Grading Rubric

Functionality

generates correct output

__/ 20 p6a

__/ 80 p6b

Submission details

__ / 5 no extraneous files

__ / 15 clean compile

__ / 120