CS100: Software Tools & Technologies Lab I

Lab-3: Practice Problems (Shell Scripting)
Date: 24-March-2023

Instructions:

1. You must solve the below problems only on the Linux terminal to make yourself familiar with linux commands and start getting good practice.
2. Try to complete lab problems during the lab hour and submit it through canvas. If you can't complete it within the lab time, you can submit it by the end of today. There will be a penalty if you don't submit on the same day.
3. Prepare all your solution files in a zip file and name it as <ROLLNO.zip> and submit on canvas.
4. Your shell scripts files should not produce any syntactic errors.

Problems:

1. Assume that you have three files CS100.txt, CS101.txt, CS102.txt. Each file name denotes course name. Each file contains several lines in the below format
   RollNo, Grade
   RollNo has 8 digits, where each digit can be [0-9]
   Grade: Can be any of A, B, C, D, E, F

   Write a shell script for each of the following
   1. List all the roll numbers who obtained an A grade in CS102 but a B grade in CS101
   2. That list course name which has maximum number of F grades
   3. Print the roll numbers of the students who have obtained E grade in any course. Replace all E grades with a C grade for the course, which has maximum E grades in it.
      [Hint: Use **sed** command inside the shell script]

2. Write a shell script named **fib.sh** that takes a command line argument n, which computes the fibonacci number of n, i.e, fib(n)
   a. Write another shell script that executes above fib.sh for n ranging from 1000 to 2000. For each iteration i, your program should compute the time (real_time) it takes to compute fib(i) and append i, real_time **time.txt**.
      [**Hint:** Understand **time** command in linux, use real time from the output of the command]
   b. Write a shell script that reads **time.txt** and computes the average of real times of fib(1500) to fib(1700)

3. Suppose you're working in a project where software (or people) create lots of files, many of them duplicates. You don't want the duplicates: you want just one copy of each, to save disk space. Write a shell script **DelDuplicates.sh** that takes a single argument naming a directory D, finds all text files **(extension .txt)** immediately under D and removes all the duplicate files. Your script should not recursively examine all files that are in subdirectories of D; it should examine only files that are immediately in D.

   If your script finds two or more files that are duplicates, it should keep the file whose name is lexicographically first (for example, if the duplicates are named X, A, and B, it should keep A and remove X and B),

[**Hint:** You can use **diff** command to check if the two files have same content**]**