Assignment 3 - Color to Grayscale

Due Feb 14 by 11:59pm **Points** 20 **Submitting** a file upload **File Types** h and cpp **Available** until Feb 17 at 12:01am

This assignment was locked Feb 17 at 12:01am.

ST. CHARLES COMMUNITY COLLEGE

CPT-182 - Programming in C++

Programming Assignment - Color to Grayscale (**20** Points)

(Number in Question Bank: Assignment **3.1**)

Program Overview

In this assignment, you are going to write a program that converts a color image to grayscale image. Your program reads in an image in **PPM** format, converts each pixel's color data to grayscale, then writes them to an output PPM image file.

Program Input

Your program reads in a PPM image file as input (filename: **`color_image.ppm`**). PPM is an image format which uses text to represent images. Please see an example below:

```
P3  // This is a "magic number" to indicate that the file is a PPM image.
800 600  // It means that in this image, there are 800 pixels per row and 600 rows.
255  // This is the color depth, which means each color value is between 0 and 255.
169  // Red value of the first pixel
157  // Green value of the first pixel
135  // Blue value of the first pixel
170  // Red value of the second pixel
152  // Green value of the second pixel
132  // Blue value of the second pixel
...
...
...
```

In PPM images, each pixel's color is the combination of **3** values, red, green, and blue. Each value is an integer between **0** and **255**. This way to represent the color of a pixel is called **RGB Color**.

In the first line, the **"P3"** is a "magic constant" identifying this file as a PPM image. The second line contains **2** integers, which represent the dimensions of the image. In the example above, the image has **800** pixels per row and **600** rows. The third line contains a single integer, which represents the color depth. In the example above, the red, green, and blue values for each pixel range from **0** to **255**.

Starting from the next line, data of the colors of pixels are listed, which is in the row-major order: red value of the first pixel, green value of the first pixel, blue value of the first pixel, red value of the second pixel, green value of the second pixel, blue value of the second pixel, and so on. All values are separated by whitespaces. Please note that the PPM standard does **not** require line breaks any place in particular (or at all); spaces, tabs, or newlines, in any combination, can separate items. In this assignment, the input image uses line breaks to separate data, one color value per line.

In this assignment, you **cannot** assume (or guess) the dimensions of the input image. In other words, no matter it is a **50**-by-**50** image or **2560**-by-**1440** image, your program should correctly process it and generate correct output image.

Program Output

The output of the program is a PPM image (filename: **`grayscale_image.ppm`**) with the same dimensions as the input image.

The first line of the output file is **"P3"**, the "magic constant". The second line of the output file contains **2** integers, which are the dimensions of the image and should be the same as the input file. The third line of the output file is **"255"**, which is constant in this assignment.

Starting from the next line, the output file lists the result color data for each pixel, one number per line.

How to Convert a Pixel's Color to Grayscale?

In this assignment, if **red** = **x**, **green** = **y**, and **blue** = **z**, then the **red**, **green**, and **blue** values after the conversion should be:

red = **green** = **blue** = **0.3x** + **0.59y** + **0.11z**

For example, the RGB values for color "🟩" is (**60**, **200**, **100**). After the conversion, the RGB values should become (**147**, **147**, **147**), which represents color "⬛".

Please note if the calculation result is **not** an integer, you need to convert it to an integer.

Sample Input and Output Files (Click the Filenames to Download)

Sample Input File 1 (https://drive.google.com/uc?export=download&id=1XST-g3Nt_3QzADreB90yW5i5S9FyaNWK)

Sample Output File 1 (https://drive.google.com/uc?export=download&id=16FI4xZBI7bZnUEnyOW3Sxq20gZYczFVT)

Sample Input File 2 (https://drive.google.com/uc?export=download&id=1iEWPeo-gt2TjtNRH_iNtWqwEYlByKjdw)

Sample Output File 2 (https://drive.google.com/u export=download&id=1N5ai eFnT25Ys2FsvWOjgd8ZePlM

Online PPM Image Viewer

Unfortunately, Windows operating systems **cannot** open PPM images directly. The link below is an online PPM image viewer. You can use it to see your PPM files as images.

http://paulcuth.me.uk/netpbm-viewer (http://paulcuth.me.uk/netpbm-viewer)

Assignment Submission and Grading (Please Read)

- Please upload all your **.h** (if any) and **.cpp** files (**not** the ~~entire Microsoft Visual Studio project folder~~) on Canvas.
- Before the assignment deadline, you can submit your work unlimited times. However, only your latest submission will be graded.
- At least **20**% of your code should be **comments**. All variable, function (if any), and class (if any) names should "make good sense". You should let the grader put **least effort** to understand your code. Grader will **take off points**, even if your program passes all test cases, if he/she has to put extra **unnecessary** effort to understand your code.
- Please **save a backup copy** of all your work in your computer hard drive.
- Your program will be graded (tested) using another valid input file (still named **`triangles.txt`**) to check whether it can generate the expected (correct) output file (with correct format and correct output values in it). As long as the input file is valid, your program should generate a correct output file. In other words, your program should work for **any** valid input file, **not** just the sample input files provided in the assignment instructions.
- In this class, you can assume that the input file (input data) is always **valid** and **has correct format**. You do **not** need to deal with ~~invalid input~~ or ~~error handling~~.
- Your work will be graded after the assignment deadline. All students will receive their assignment grades at (almost) the same time.