COSC 3750, Spring 2021
Linux Programming
Homework 2, A shell script

1 Intro

This is about writing a simple shell script. This script will use the basic Bourne shell “for”, “if”, and “here-document” along with *ed* to find and modify some files. This is worth 50 points with a possible 5 points extra credit.

2 Your job

Write a Bourne shell script that does the following:

1. The script, named exactly “**modify**”, will process its arguments in order. If there are no arguments, it will print a usage message and exit. The message will be

   ```
   usage:  modify filename [filename ...]
   ```

   You will use “for” for going through the arguments, NOT “shift.”

2. As the script looks at each argument, it has to make decisions:

   - If the file is ASCII text, edit it (see below).
   - If it is a directory print the message
     modify: *filename*, is a directory.
   - If it is executable print the message
     modify: *filename*, cannot edit executables.
   - Otherwise print the message
     modify: *filename*, filetype not supported.

   These messages are each on a separate line with no extra lines between. They are not errors and the script will NOT exit, just process the next file. The *filename* is the current argument the script is processing.

2.1 Editing

The script will use *ed* to modify all the ASCII text files in the following way:

- Append to the **end** (after the last line) of the file the two lines:

  ```
  Modified on:  date_time
  Modified by:  your_name
  ```

- Interchange the first line and the second line of the file.
- Change any occurrence of ”Gradez” to ”Grades”.

2.2 Notes:

- The *ed* changes will be made in the order given.
- The *date_time* will be current date and time that the script is run. It will be exactly the same for all files edited (get it once when the script starts). The format for this will be, exactly

  ```
  Thursday, Sep 08, 2016:  0948
  ```

  That is full name of day, a comma and a single space, abbreviated month, a space, day number, a comma and space, year, colon and space, the 24-hour clock hour concatenated with the minute.

  Look at the system *date* utility and its man page.
- If *ed*'s search (**s**) command makes no matches, *ed* thinks it is an error. If there is an error, it will print **?**. If it is reading commands from a here-document or file, *ed* will then exit without saving anything. So, make the search the last thing. Write the file after the last change *before* the search. Then if the search fails and the program exits, the changes are saved.
- Figure out how, inside the script, to keep that **?** from being displayed by *ed* if there is an error. (5 points extra credit)
- The $ in a here-doc is subject to shell expansion. That means you have to escape it to make it dollar sign for *ed.*
- You will have to use something like a combination of the *file* utility and *sed* to find out if a “regular” file is ASCII text. Being ASCII text is NOT the same as “writable”.