# Dalhousie University
# CSCI 2132 — Software Development
# Winter 2017
# Lab 9, March 30/31

In this lab, you will get practice on file manipulation in C and UNIX shell scripting. You will be asked to write two programs. Treat this as an assignment, even though sample solutions are given. Please note that you need the knowledge from next week's lectures, which haven't been given yet. However, it is better to give you these two questions now instead of waiting till the last lab, so that you can start to work on these two questions early.

Be sure to get help from teaching assistants whenever you have any questions.

1. First, perform the following steps to get started:

   (i) Login to server bluenose.cs.dal.ca via SSH from a CS Teaching Lab computer or from your own computer.

   (ii) Change your current working directory to the `csci2132` directory created in Lab 1.

   (iii) Create a subdirectory named lab9.

   (iv) Change your current working directory to this new directory.

2. This question asks you to write a C program to manipulate a text file. This is Programming Project 2 on page 584 of the C textbook (C Programming, A Modern Approach by K.N. King).

   This program converts all letters in a file to upper case, and characters that are not uppercase characters will not be changed. The name of the input file is given as a command-line argument, and the program writes its output to `stdout`.

   Hint: You may find the library function `toupper(ch)` useful (in `ctype.h`). This function converts `ch` to the corresponding upper-case letter if `ch` is an lower-case letter. Otherwise, it returns `ch` itself.

   A sample solution is given in the author's website:

   `http://knking.com/books/c2/answers/c22.html`

3. This question requires you to learn UNIX shell scripting first, which will be covered in the next two or three lectures. You can work on this question after you learn shell scripting in class.

   This question is from a book titled Linux for Programmers and Users, written by Grass and Ables. It asks you to write a shell script named `junk.sh`. The syntax of running this script is:

```
./junk.sh [-l][-p] {fileName}*
```

Here, square brackets are used to indicate that certain command-line arguments are optional.

This utility moves the list of regular files supplied in the command-line argument into a directory called `.junk` in your home directory. These arguments do not contain wildcards. For each file that does not exist, print an error message.

If `.junk` does not exist, the utility creates this directory before moving files.

When `-l` option (this is the letter l) is present, the current content of `.junk` is listed.

When `-p` option is present, the files in the `.junk` directory are removed.

These two options cannot be given at the same time.

Hint: To solve this question, the following two features of shell scripting will be useful:

(a) The predefined local variable `$@` stores a list of words which are the command-line arguments, excluding `$0`. Thus it can be used in a for loop after the keyword in.

(b) If you would like to use the `case` statement, you can use `*` as the default case that matches anything, as the pattern before `)`.

(c) When you test your program, you need know that, by default, any directory whose name starts with a dot will not be displayed when the `ls` command is invoked to list all the files in the current directory. To see these directories, use the `-a` option. Enter `man ls` for more details.

The sample solution can be found on the course web page for labs.