EEET2250 – Week 3 Lab Task (due Week 5)
File I/O and Bitwise Operations

Total possible marks: 15 marks (3% of final mark)

Note: Although these tasks are due after the lab test, they are designed to be done prior.

Aim:
The Lab Tasks are aimed at helping you to build up code for the Lab Test - the Week 3 Lab Task is directly related to your first Lab Test to be held during your Week 4 lab session.

The below lab tasks require you to read data from a file and use this data to interact with the OUSB board using bitwise operations. You will need to create the file to be accessed: create a text file in the same directory as your program code that contains several lines of numerical data (one number per line, between 0-15 including 0 and 15) - it doesn't matter what you name your file as you will need to enter the filename as a command line parameter (i.e., using `argc` and `argv` in `main(...)`) into your program.

You will be required to write a program that performs the following tasks:

- Open a file stream to read in your file's data, you will need to be able to read in multiple lines of data and operate on each line of data until the end of the file is reached.
- Use the data to control PORTB on the OUSB board, maintaining the current state of the top four (4) bits of PORTB (PB4-PB7).
- Perform bitwise operations on the data and write the modified data to the end of the original file (append, not overwrite!).

The tasks specified below are aimed at helping you to break down the program into smaller functional 'modules': you should attempt to code in modules, checking the functionality of each module before continuing to code (i.e., do not write the whole program at once and expect it all to work!).

Once you have finished all of the tasks, notify your tutor who will assess your work. The marking for these tasks is binary – it either works or does not.

Remember to use the lecture notes as a reference as well as the prescribed textbook.

By completing the Week 3 Lab Tasks, you would have effectively written most (if not all) of your Lab Test 1 code, except for the required output for error checking (for the autotester).

The Lab Test 1 proforma cpp file is available on EEET2250 Canvas shell, and you may wish to code the below tasks using the proforma to gain familiarity with the Lab Test 1 exercise for Week 4.

Tasks

1. When you attempt each of the tasks below, remember the six (6) steps of engineering design: draw flowcharts and write pseudocode to represent the logical flow of your program. (Hint: it might help to have a flowchart for the overall program as well as for each operation/function!

   [2 marks]

2. Enter the filename as a command line parameter (i.e., using `argc` and `argv` in `main(...)`) into your program. Remember to check that only two parameters (one being the program name) have been entered into the command line!

   Open a file stream to read in the data, and print this data onto the screen to check that the data has been read correctly. Your code will need to be able to read in multiple lines of data and print out each line until the end of the file is reached. Remember to check that the file has opened without any errors!

   [2 marks]

3. For each line of data read in from file, check that the number range is valid for use with PORTB: the data must be between 0-15 (i.e., only use the bottom four bits of PORTB) - note that 0 and 15 are valid data. You do not need to check for non-numerical data or floating point numbers, just for the correct integer range.

   Examples of valid number input: `6 0 15 9 12`
   Examples of invalid number input: `16 -4 8.54 8v +6 127 b u7`

   For any lines of invalid data, write an error message to the console and do not write the data to PORTB. Then continue reading in the next line from the data from and operate on it, i.e., keep reading in data from the next line and writing data to PORTB.

   [2 marks]

4. Write only valid integer numerical data (as determined in task (3) above) to PORTB. Consider sleeping for up to one second in between each number output so that you can see the output on PORTB changing correctly.

   [2 marks]

5. As per task (4) above, output each line of data onto PORTB of the OUSB board, sleep in-between each number for visibility if you like. However, in this task, maintain the existing value of the top four (4) bits of PORTB (hint: you will need to use a bitwise AND then OR operation to get the current value of the top four bits of PORTB and set the new PORTB value). To test this task, you will firstly need to set PORTB to a number greater than 15 (why?) then ensure that the top four (4) bits of PORTB (PB4-PB7) do not change as you output the valid numbers read in from file.

   [2 marks]

6. Perform the following three bitwise XOR operations on your student number and for the result of each bitwise operation, print the result to screen and append (not overwrite!) the result to your file:

   variable A: arbitrary integer (hint: you can ask the user for to input a number using `cin`)
   variable B: arbitrary integer (hint: you can ask the user for to input a number using `cin`)

   A = A xor B
   B = A xor B
   A = A xor B

   What operation have you performed using the above sequence of bitwise XOR? Remember to close the file stream when you have finished!

If you have time, investigate the bitwise shift operations (left/right shift): what happens when you left or right shift an integer, what mathematical operations are you performing? Why might left-shifting a negative integer give an undefined result?

[5 marks]

Once you have finished these tasks you are free to move on to working on and practise for Lab Test 1. Make sure that you download and read the Lab Test 1 proforma cpp file which is available on the EEET2250 Canvas shell. This will be the file provided to you during the week 4 Lab Test.