Assignment 2 – Turning Image upside down

CPSC 1021 – Spring 2020

CLEMSON School of COMPUTING

Introduction

In this assignment you will use the following skills:

1. Use of classes and classes containing objects as members.
2. Use of a container class (vector) to keep track of data.
3. Use of file I/O streams.
4. Use of makefiles

Part I – Problem at hand.

This project is done in C++ in *teams of 2 (and no more than 2), or individually*. Your teammate should be from the same lecture section of the class, since you have done the bulk of the work together already and will use assignment1 files as a start up for asg2. You are NOT allowed to discuss your solution or share your asg1 or asg2 files with other teams/students, except your own partner (if you have one). Only one person from each team should submit the files, and all files (except makefile) should have author/s names in the comments on top of the file.

You will work with .ppm and .pgm files again, but this time you will only use the P2 and P3 files in ASCII format. You can use the same image files as you used for asg1, or find other ones you like. Your program will read data from a .ppm (or .pgm) file and produce an upside-down copy of the original. This copy will not look like a mirror image of the original. It will look like an image that was turned 180 degrees. You will have to devise a clever way to achieve this.

Part II – Implementation Requirements

1. You can use the same classes you have created for asg1: ColorPixel (*colorPixel.h*) that will contain the red, green, and blue values of each color pixel, and class GrayPixel (*grayPixel.h*) to contain values of the gray pixel. Again, provide parameterized constructors, destructors, and getter methods for each class. If you have a parameterized constructor, you will not need setter methods to save the pixel values, as the only time you will need to set values of the pixel is when you are creating an instance of that pixel. This will be a very small class with most, if not all functions inlined. (Same as in asg1).

2. You will create a class Ppm and a class Pgm. Each class will have a header file (named *ppm.h* and *pgm.h*) and an implementation file (*ppm.cpp* and *pgm.cpp*). You will need member variables for storing magic number, height, width, max_value for each pixel, (other members, if needed), and a vector of ColorPixel or GrayPixel objects, correspondingly.

3. Create a main function that will drive the processing. Your file should be called upsideDown.cpp. Your executable will take two command line parameters – the original image file and the output image file.

4. Create a makefile and provide the functionality to remove object files (*clean*).

5. Create a tarball containing all the source code and the makefile, but *not* the object code. Please run *clean* before creating a tarball. You do not need to submit image files. Graders will use their own image files to test your program.

6. Submit to canvas before the deadline: **April 16th before 5pm**.

NOTE: If you submit the archive that cannot be open, or is corrupt, you will not be given the opportunity to redo the work and resubmit. You have one shot to get it right.

How to create your own .ppm and .pgm files in GIMP (so you can use them to test your program)

Two *.ppm* and two *.pgm* files (ASCII and raw format) are provided for you. They can be found in Canvas under the assignment1 link: chips.ppm, chips.pgm, potatochips.ppm, potatochips.pgm. BUT if you would like to test your program with some more interesting/exciting files, you can convert your favorite image to *.ppm* or *.pgm* format in GIMP. Open you image in GIMP, then click on Files-> Export As, give it a name on the very top, then on the bottom click on Select File Type (by Extension), scroll down, select .ppm (or .pgm). After that it will bring up a box with options to save in ASCII or raw format. Select format and save it.

Part III. Where to Get Help

1. First of all, try to use textbook, online sources and search engine of your choice to look at code examples, programming concepts explanation, etc. You need to learn to be independent and to be able to use available resources to help yourself.

2. You can also get help with this assignment from the course teacher or TAs. Though TAs may not be familiar with the details of the assignment or design requirements, they can help you with programming questions and debugging.

3. Many questions can be answered in e-mail. Please do not e-mail code unless you were asked to do so. Do this ahead of time, not at the last moment. You are still responsible for submitting your assignment on time, whether you got your question answered or not at the last moment.

   Most importantly: make a plan A and a plan B. You should always have a plan A and a plan B!! Unexpected circumstances happen, and it is better to be safe, than sorry. Start working right away, DO NOT WAIT TILL THE LAST MOMENT. Make frequent backups!!!!