Due Tue, October 17, 11.59pm

In this project you develop an image editor to perform simple operations on input images, with the following learning outcomes:

- Read/Write PPM images
- Images will be held in a C++ class, and image data will be allocated based on the input image dimensions.
- Image operations will take an input image and produce an output image in PPM format.
- Use of constructors and destructors.

In part 1, you will develop the basic Image class and its methods; in part 2 you will perform image processing operations using the class.

Image Representation

Your image class should hold the following attributes and implement the associated methods (may include additional attributes/methods as needed).

```
class Image {
   private:
                                    // image dimensions
      int width, height;
                                    // pointer to the dynamically allocated image array
      int *image_array;

   public:
      Image();        // constructor - creates an empty image object,
                      // creates an image object by reading the input file
      Image(string input_file);
                      //creates an image object using the given dimensions
      Image (int width, int height);
      ~Image();       // destructor - provide as many destructors as needed

                      // accessors/mutators
      int getWidth();
      void setWidth(int w);
      int getHeight();
      void setHeight(int h);

                      // set/get an image pixel given row and column addresses
                      // pixel is a 3 element r,g,b triple
      void getImagePixel (int col, int row, int *pixel);
      void setImagePixel (int col, int row, int *pixel);

                      // reads an image from  the given input image file
      read(string infile);
                      // writes image to file
      write(string outfile);
}
```

In this part of the project, you will implement and test the Image class detailed above. **You will continue to use the pointer based image representation from the earlier project**. Images will continue use the PPM (text) format as in previous projects. Constructors and destructors will manage the allocation/deallocation of the image array and any other allocated memory.

Tasks.

You will be given input images in PPM format for testing. You will test your class using a driver program as follows:

1. **Read/Write.** Use the class methods to read in the given input images and write it out (to a different file) in PPM format.
2. Use the setPixel() member function to draw a horizontal 20 pixel red band across the image. Write the image to a file in PPM format.
3. **Documentation.** Follow the earlier instructions for documenting this class and generate doxygen based documentation; **each of your methods should be documented, including each input parameter, return value, etc.**

Evaluation:

- As per rubric (on Canvas).
- **To Turn in to Canvas:** All source code files, two of the images (from testing as described above) in PPM or PNG format, doxygen based documentation.
- **Late Policy.** Upto 2 days and for a maximum of 50% credit. a reduction of 25% credit, each day. No credit beyond 2 days past the deadline.