Lab 9: Intro to PPM

CPSC 2310 – Spring 2023

CLEMSON School of COMPUTING

Due Date: Monday, April 3, 2023 @ 11:59 PM

Overview:

You are going to write a program that will read the header and pixel values of a ppm image, ignore all comments from the header then write the image back out to another ppm file minus the comments.

I will provide a file named ppmUtil.h that will contain two structs, one to represent the pixels of a ppm image and one to represent the data that makes up the header of a ppm image. The ppmUtil.h file also contains prototypes of functions you are going to implement.

I will provide multiple images for testing. Each image will have a variety of comments throughout the header. Your program should be able to ignore the comments from each of the ppm files provided. Your program should read the header information and the pixel values of the image, storing each. Then use the stored information to create a new ppm image.

You are required to use **fread and fwrite** when reading and writing the **pixel** data. You will use **fscanf and fprintf** when reading and writing the **header** information.

Below I will describe the ppmDriver.c

The driver should have minimal amount of code in it.
It should create and open the needed file pointers.
It should create a header_t and a pointer of type pixel_t.
The driver should then call the function **read** then **write**.
The driver should then call the function to give the memory back to the OS and close the file pointers.

Below I will briefly describe function parameters and what each function does. These functions should be implemented in ppmUtil.c.

pixel_t* read(FILE*, header_t*);
FILE* - Represents the input image.
header_t – represents the header data of the input image.

Calls readHeader.
Calls readPixels.
Returns the pixel_t pointer created for readPixels.

void readHeader(FILE*, header_t*);
FILE* - points to the input file.
header_t* - is used to store the information read from the input image.

This function will read in each of the 4 parts of the header of a ppm image. You should use fscanf when reading. Read one part of the header then call the function to check for comments. (We discussed in class the rules for comments in a ppm header.). If you missed the class you should come to my office or ask your TA for help.

pixel_t* readPixels(FILE*, header_t);
FILE* - points to the input file.
header_t – contains the information needed to read in the pixels for the image.

You will need to call the function allocatePixMemory to allocate the memory for the pixels.
You will then use **fread** to read the pixels storing them in the allocated memory, then return the allocated memory.

void write(FILE*, header_t, pixel_t*);
FILE* - represents the output file.
header_t - represents the header data of the output image.
pixel_t – contains the pixel data to be written.

Calls writeHeader;
Calls writePixels;

void writeHeader(FILE*, header_t);
FILE* - points to the output file.
Header_t – contains the information needed to write the header for the output image.

Use fprintf to write the header information to the output file.

void writePixels(FILE*, pixel_t*, header_t);
FILE* - points to the output file.
pixel_t* - contains the pixels read from the input image.
header_t – contains the header information from the input image.

Use **fwrite** to write the data from the input image to the output image.

pixel_t* allocatePixMemory(header_t);
header_t – contains information about the header.

Allocates the memory for the pixels of the image and returns the allocated memory.

void freeMemory(pixel_t*);
pixel_t* the memory that contains the pixel information.

Give the memory back to the OS.

void ckComment(FILE*);
FILE* - points to the input file.

Checks for and ignores the comments in the ppm header.

Submission Information:

Submit ppmUtil.h, ppmUtil.c, and ppmDriver.c to the lab9 handin bucket. Please don't submit the ppm images to handin.