# CSE 4251: Lab 4

# Lab 4 has two exercises. Please do this lab 4 independently. You may choose to use bash script for this homework. If you are familiar with zch, csh or tcsh, you may also use zch, csh or tcsh for this homework.

**Exercise 1**

Please find the attached ***guess.sh*** file on Carmen. This script intends to implement a number guessing game. The script will initialize a random number and let a user guess that number. This exercise has the following tasks:

1. Try to execute the ***guess.sh***, you may need to correct any errors it may have. After successfully running the script without any errors, add comments in different parts of the ***guess.sh*** to explain how the script works.
2. Make a copy of the ***guess.sh*** file with the name ***guess\_game.sh*** and implement the following changes:
   1. When a user run the ***guess\_game.sh*** script, it should prompt a main menu as follows:
3. Run the guessing game
4. Exit
   1. When the user chooses the 1st option to run the game, the game will do the same logic as the ***guess.sh*** except that the ***guess\_game.sh*** keeps track of the number of times the user inputs a guess number in this game ( i.e. the number of guesses the user has tried in this game) and save the number of guesses for each game the user has played since the user executes the script ***guess\_game.sh***.
   2. When finishing the game, the script returns to the main menu, so the user can play again.
   3. Once the user chooses the 2nd option to exit, before it exits, the script should print out the average number of guesses the user has tried for all games. The output average number can be rounded to the nearest integer. An example can be:

Exiting the guessing game

Your average number of guesses is: *calculated average*

# Exercise 2

In this exercise, you may write a script which shows files and contents in the sub directories recursively in a tree-like structure like the tree command in this reference <https://manpages.ubuntu.com/manpages/trusty/man1/tree.1.html>. This script’s name is ***dirtree.sh*** and it works in a Linux file system. It has the following functions:

* 1. When a user executes the ***dirtree.sh***, it should prompt a main menu showing three options as follows:

The current directory is: /home/usr/

* + 1. Print the directory tree on terminal
    2. Print the directory tree to a file
    3. Exit
  1. The script regards the current path of this script file in the file system as the current directory and shows the current directory in the main menu in (A).
  2. When the user chooses “1. Print the directory tree on terminal”, the script will print the file system structure under the current directory in a tree-like structure. An example of the output can be found [here](https://en.wikipedia.org/wiki/Tree_(command)#/media/File:ReactOS-0.4.13_tree_command_667x434.png) (you don’t need to follow the format exactly, only need to make sure it is easy to read). After printing the output, the script should return to the main menu in (A) and the user should be able to see the main menu and choose any options from the menu again.
  3. When the user chooses “2. Print the directory tree to a file”, the script will ask the user for the name of the file and then will save the file system structure under the current directory in a tree-like structure to the file named by the user. The format of the content in the output file can be the same as the format of the output in (C). After outputting to the file, the script should return to the main menu in (A) and the user should be able to see the main menu and choose options from the menu again.
  4. When the user chooses the last option “3. Exit”, the script will exit.

# In Exercise 2, single dot “.”, double dot “..”, and hidden files whose names start with “.” don’t need to be shown in the output of (C) and (D). You may need to write a function in the script and call the function recursively in Exercise 2. For both Exercise 1 and Exercise 2, checking the exit code of the previous command in the script can be helpful. Submission Instructions

To submit, create a single zip file that contains the files from the exercise 1 and exercise 2. The zip file should have **3 files:**A modified ***guess.sh***

## *guess\_game.sh*

## *dirtree.sh*