Assignment 11

01/24/2017

A very simple image file format is the Truevision TGA (or TARGA) format. You are to write a program that reads a TARGA file and outputs information about the image stored in that file. The information includes a luminance histogram (see below).
A detailed description of the file format is available on the Wikipedia page describing the format. An example file containing a flower image: flower.tga.
Output of your program should be similar to the following:

```
File name                       : flower.tga
Image format                    : Truevision TGA
Image size (width x height)     : 512x512
Bits per pixel                  : 32
Luminance range (min-max)       : 0.0000-0.9691
Average pixel value (r,g,b)     : (120.4731, 109.2262,
26.0517)

Luminance histogram
---------------------------------------------------------
-------------------------
       508 | ***
       935 | ******
     21360 | ****************************************
           | ****************************************
           | ******************+
     27495 | ****************************************
           | ****************************************
           | ******************+
     13716 | ****************************************
           | ****************************************
           | *********
     10750 | ****************************************
           | ******************************
      7562 | ****************************************
           | **********
      6466 | ************************************************
      5713 | ************************************
      4891 | ******************************
      4387 | ****************************
```

```
3854 | ************************
3352 | **********************
3475 | ************************
3269 | **********************
2681 | ****************
2228 | **************
2240 | **************
2354 | ***************
2370 | ***************
2641 | ****************
2493 | ****************
2603 | ****************
1729 | ***********
1601 | **********
1683 | ***********
1890 | ************
1737 | ***********
1132 | *******
 902 | ******
 886 | *****
1000 | ******
1213 | ********
1443 | *********
1588 | **********
1583 | **********
2070 | **************
2722 | ******************
3062 | **********************
3308 | ************************
3839 | ***************************
4167 | ****************************
5141 | ***********************************
6366 | ******************************************
7561 | ****************************************
     | *********
8537 | ****************************************
     | ***************
9518 | ****************************************
     | **********************
9288 | ****************************************
     | ********************
```

```
 9897 | ****************************************
      | ************************
 9759 | ****************************************
      | ************************
 8588 | ****************************************
      | ****************
 6747 | ************************************************
 3822 | *************************
 1594 | **********
  323 | **
   67 |
   18 |
   19 |
    1 |
    0 |
```

You may assume that the input file is 32 bits per pixel.
Each pixel in the file is represented by 4 bytes. The first three are the blue, green and red intensities of the pixel (in that order). The values for the three RGB components are the max and min values for a byte (0 - 255).
The luminiance of a pixel is a measure of the perceived brightness of the pixel. It is commonly computed with the following equation:

```
0.2126 * (R/255.0) + 0.7152 * (G/255.0) + 0.0722 * (B/255.0)
```

The luminance histogram is a measure of the number of pixels that have luminance within a set of ranges. In the above diagram, the range from 0 to 1.0 has been broken up into 60 “bins” (one line per bin). The first bin is the number of pixels that have a luminance between 0 and 1/60 (508). The next bin is the number of pixels that have a luminance between 1/60 and 2/60 (935), and so on. Each star to the right represents 150 pixels. To avoid wrapping, if a line will contain more than 100 stars, then it should be truncated, and a “+” symbol added to the end to indicate that it has been truncated.

Submit your program via online submission system and turn in a printed copy by due date. Thoroughly test your program.