CS 3005: Programming in C++

Digital Image Processing - Operator Overloading

Digital Image Processing is pervasive in today's technology. Using computer algorithms to manipulate the pixels of digital images can change or enhance the image.

Assignment

In this assignment, you will modify your [PPM] class to support a required [API]. The bulk of the requirements will come as overloaded operators. You are required to write a program that tests your class.

Programming Requirements

Your [PPM] class must have the following methods.

- A default constructor. This is the one that takes no parameters.
- An operator to send the object to an [std::ostream]. e.g. [fout << ppm_object;].
- An operator to obtain an object from an [std::istream]. e.g. [fin >> ppm_object;].
- All six comparison operators to compare [PPM] objects with [PPM] objects. The comparison criteria is the number of pixels in the image. More pixels is considered greater than, less pixels is considered less than, and same number of pixels is considered equal to. e.g. [ppm1 < ppm2].
- The operator to add one [PPM] object into another. e.g. [ppm1 += ppm2]. This should add each pixel of the images. A pixel addition is completed by adding the red channels, adding the green channels and adding the blue channels. If any channel exceeds the maximum color value, set the channel to the maximum color value.
- The operator to subtract one [PPM] object from another. e.g. [ppm1 -= ppm2]. This should subtract each pixel of the images. A pixel subtraction is completed by subtracting the red channels, subtracting the green channels and subtracting the blue channels. If any channel is less than 0, set the channel to 0.
- The operator to add the values of two [PPM] objects, creating a new one. e.g. [ppm3 = ppm1 + ppm2;]
- The operator to subtract the values of two [PPM] objects, creating a new one. e.g. [ppm3 = ppm1 - ppm2;]
- The operator to multiply a [PPM] object by a [double]. e.g. [ppm1 *= 0.67]. This operation will cause each channel (red, green, blue) of each pixel to be multiplied by the number, then converted back to integer. If any channel would exceed the maximum color value, set it to the maximum color value. If any channel would become less than 0, set it to 0.
- The operator to divide a [PPM] object by a [double]. e.g. [ppm1 /= 0.33]. This operation will cause each channel (red, green, blue) of each pixel to be divided by the number, then converted back to integer. If any channel would exceed the maximum color value, set it to the maximum color value. If any channel would become less than 0, set it to 0.
- The operator to multiply the values of a [PPM] object by a [double], creating a new [PPM] object. e.g. [ppm3 = ppm1 * 0.67;]
- The operator to divide the values of a [PPM] object by a [double], creating a new [PPM] object. e.g. [ppm3 = ppm1 / 0.25;]

User Interface Requirements

There are no UI requirements. Your program must test all of your class methods and operators.

PPM Image File Format

```
P6 WIDTH HEIGHT MAX_COLOR_VALUE
BINARY REPRESENTATION OF COLORS FOR EACH PIXEL IN THE SAME ORDER AS THE COLOR FILE
```

Additional Documentation

- Grayscale on Wikipedia
- Digital Image Processing on Wikipedia

Show Off Your Work

To receive credit for this assignment, you must upload the source code (.cpp and .h files) and the Makefile to
the Canvas submission system.

Additionally, the program must build, run and give correct output.