

No Debug For J

Lab J

Background

Pictures that you take with your phone or download from the Internet are essentially a large number of fields that make up what are known as pixels. Pixels are the small dots on your TV or monitor that can display a blending of three colors: red, green, and blue. You've probably heard of a resolution such as 1920x1080. That means there are 1920 dots spanning the width and 1080 dots spanning the height of your monitor. Essentially, higher the resolution, the more dots and the more "resolute" the picture on your monitor becomes.

Problem

In this lab, you will be reading a file called a portable pixel map (PPM). A portable pixel map is a color picture that has a text format, making it easy to use input file streams to read from. You will be reading from this file, making a series of adjustments, and then outputting the transformed picture to an output file.

Portable Pixel Map File

A portable pixel map (PPM) file has a specific text format:

Clara Van Nguyen (CS102 TA) has put together a [great reference for PPM files](#) and their associated format.

[Sample PPM File \(Bee\)](#)

[Sample PPM File \(Parents\)](#)

[Online PPM Viewer](#)

Pixel Structure (Class)

Prototype a new structure that stores information about a pixel and contains the following members:

1. Three public integers: red, green, and blue
2. A public 3-argument constructor that takes three integers: red, green, and blue. This constructor will set the red, green, and blue member variables to the red, green, and blue parameters.
3. A public 0-argument constructor. This constructor will call the 3-argument constructor and pass 0 for red, green, and blue.

Picture Class

Prototype a new class that stores pixel information in a "vector of vectors" given the following declaration:

```
vector<vector<Pixel>> picture_data;
```

The Picture class will contain the following members:

1. A private vector of vector of pixels (declaration shown above).
2. A public function that takes an input file stream and reads a given PPM file into the private vector. This function will return true if the file was able to be fully read, or false otherwise.
3. A public function that takes an output file stream and writes to a PPM file. This function will return false if the number of rows to write is 0, or true otherwise.

4. A public function that flips the PPM vector around the Y axis. The resulting PPM file will be "backwards".
5. A public function that flips the PPM vector around the X axis. The resulting PPM file will be "upside-down".
6. A public function that inverts the intensity of each R, G, and B values in the vector. This is essentially the maximum intensity minus the given intensity for R, G, and B.

Inputs

Inputs for this program come from command line arguments. There will be either two or three arguments: `inputFilename outputFilename optionalModificationCode`

You will be reading a PPM file specified by the user in the **first** user-supplied command line argument (the input file name).

You will be making some sort of modification to the PPM file, which will be provided in a possible **third** user-supplied argument (modification code) as either Y, X, or I.

- If the user specifies Y, then you will flip the PPM file around the Y axis
- if the user specifies X, then you will flip the PPM file around the X axis.
- If the user specifies I (for intensity), you will invert the intensity of the PPM file.
- Finally, if the user doesn't specify anything, you will make no modifications to the PPM file.

You will then write to a separate PPM file specified by the user in the **second** user-supplied command line argument.

Use the following error messages for the given conditions:

If the user doesn't specify enough arguments output the following error message and return an error code:
Note: there could be either 2 or 3 user supplied command line arguments.

```
Usage: inputfile outputfile [XYI]
```

If the user specifies a third command argument, but it is not an X, Y, or I, output the following message (replace Z with whatever command the user specified), and return an error code:

```
Error: Z is an invalid command. Use either X, Y, or I.
```

If the result of reading the input file is false (either the file could not be opened or is not valid), output the following error message, and return an error code:

```
Error: unable to read PPM file 'given input file'
```

If the output file could not be opened for writing, output the following error message, and return an error code:

```
Error: unable to write PPM file 'given output file'
```

If the result of your output member function is false (meaning that the number of rows is 0), output the following error message, and return an error code:

```
Error: tried to output an uninitialized PPM picture
```

Output

The PPM file that you output will have the following format:

```
P3
width height
intensity
```

```
pixel_row0_col0_red pixel_row0_col0_green pixel_row0_col0_blue
pixel_row0_col1_red pixel_row0_col1_green pixel_row0_col1_blue
...
```

In other words, you will output the red, green, and blue values of each pixel on a single line. You will output an entire row first (meaning output each column of one row before moving to the next row).

Problem Solving

You have been given what your program will be doing, however in this lab, you will be fully developing your solution to solve it. Use Code Assessor to test your code and determine what your code should do in certain circumstances.

C++ Topics Covered

- Vectors
- Input file streams
- Output file streams
- Classes/structures

Textbook Chapters Covered

- Chapter 8.3 (Vectors)
- Chapter 10.2 (Constructors)

Relevant Reading