CS241 Lab 1

Edit and Run C Programs

1. Start a text editor such as vim (or vi) on the command line and enter the program given below (*vim lab1.c*):

```
#include <stdio.h>
#include <ctype.h>

int main () {
        char ch;

        puts ("Enter text (Ctrl-D to quit).");
        while ( ch = getchar(), ch != EOF ) {
                if ( islower(ch) )
                        ch = toupper(ch);
                else
                        ch = tolower(ch);
                putchar(ch);
        }

        return 0;
}
```

2. Save your program in file *lab1a.c* and exit the editor. Type *gcc lab1a.c* to compile your program and then *./a.out* to run its executable.

3. If there is an error in your program, make necessary changes using the editor and then compile and run your program again.

   Modify the above code in order to read in data, character by character, and then count the number of words and the number of symbol sequences in the input. Save your modified code as *lab1b.c* (*:w lab1b.c*)

   By definition, a word is a sequence of letters, digits, and underscores, such as *count*, *num1*, and *total_bal*. A symbol sequence contains only symbols (that is, non-whitespace characters other than letters, digits, or underscores), such as += and ++. Note that *num1*++ should count as one word and one symbol sequence. User's input ends with the EOF character (typing Ctrl-D from the keyboard). Some functions defined in *ctype.h* may be useful for this program, such as *isalnum(char)*, *isalpha(char)*, *isdigit(char), ispunct(char),* and *isspace(char)*. All functions in *ctype.h* and their meanings can be found from the link below:

   https://www.tutorialspoint.com/c_standard_library/ctype_h.htm

4. Compile your program and test it against data from the keyboard. Make necessary change to the program according to the testing results.

5. Use a text editor to create a text file containing testing data, such as what you entered from the keyboard earlier.

6. Run your program again with redirecting input to the program from the data file that you created, such as *lab1b < lab1b.dat*.

7. Repeat the above step with redirecting the output of your program to a text file, such as *lab1b < lab1b.dat > lab1b.out*.

8. Show your results and program to your instructor when you have finished successfully.