➢ Background and Motivation

C is a simple language that follows the structured approach to programming. The language was created way before most of the current popular languages, which have borrowed many concepts from the C programming language. C enables you to work closely with the hardware, making it a top choice for system programing. Thanks to its low-level programming abilities, it is also ideal for developing operating systems and compilers. In addition, C is popular for its flexible memory management. Programmers have the opportunity to control how, when, and where to allocate and deallocate memory.

This assignment will give you the opportunity to learn C by using it to perform some low-level operations related to image file manipulation, getting down to the bits and bytes of the problem. You will learn how to approach and break down a problem in a structured way and build the code that mirrors that structure. You will use the control flow mechanisms that we discussed, and will need to work with pointers, arrays, memory allocation and addresses in order to complete the program.

You should become accustomed to reading through technical specifications and ensuring that you are clear on the technical details before beginning a project. This is a very useful skill and it is especially important when dealing with low-level designs and protocols.

➢ Requirements

Your assignment is to implement a C program that will horizontally and/or vertically flip an image. Your program will read an image file into memory, operate on the data, and write the data of the flipped image to a new file. For those who are spatially challenged, flipping means:

input image

horizontal flip

vertical flip

horizontal and vertical flip

Your C program must compile and link and produce a command-line executable, which a user will be able to run from the command line like so:

```
$:> imageflip H myimage.bmp
```

Bitmap (BMP) Images

There are various different image file formats. Each format encodes the image in a different way. **For this assignment, we are only going to handle BMP files.** In order to process the data properly, you will need to understand the format of the BMP file. In particular, you will need to know how to access the header of the file and the pixel array, which contains the actual image data. Here are some links that you can start with. You are encouraged to research further, if necessary.

- BMP file format - Wikipedia *(specifically this image)*
- A Beginners Guide to Bitmaps
- The BMP file format

Information about the file format and particularly the header file is critical to understanding what you need to do.

BMP Color Depth

While we are only dealing bitmap files for this assignment, we are going to need to handle files of different color depths. In general, BMP files are characterized by two parameters: the number of pixels, and the color depth per pixel, which is effectively the number of bits required to represent the data for one pixel. The greater number of bits needed to represent a pixel, the larger the image will be.

As you will read, the color depth can range from 1 bit per pixel up to 32 bits per pixel. The standard color bitmap files are 24-bit BMP files (see Windows Paint). The standard grayscale image is 256 colors, which is 8 bits. **For this assignment you will not need to handle a bit depth of less than 8. You need only handle files with a bit depth of 8 bits or more.** These will be files with bit depths of 8, 16, 24 and 32. You can use this handy website to convert images to different bit depths. *(Note, on that site you will see formats with RGBA in them. I will be testing all the formats there. In the repo, I will provide test images for each.)*

32 (True color, RGBA)
32 (True color, RGB)
24 (True color)
16 (5:5:5:1, RGBA Hi color)
16 (5:5:5:1, RGB Hi color)
16 (5:6:5, RGB Hi color)
8 (Indexed)

The Output File

Take note, you are only flipping the image, which means that you are only going to be changing information that relates to the image data. Since you are not going to be changing the size, shape or format of the file, you should note that the header information is not going to change and will be the same in both your input file and output file. (As opposed to, for example, if you were rotating the image. Think about it …)

➢ Homework Specifics

For this homework we will provide you with the scaffolding of your C program. We have defined data types and functions which should help guide you toward a solution. However, you will need to read and understand the BMP file specs in order to properly fill in the missing information.

Begin by downloading the code from the class repository.

Make sure you can build and run the code, in Linux and on Windows. My recommendation is to use CLion from JetBrains. I have included the make file that the CLion project uses, so you should be able to download it to a clean directory, right-click, and open folder as a CLion project. The first time you open the project, it will create a default Debug configuration which you can accept. On Linux the command line GCC compiler should work like so:

```
sack@HP7-SACK:/git/COM3640_Fall22/Imageflip$ gcc -lm *.c -o imageflip
sack@HP7-SACK:/git/COM3640_Fall22/Imageflip$ ./imageflip
Hello world sanity check
```

1. The specific requirements regarding command line inputs, error codes to report, and messages to print to the screen are all detailed in the `main.c` file in the comments.

2. The basic structure to follow is provided in TODO comments. C is structural in that you break down problems and encapsulate functionality within subfunctions. Feel free to create any other functions that you like that can be called from either within the main function, or from any other function.

3. Your program should be able to build and run on both Windows and Linux. (i.e., if it is built on Windows, it will run on Windows; if it is built on Linux, it will run on Linux).

4. We will upload sample BMP files to the repo which you can use to test your program.

5. As a suggestion, get your program working to the point where you have performed all the command line checks, are able to read the input file and write the *unmodified* data to the output file – i.e., your output file is a copy of the input file. Once that is working, tackle the data transformation part. (Use a function to abstract out the transformation!)

COM-3640 Programming Languages
Homework #3: Flipping an Image in C
Due Tue Nov 8 11:59 PM

General HW Submission Instructions

- All submitted assignments must be checked into the YU GitHub system.

- Say that your Git url is https://github.com/Yeshiva-University-CS/SmithBob. When you *git clone* that url to your computer, the result is a directory named SmithBob. That root (Git) directory will be referred to as **$MYGIT**.

- All of your submitted assignments must be rooted in a directory named: **$MYGIT/PL-COM3640/assignments**.

- Each assignment will be associated with a specific subdirectory. We will refer to that directory name as **$DIR.**

- Therefore, all submissions for a given assignment will be checked into and rooted at:
  $MYGIT/PL-COM3640/assignments/$DIR

This HW's Submission

- The subdirectory for this assignment is **Imageflip**.
- The assignment must be checked into: **$MYGIT/PL-COM3640/assignments/Imageflip**
- Unless you decide to include more .c or .h files, your submission should be the exact same files that you checked out.