

# TCSS333

## C for System Programming

### Programming Assignment 4

### Image Manipulation without Arrays

DUE: See Canvas Programming Assignment 4 link.

You are to submit a single .c file to Canvas and nothing else (do not zip anything, no Eclipse projects, etc.)

The concepts involved with manipulating bitmap files is already known by you. This assignment has you combine that knowledge with the use of pointers, pointer arithmetic, passing pointers to functions to use indirection (change the data) and double indirection (change the address to where the pointer points), memory allocation, and the free of previously allocated memory.

This assignment requires you to manipulate and combine the data that is stored in two image files and create two new image files based on the instructions seen below. As you learned in Assignment 2, there are several types of image file formats (bmp, Jpeg, tiff, gif, png).

As with Assignment 2, you will be using 24-bit color with the width and height of the images as always multiples of 8. However, your program should work with images of various sizes (still multiples of 8) without having to recompile. Several examples of image files will be posted on Canvas. Your goal is to read two image files and from them create two new image files:

- A double exposure of the two input files or images (**you should name this output file “blend.bmp”**)
- An 8x8 checkerboard pattern of the two input files or images (**you should name this output file “checker.bmp”**)

Your program will ONLY use the 2 files you enter as command line parameters, i.e. when the program name is typed at the command line, you will include two filenames. YOUR PROGRAM SHOULD NOT PROMPT THE USER FOR FILE NAMES as some of you did on Assign. 2. The file names can be anything and a filename entered at the command may not even exist. Their existence, size, and bitmap type will be determined by your code, i.e.

If either file does not exist or contains no data, terminate the program (ONLY through main NOT from some other function, though a function may determine an error but, should report this back to main so main can terminate).

If both files exist and do contain data, you will need to allocate memory to pointers so the entire contents of each file can be read with a fread (i.e. one fread per file).

Once read, if you discover that the files are NOT a BM format, you should free all allocated memory, display an error message, and terminate the program.

If you need more details on bitmap files, refer to the links contained in Assignment 2.

As mentioned above, you will only use pointers and dynamically allocated memory to complete this project always keeping in mind:

Absolutely no indexing (square brackets [ ]) should appear ANYWHERE in your code:

All image data will be accessed via pointers to dynamically allocated memory, pointer arithmetic will be applied to move from one byte (or group of bytes) to the next or calculations on the pointer to access specific data at some known offset. Of course, pointers to non-allocated data may and will most likely need to apply.

You will be passing pointers or the address of the pointer to functions in order to change the data to which the pointer address (indirection) and/or change the address to which the pointer references (addresses), respectively.

Decompose your solution keeping in mind (again), this means you will at times be passing pointers to functions in order to update data and sometimes the address of the pointer to update the address to which the pointer references.

No prompts for input should appear.

Any files (file types) may be entered at the command prompt, even more or less than 2!

If the input files are valid BM files (these will always have rows and columns multiples of 8) the program will generate two output files with the names “**blend.bmp**” and “**checker.bmp**” based on the same requirements of how these are created from Assignment 2. However, this time without the use of any arrays or multiple calls to fread or fwrite.

Even the second parameter coming into main cannot be an array (should not see square brackets [ ]).

There should not be any more than 2 fread and 2 fwrite in your entire program.

Whatever memory you allocate should ALWAYS be deallocated.

More so than previous assignments, your program will also be graded on good coding practices such as documentation, decomposition, indentation, meaningful identifiers, proper spacing, etc.