

# CSE 235 - Assignment #1

**Total points: 100**

## Instructions:

This assignment consists of some programming exercises based on C programming concepts. Write the programs in the Linux/Unix environment with gcc compiler. After writing the codes, test the code on your Linux computer and paste the output of the programs in a docx/pdf file. Also provide the source codes (.c, .h) files of the programs. The output file and all the source codes should be put together in a .zip file (DO NOT use .RAR compression) and uploaded in the blackboard. This assignment needs to be completed individually. Discussions/collaboration with other students is prohibited. If you have questions, please reach out to the TA or the instructor.

## Questions:

1. Write a program containing three .c files, with the following filenames: “**name.c**”, “**name\_reverse.c**” and “**name\_length.c**” and two .h files: “**name\_reverse.h**” and “**name\_length.h**” and do the following.
  - a) The file “**name.c**” will contain the **main()** function. The main() will declare your name as a string variable (char pointer or char array) whose value will be your “**Firstname Lastname**”. There should be one or more spaces between the Firstname and Lastname. Note that, if you have one or more Middlename(s), do not use that. Use only two words, the firstname and the lastname, so that the complexity is even for everyone. Then it should call two functions: **reverse\_order()** and **letter\_count()** where it will pass the string variable as (char \*) in both functions. But this file will not contain the definitions of those two functions.
  - b) The file “**name\_reverse.c**” will contain the definition of the function **reverse\_order()** where the signature of the function will be declared in the “**name\_reverse.h**” file. In the definition of the **reverse\_order()** function (i.e., in the .c file) write the logic for reversing the string orders such that it converts the “**Firstname Lastname**” to “**Lastname Firstname**” and print it.
  - c) Similarly, “**name\_length.c**” will contain the definition of the function **letter\_count()** where the signature of the function will be declared in the “**name\_length.h**” file. In the definition of the **letter\_count()**, count the number of letters in your full name (i.e. firstname and lastname without spaces). Print the letter counts as the length of your name.
  - d) Write a **Makefile** for compilation of these files (three .c and two .h files) and run the ‘make’ command to compile to generate the executable. Show screenshot of compilation as well.
  - e) Run the executable and take screenshot of the output from the terminal.

**(10 + 15 + 15 + 15 + 10 = 65 points)**

2. Implement swap function for two integers a and b, i.e., swap(a,b) using macro substitution.  
**(10 points)**

3. Create an array of 10 structs having an int member and a string member. Initialize each structure with its **string** value as the name of a state in the U.S. and the **int** value will be set as the number of characters in the name of the state. For example, in one of the structures, you can populate the string with the value “Kentucky” and corresponding int value in the same structure will be 8. For 10 structures, choose the values from any 10 states of the U.S. After that do the following:

**Sort** and **search** the array by the int member value in the ascending order. For sorting, use any sorting algorithm and use pointers for iteration over the array elements.

**(25 points)**

Note: Extra credit may be given for good coding style, e.g., modularized, efficient, clean, appropriate naming and comments etc.