Programming Assignment 5:
Image Files
Due Tuesday, October 10th at 11:59 pm

1. Overview

The purpose of this assignment is to give you experience (or more experience) with:

- designing algorithms
- implementing functions
- nested for loops
- multi-file programs and their Makefiles
- file i/o

As in the prior project, this project is an exercise in planning (i.e., developing algorithms) as much as it is an exercise in C programming. For this assignment, you will write several small programs that produce image files in the PPM format. The format is described in a separate document.

You will be provided with several files:

- p5.c - the main driver program
- images.h - the function prototypes
- images.c - the function implementations
- Makefile - compiles it all together into p5c

The p5.c file is the main driver program. It specifies 10 colors by specifying 3 arrays of 10 integers in the range 0..255, one array for the red component, one for the green component and one for the blue component. It generates four image files by calling four different functions. For each function, the driver program:

- opens a file for output
- invokes the function, passing the file pointer and other parameters that control the size and colors for the generated images
- closes the file

The images.h file contains function prototypes for:

- make_rect -- this code is provided. It generates the PPM specification for an image of the specified size and color of pixels, and writes it to the specified file. You can use it as a simple example to get you started with implementing the other functions.
- stripes
- vert_stripes
- four_square

Your assignment for this program is to write the functions stripes, vert_stripes and four_square in images.c. These functions should conform to the function prototypes in images.h and produce output images that match those provided. If the parameters in the main program are changed (different dimensions, for example), then the function's output should adjust accordingly.

<u>The stripes function:</u>

- should generate a ppm file that contains the specified number of stripes. The color of each stripe should conform to the specified colors in the arrays passed as a parameter. That is, the first stripe should have the color associated with red_col[0], green_col[0] and blue_col[0]. The next stripe should have the color associated with red_col[1], green_col[1] and blue_col[1]. All of the stripes except the last should be of the same number of rows. The last stripe should be at least the number of rows of other stripes and should also contain additional rows as needed to generate an image of the specified size.

The vert stripes function:

- should generate a ppm file that contains the specified number of vertical stripes. The color of each vertical stripe should conform to the specified colors in the arrays passed as a parameter. That is, the first stripe should have the color associated with red_col[0], green_col[0] and blue_col[0]. The next stripe should have the color associated with red_col[1], green_col[1] and blue_col[1]. All of the stripes except the rightmost should be of the same number of columns. The last stripe should be at least the number of columsn of other stripes and should also contain additional columns as needed to generate an image of the specified size.

The four square function:

- should generate a ppm file that contains four quadrants, numbered as seen below. The color of each quadrant should conform to the specified colors in the arrays passed as a parameter. That is, the upper left quadrant should have the color associated with red_col[0], green_col[0] and blue_col[0], and so on. If the number of columns is even, then quadrants 0 and 1 will have the same width. If odd, then quadrant 1 will be one pixel wider than quadrant 0 (and the same for 2 and 3). Similarly, if the number of rows is even, then quadrant 0 and quadrant 2 should have the same number of rows. If the number of rows is odd, then quadrant 2 will have one more row than quadrant 0(and the same for 1 and 3).

0	1
2	3

Begin by implementing the stripes function. Use pencil and paper to plan it out. Sketch out the format first, then plan your algorithm. You'll need to add variables (use meaningful variable names), use nested loops, and access the RGB values from the arrays defined in the skeleton files.

Notes:

- the PPM format that we'll be using (mode 6) consists of a header followed by a stream of characters – no line breaks.
- Be sure to consider rectangles as well as squares.
-

2. Sample output:

See the provided sample outputs (the ppm files), as produced by the provided p5.c program. Be aware that we will test with both the provided p5.c program and with other versions of p5 that call the functions with different parameters. Your functions should make use of the parameters rather than hard-coding colors, number of rows, or number of columns.

3. What to Turn In

Via Gradescope, submit images.c containing the four functions (one that was provided and three that you implement). Your program should compile without warnings. Do **not** submit any extraneous files (no executables, no image files).

Your code will be evaluated not only on its correctness but also on its adherence to the coding style standards. Remember to use meaningful variable names, and to indent and comment properly.

4. Notes on Collaboration

You are required to work individually on this assignment. **Please do not consult *anyone* other than your instructors on *any* aspect of this assignment**. If you submit questions via the class piazza page, please don't include significant source code.

Review the section on "Academic Integrity" on the Course Syllabus for the policy on what happens when copying or sharing code.

For any of the programming assignments in lecture and/or lab, it is considered cheating to do any of the following:

- discuss in detail the C code in your program with another student
- show or share the C code in your program with another student
- use C code obtained from another student, or any other unauthorized source, either modified or unmodified (each student is responsible for protecting his or her files from access by others)
- use re-engineering tools
- submit work of others, from the Internet or any other source, even if attributing the work to others

5. Grading Rubric

Functionality

generates correct image files, out of range values not accepted, user prompted to enter appropriate values

____ /15	user_choice.c implementation including range-checking of command line parameters
____ / 25	stripes.c implementation including range-checking of command line parameters
____ / 30	vert_stripes.c implementation including range-checking of command line parameters

Style and format

____/ 20	“clean” and readable code, standard formatting and indentation

Submission details

____ / 5	no extraneous files
____ / 5	clean compile