Name:	

Assignment 1

List your lab time: 1:30, 2:30, 3:30, 5:30

Due: Saturday, February 15, 2020 at 11:59 midnight.

Read the entire document to ensure you are aware of all requirements. I will not accept the excuse you did not know you were supposed to do something.

This assignment consists of two parts. The first being hand written, with the second involving programming.

# Part 1: 15% of the assignment Written

Consider the following function. This function demonstrates how using the ^ (XOR) operator we can swap two numbers without using a temporary variable. There is not a real advantage of this method over the method that uses a temporary variable. This method simply demonstrates an interesting way to swap two numbers. This function can use any type of data. I chose uint8\_t because I wanted to limit the data to 1 byte (8 bits). Also, since this is unsigned the values should be limited to 0 - 255.

In the written portion of this assignment you will convince yourself that the function inplace\_swap does indeed successfully swap the values of "a" and "b". By using the 8 bit binary representation of "a" and "b", show that **after** the function is executed "a" will equal 20 and "b" will equal 25. At each step show the value of what the pointer is pointing to (a and b) in binary, hexadecimal and decimal.

Step 1
$$*y = *x ^ *y$$
 (show your work)

After Step 1

·	Binary	Hex	Decima I
* <b>y</b>			
*x			

Step 2
$$*x = *x ^ *y \text{ (show your work)}$$

After Step 2

	Binary	Hex	Decima I
* <b>y</b>			
*x			

Step 3
$$*y = *x ^ *y$$
 (show your work)

After Step 3

	Binary	Hex	Decima I
* <b>y</b>			
*x			

# Part 2: 85% of the assignment Programming

You are to write a **C** program that will call the inplace\_swap function and then print the Binary, Hexadecimal, and Decimal values of **a** and **b** in a table format. I am allowing you the opportunity to use creativity with the format of the table. However, at a minimal you must separate the output data and the column and row labels with vertical/horizontal lines. You must have the variables a and b (the rows) on the side and Binary, Hex, and Decimal (the columns) at the top, similar to the tables listed above.

The C programming language does not provide a mechanism to print the binary representation of a number, therefore you must write a function that will print the binary representation of a given number. There are a multitude of ways to write this type of function, for example you have probably, in the past, used the division (/) and mod (%) operators to write this type of function. For this assignment I am requiring you to use at least one of the bitwise operators (&, |,  $^$ , <<, >>,  $\sim$ ).

#### Other Requirements:

You will need to provide a **driver.c** file that will contain the main. Your driver should have minimial amount of code in it. Your program should use command line arguments to determine the value of "a" and "b". You should also create a function that checks that the appropriate number of command line arguments are passed to the program. If the appropriate number of command line arguments are not passed inform the user they did not enter the correct number of command line arguments and exit the program. You should have a function that ensures the value passed on the command line is within the appropriate limits of an uint8 t. The prototypes for your functions must be located in a file called **functions.h.** Your **functions.h** file must include header quards. The implementation of the functions must be in functions.c. You must also submit a makefile that will compile the program and for the fun of it, use the flag necessary to save the temporary files created during the compilation process. Your makefile should include a make clean that will remove the executable and all temporary files during the compile process.

## Formatting:

With the exception of the the makefile, you should have a header in each of your files that contains the following information.

\*

- \*Your name \*
- \*CPSC 2310 Spring 2020.
- \*Due Date:

#### Academic Integrity:

This assignment is an individual assignment. You are not allowed to obtain help from anyone other than myself or your lab TA's. You are NOT allowed to receive help from any other tutors on or off campus. You are allowed to discuss concepts with your fellow classmates but you are not to share code nor verbally dictate how to solve the assignments. If it is determined that you violated any of the above it will be considered a violation of academic integrity and handled as such.

#### Submission:

You are required to tar.gz all documents and submit them through handin. Let me remind you, if you neglect to validate that you tar.gz'ed your files correctly or there are files missing you will receive a 0 on the assignment. While it is good practice to create a folder to hold all files for this assignment, you are NOT to tar.gz the entire folder. You should change directory into the folder and then tar.gz the content. What you name the tar.gz file is up to you as long as it is a tar.gz file. Also, after you submit through handin you should download the tar.gz'ed file and open it to ensure nothing happened during this process. This could take time so don't wait until the last minute to submit. Late submissions will not be accepted.

Listed below are some things that will cause you a reduction of points on this assignment. This list is not an exhaustive list, simply some items off the top of my head I will be looking for when grading. Anything in the writeup that says you should or you must is a good indication I will deduct points if you do not do them.

- Your program should compile with no warning and no errors. There could be up to a 60-point deduction if your program has compile errors.
- You should use meaningful variable names throughout the program.
- Your code should be well documented. (comments see example below)
- There should be no line of code longer than 80 characters.
- You should use proper and consistent indention.
- You must test your program on the School of Computing servers prior to submitting your work. I will test your program on one of the SoC servers. I will not accept "But it runs on my computer."

#### **Documentation Guideline:**

Here are guidelines for documenting the code in this assignment.

Before each function, in the functions.h file, you should have a detailed description of what the overall function does.

#### As an example:

```
/* Parameters: img - image_t pointer array holding the image data for

* each of the input files

* Return: output - image_t struct containing output image data

* This function averages every pixels rbg values from each of the

* input images and puts those averages into a single output image

*/
```

You are required to have this type of comment block before each function.

Also, if you include comments in the body of the function or in your main (and you should) they should be placed above the line of code not beside the code.

### Example:

```
Good
//This is a
comment
if(something)
{
    do
something;
}
```

```
Bad
if(something) //This is a
comment
{
    do something;
}
```