CSE1PES: PROGRAMMING FOR ENGINEERS AND SCIENTISTS

ASSIGNMENT 3

Assignment developed by Matthew Felicetti 2016, updated 2017(MF), updated 2018(MF & JM), updated 2019(MF)

TOOLS

- Unix server through Putty for compiling and testing
- Notepad ++ for writing solutions
- LMS for submission

SUBMISSION

- **The code**
  - **Submitted as a .c file in the normal assignment submission portal. (Do not submit the executable.)**

ACADEMIC INTEGRITY

Plagiarism is the submission of somebody else's work in a manner that gives the impression that the work is your own. For individual assignments, plagiarism includes the case where two or more students work collaboratively on the assignment. The School of Engineering and Mathematics treats plagiarism very seriously. When it is detected, penalties are strictly imposed.
http://www.latrobe.edu.au/students/academic-integrity

IMAGE PROCESSING

BACKGROUND

This program will perform different operations on images saved as Bitmaps. (extension .bmp)

The output files for this assignment are bmp, and hence you will need to open the results in windows or on macOS using a suitable image editor.

If you want to add your own images you can but they must be 24-bit Bitmaps, and the width of the image*3 must be divisible by 4

PROBLEM

The program will perform the following different operations on images:

Print image information – Print a summary of the image properties

Save a copy of an image – Load and save a copy of an image

Change Luminosity – Make image lighter or darker

Remove channel - Remove either the red, blue or green components of an image

Quantize – Reduce the number of colours in an image

Invert – Invert all the colours in an image

Flip Horizontally – Flip the image horizontally

Crop – Cut a section out of the image to make a new image

Please note this is not the ideal solution, but one designed to test your understanding of various topics.

FUNCTION PROTOTYPES AND INCLUDES

1. You must include the stdio.h, stdlib.h and string.h libraries
2. Add the following macro

```
#define MAX_FILE_NAME_SIZE 100
```

3. The following structs are to be used

```
struct Pixel
{
    unsigned char red;
    unsigned char green;
    unsigned char blue;
};

struct RGB_Image
{
    char file_name[MAX_FILE_NAME_SIZE];
    int height;
    int width;
    int size;
    struct Pixel** pixels;
};
```

4. The following function prototypes are to be used:

```
//FILE FUNCTIONS AND DYNAMIC MEMORY ALLOCATION
int load_image(struct RGB_Image* image);
int save_image(struct RGB_Image image);

//FREE FUNCTION
void free_pixels(struct RGB_Image image);

//REALLOC FUNCTION
void re_allocate_pixels(struct RGB_Image* image, int new_height, int new_width);

//IMAGE FUNCTIONS
void save_copy_image();
void remove_channel_image();
void invert_image();
void quantize_image();
void flip_horizontal_image();
void change_luminosity_image();
void print_information_image();
void crop_image();

//PIXEL FUNCTIONS
void invert_pixels(struct Pixel** pixels, int height, int width);
void flip_horizontal_pixels(struct Pixel** pixels, int height, int width);
void quantize_pixels(struct Pixel** pixels, int height, int width, int quantization_level);
void remove_red_pixels(struct Pixel** pixels, int height, int width);
void remove_green_pixels(struct Pixel** pixels, int height, int width);
void remove_blue_pixels(struct Pixel** pixels, int height, int width);
void change_luminosity_pixels(struct Pixel** pixels, int height, int width, int luminosity_level);
```

   a. All are described below
   b. You **can** change the naming
   c. You **cannot** change the types

FUNCTIONS - 20

Each function is described by a different document. These documents can be found on LMS or the links below.

First four functions involve opening a file and printing the information in the file, make sure to test and ensure these are working before continuing.

FUNCTION – MAIN

FUNCTION – LOAD_IMAGE

FUNCTION – FREE_RGB_PIXELS

FUNCTION – PRINT_INFORMATION_IMAGE

The next two functions involve saving to a file, make sure to test and ensure these are working before continuing.

FUNCTION – SAVE_IMAGE

FUNCTION – SAVE_COPY_IMAGE

The next two functions are for changing the luminosity, make sure to test and ensure these are working before continuing.

FUNCTION – CHANGE_LUMINOSITY_IMAGE

FUNCTION – CHANGE_LUMINOSITY_PIXELS

The next four functions are for removing a channel, make sure to test and ensure these are working before continuing.

FUNCTION – REMOVE_CHANNEL_IMAGE

FUNCTION – REMOVE_RED_PIXELS

FUNCTION – REMOVE_GREEN_PIXELS

FUNCTION – REMOVE_BLUE_PIXELS

The next two functions are for inverting the image, make sure to test and ensure these are working before continuing.

FUNCTION – INVERT_IMAGE

FUNCTION – INVERT_PIXELS

The next two functions are for quantizing the image, make sure to test and ensure these are working before continuing. THESE FUNCTIONS ARE MORE DIFFICULT

FUNCTION – QUANTIZE_IMAGE

FUNCTION – QUANTIZE_PIXELS

The next two functions are for quantizing the image, make sure to test and ensure these are working before continuing. THESE FUNCTIONS ARE MORE DIFFICULT

FUNCTION – FLIP_HORIZONTAL_IMAGE

FUNCTION – FLIP_HORIZONTAL_PIXELS

The next two functions are for cropping the image, make sure to test and ensure these are working before continuing. **THESE FUNCTIONS ARE VERY DIFFICULT, ONLY ATTEPT IF YOU HAVE TIME AND UNDERSTANDING**

FUNCTION – CROP_IMAGE

FUNCTION – RE_ALLOCATE_PIXELS

CONSTRAINTS

- You are only allowed to include the stdio.h, stdlib.h and string.h header files. No others. Do not use conio.h
- You cannot use goto statements.
- Formatting can be changed if the user interface is user friendly
- Text to user needs to be easily understandable. You can change the text but the same inputs must be used.
- The program must print YOUR student number, name and the assignment number as specified.
- The program must print the password as specified.
- The code must be done using ANSI C and compile on latcs5.cs.latrobe.edu.au using gcc
- Code must not crash or have the potential to crash due to stack overflow or segmentation fault
- Types should be used appropriately.
- Variables should be limited to an appropriate scope
- You must use comments to explain significant lines of code. Read documentation guide
- You must use comments to explain how to use the functions and solution

HINTS

- The code should be around 400 – 600 lines of code without the bonus (without comments). If your code is **significantly** larger you may want to reconsider your approach.

SUBMISSION

Part 1. Solve the problem by implementing a program using C code.

EXAMPLES

Given on LMS, assignment tab

RUBRIC

Automatic Deductions

- Use of goto — minus 40 marks
- No student number — minus 100 marks
- No password — minus 100 marks
- Includes header files outside of stdio.h, stdlib.h and string.h — minus 60 marks

main – 5 points				
Main menu exits, selects function and repeats correctly	0 Does not compile or not attempted.	1 Many or major mistakes	2 Some mistakes	2 Implemented correctly
Functions are called correctly	0 Does not compile, not attempted or any mistakes. (function not implemented can be commented out)			1 Implemented correctly (ALL)
Menu is user friendly	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly

load_image – 14 points				
struct RGB_Image is used correctly	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly
concat is used correctly	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly
fopen called correctly, and if file not found prints message to user and returns 1	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly
fseek called correctly	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly
header information is read and is stored appropriately	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly
Dynamic memory is created correctly	0 Does not compile or not attempted.	1 Many or major mistakes	2 Some mistakes	5 Implemented correctly
Pixels are read and stored correctly	0 Does not compile or not attempted.	1 Many or major mistakes	2 Some mistakes	3 Implemented correctly
File is closed correctly	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly

free_pixels – 2 points				
rows are freed correctly	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly
Row pointers are freed correctly	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly

print_information_image – 4 points				
load_image called correctly	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly
Information printed correctly	0 Does not compile or not attempted.	1 Many or major mistakes	2 Some mistakes	3 Implemented correctly regardless of output.

save_image – 10 points				
strcpy is used correctly	0 Does not compile, not attempted or any mistakes.			2 Implemented correctly
fopen called correctly for writing	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly
Header information correct	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly
Pixels are written correctly to the file	0 Does not compile or not attempted.	1 Many or major mistakes	2 Some mistakes	3 Implemented correctly
File is closed correctly	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly

save_copy_image – 5 points				
load_image function called correctly	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly
save_image function called correctly	0 Does not compile, not attempted or any mistakes.			2 Implemented correctly
Image file is copied correctly (check file)	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly and produces output
Memory is freed	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly

change_luminosity_image , change_luminosity_pixels – 8 points				
Luminosity level scanned from user correctly	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly
Concatenation of file name performed correctly	0 Does not compile, not attempted or any mistakes.			2 Implemented correctly
change_luminosity_pixels called correctly	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly
Luminosity changed correctly, positive, negative, limits (check file)	0 Does not compile or not attempted.	1 Many or major mistakes	2 Some mistakes	4 Implemented correctly and efficiently and produces output

remove_channel_image, remove_red_pixels, remove_green_pixels, remove_blue_pixels – 7 points				
Menu implemented correctly	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly
Concatenation of file name performed correctly	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly
The three pixel functions are called correctly	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly
Colour is removed from image (check file)	0 Does not compile or not attempted.	1 Many or major mistakes	2 Some mistakes	4 Implemented correctly and efficiently and produces output

invert_image, invert_pixels – 5 points				
Invert_image function is implemented correctly	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly
Pixels are inverted correctly (check file)	0 Does not compile or not attempted.	1 Many or major mistakes	2 Some mistakes	4 Implemented correctly without unnecessary loops and produces output

quantize_image, quantize_pixels – 6 points				
Menu implemented correctly	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly
Concatenation of file name performed correctly	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly and efficiently.
Images are quantized correctly (check file)	0 Does not compile or not attempted.	1 Many or major mistakes	2 Implemented correctly	4 Implemented correctly without unnecessary loops and produces output

flip_horizontal_pixels, flip_horizontal_image– 8 points				
Images are flipped correctly (check file)	0 Does not compile or not attempted.	2 Many or major mistakes	6 Implemented correctly	8 Implemented correctly without unnecessary loops

crop_image, re_allocate_pixels – 7 points				
Realloc used correctly	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly. The pointer to the memory is used appropriately. Do not use the same pointer as a parameter and store in what is returned.
Members updated correctly – including file size, width and height	0 Does not compile, not attempted or any mistakes.			1 Implemented correctly
All memory freed correctly with no memory leaks.	0 Does not compile, not attempted or any mistakes.			4 100% - Implemented correctly
Images is cropped correctly (check file)	0 Does not compile or not attempted.			1 Implemented correctly and produces output

	All - 20 points			
Is the code correctly indented?	0 The code is not indented or very poorly indented, or lack of attempt	1 Several errors in indenting.		2 Code is indented appropriately.
Are variables named appropriately?	0 Variables are named poorly, or lack of attempt. Any single letter variables, excluding i,j or n used appropriately.	1 Variables are named suitably for the code.		3 Variables are named suitably for the code and the domain.
Are variables limited to an appropriate scope? And function signatures have not been modified.	0 Any globals used, or lack of attempt. Functions modified	1 Variables are not limited to an appropriate scope.		3 Variables are limited to an appropriate scope.
Are appropriate variable types used where appropriate? And function signatures have not been modified.	0 All variables are declared as the largest types, or lack of attempt. Functions modified.	1 Some types are not suitable.		2 All variable types are suitable.
Is there no unsafe behavior	0 Unsafe behavior, or lack of attempt			5 Code does not crash or have the potential to crash due to stack overflow or segmentation fault
Comment at the top of the code explain implementation and purpose. And suitable comments throughout the code.	0 No comments	2 Poor comments	3 Reasonable comments	5 Excellent documentation. Program description. Program purpose. Function descriptions. Units. Pixel operations explained clearly.