INTRODUCTION:

This lab is designed to give you practice working with popular sorting and searching algorithms.

Please watch the following videos before attending lab:
https://www.youtube.com/watch?v=eXeP__gBMrM
https://www.youtube.com/watch?v=vZWfKBdSgXI

LAB OBJECTIVES:

In addition to practicing concepts used in previous assignments and labs, upon completion of this lab, you should be able to:

- Sort an array of values using selection sort
- Search for an element in a sorted array using linear search

SUBMISSION INSTRUCTIONS:

Submit your files to handin (http://handin.cs.clemson.edu) by the due date announced in class as a compressed tar.gz file named username-lab11.tar.gz. You will need to submit **driver.cpp**, **functions.cpp**, **functions.h**, and **makefile**.

LAB INSTRUCTIONS:

Download the necessary starting files from blackboard. You will notice that driver.cpp contains pseudo-code to help guide you through this lab.

Task 1:

- Read data from the .txt file specified in argv[1] using command line arguments. The first item read in will indicate the number of items to read and store into an array. Read from the file and populate the array. Print the contents of the array once, then sort the array using selection sort, and then print the array again after sorting. You **MUST** implement and use the print_array and selection_sort functions for these tasks.

Task 2:

- Prompt the user to enter a number with the string "**Enter search item:** ". Then, search the array for that number using a linear search. If the number was found in the array, print out the message "**(number) found in array at index (index)**". If the number was not found in the array, print out the message "**(number) not found in array**". You **MUST** implement and use the linear_search function to search the array for the specified number.

You should also provide a makefile (lowercase m). Your makefile should create a target that will compile your program and a target that will remove the executable. To compile I should be able to just type ***make***. To remove the executable I should be able to type ***make clean***. The makefile should also provide a target called run that will run the compiled program with the input1.txt input file and then again with the input2.txt input file when the command **make run** is executed**.**

FORMATTING:

You will need to add a header to each of your files similar to the following:

```
/************************
 *Your name
 *CPSC 1021 your section, F16
 *Lab 11
 *Your username
 ************************/
```

Your program should compile without warnings and no errors.
Your code should be well documented with comments.
You should use proper and consistent indentation.
You **MUST** test your program on one of the School of Computing servers prior to submitting. Your program will be tested on the SOC lab machines.