Project 1

You are asked to write five programs, all related to the manipulation of color in digital images.

First program

Write a program that displays continuous changes in color for the xy and the Luv representations. The input to the program is a width and a height. The output is two images of dimensions width $\times$ height that are displayed on the screen.
For the xy image, the pixel at row i and column j should have the color value:

$$x = j/\text{width}, \qquad y = i/\text{height}, \qquad Y = 1$$

For the Luv image, the pixel at row i and column j should have the color value:

$$L = 90, \qquad u = 512 * j/\text{width} - 255, \qquad v = 512 * i/\text{height} - 255$$

The main programming effort is writing the routines to convert xyY and Luv pixels to $sRGB$. The provided example program **proj1a.cpp** does everything with the exception of this conversion. It is recommended that you write your program by changing proj1a.cpp so that it fulfills the requirements.

Second, third, and fourth program

These programs change the color of the image based on a histogram computed from a window in the image. The window is specified in terms of the normalized coordinates w1 h1 w2 h2, where the window upper left point is (w1,h1), and its lower right point is (w2,h2). For example, w1=0,h1=0,w2=1,h2=1 is the entire image, and w1=0.3,h1=0.3,w2=0.7,h2=0.7 is is window in the center of the image. The provided example program **proj1b.cpp** shows how to go over the pixels of this window.

Second program

Write a program that gets as input a color image, performs linear scaling in the Luv domain, and writes the scaled image as output. The scaling in Luv should stretch only the luminance values. You are asked to apply linear scaling that would map the smallest L value **in the specified window** and all values below it to 0, and the largest L value **in the specified window** and all values above it to 100.

Third program

Write a program that gets as input a color image, performs histogram equalization in the Luv domain, and writes the scaled image as output. Histogram equalization in Luv is applied to the luminance values, as computed **in the specified window**. It requires a discretization step, where the real-valued L is discretized into 101 values.

As in the second program, all L values below the smallest L value in the window should be mapped to 0, and all L value above the largest L value in the window should be mapped to 100.

Fourth program

This is the same as the second program, except that the scaling is to be performed in the xyY domain. The scaling should stretch only the luminance (Y) values. You are asked to apply linear scaling that would map the smallest Y value **in the specified window** and all values below it to 0, and the largest Y value **in the specified window** and all values above it to 1.

Fifth program

This should be identical to the second program, but the color conversion should be performed using the **cvtColor** routine of OpenCV. See **ColorConversion.cpp** for an example of using cvtColor.

Evaluation

We will test your programs on several images. These will include color image that should be "improved" by the scaling, and artificial image with very few pixels. The results of running your program on the artificial image will be printed and compared to the correct values.

1 What you need to submit

Submit a report and source code/files in machine readable form. Make sure that you are supplying all the files that your program uses.
Please notice that it is your responsibility to provide us with all the above information. We will NOT go to your files to look for programs that can be complied to create the necessary executables.

1.1 Submitted material

- Source code for your program.
- Explanation of any decision that you have made, which is not a simple application of a formula. (For example, how do you handle division by 0 in your program, and other out of range values.)
- Describe the results that you obtain in applying your program. Can you find situations in which your program makes a picture look "bad"? Give an example of a such a picture.

1.1.1 Description of the machine-readable files

- A description of the relevant files.

1.2 Information in machine readable form

- Source code and executable files for all the programs.

Testing your programs

We are going to run your programs and evaluate how they work. You must be present when your program is tested.

Due Date: TBA