INTRODUCTION TO SOFTWARE SYSTEMS
Assignment 1 – Shell Programming
Submission Due Date: 14 June 2021, 5 pm

Important Notes:

- This assignment is an individual submission.
- Total Marks of 100M with duration of 2 weeks.
- All script submissions should be in .sh file format.
- Inputs/output should fit the criteria mentioned in respective question.
- Evaluation will be conducted based on a fixed grading rubric (syntax, logic, input and output) and the marks are divided as per prescribed weightage in respective question.
- For queries, reach out to TAs via Moodle or Course MS Teams Group.
- Submissions should be done via moodle and also via your Github repo

Q1: *Masking the letter in a word from the input file*. Consider an input file consisting of bunch of words in a new line with word length ranging between 2 and above. Write a code to mask all the letters after first four letters in each word with # symbol as shown in example below. **(Total: 5M)**
Example:
Input: ISSStudentsAreBest
Output: ISSS###############

Q2: Consider an input file with information about a person name and his/her date of birth. Generate an output file with same person name and his/her age. **(Total: 5M)**
Example:
Input file: <name of person> <date in "dd/mm/yyyy" format>
Output file: <name of person> <age>

Q3: Consider a file with a paragraph of words. **(Total: 20M)**
Print all words in the file which satisfy following conditions:

1. Word starts with 's' and is not follow by 'a'. **(5 Marks)**
2. Word starts with 'w' and is followed by 'h' **(5 Marks)**
3. Word starts with 't' and is followed by 'h' **(5 Marks)**
4. Word starts with 'a' and is not followed by 'n' **(5 Marks)**

Example:
Input: A File with a paragraph text
Output:

1. Words - start with 's' and is not follow by 'a'
   ……
2. Word starts with 'w' and is followed by 'h'
   ……
3. Word starts with 't' and is followed by 'h'
   ……
4. Word starts with 'a' and is not followed by 'n'
   ……

Q4: Build a Search Engine on input files. (Total: 15 Marks)

Input: Consider a .txt file with bunch of sentences separated by a line break. We provide a search word as an input to search against the available .txt file.
Output: By reading the input, the code should run against these sentences and return the search score (term frequency) against each of these sentences.

Term frequency = No. of Occurrence of a given word/ Length of sentence.

Address this question by following these steps:

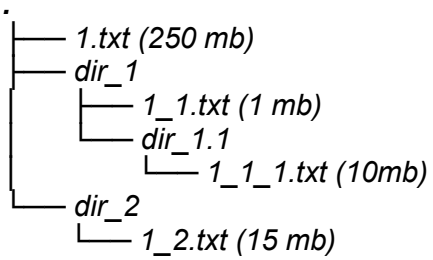
1. **Remove the noise words from the input TXT files using the supplied stopwords.txt (5 Marks)**
2. Calculate term frequency using the input search keyword against the input TXT file which is free from noise words **(5 Marks)**
3. Print the *sentence_withnostopwords, <search score>* in Output.txt file **(5 Marks)**

Q5: Write a shell script to print files and their sizes as per below criteria: **(Total: 20 Marks)**

a. Write a shell script to print all directories and files present in a folder. **(5 Marks)**
b. The directories should be printed after sorting based no. of files in the directory (descending) **(5 Marks)**
c. The files should be printed in descending order based on its size **(5 Marks)**.
d. Script should consider all the files and directories recursively and not just the files which are immediately inside the folder. **(5 Marks)**

Example:

Input: Say for a folder with structure:



Output:

$bash script.sh
> Directories:
> dir_1, 2 file(s)
> dir_1.1, 1 file(s)
> dir_2, 1 file(s)
>
> Files:
> 1.txt

> 1_2.txt
> 1_1_1.txt
> 1_1.txt

Q6: Managing Personal Contact Book. **(Total: 35 Marks)**

Write a shell script to manage your personal contact book which will keep record of your contact details. Follow the below instructions to write the shell script.

- Input: contacts.csv is the input file with fields fname (FirstName), lname (LastName), mobile (Mobile Number), Office (CompanyName).
- This contacts.csv will be updated after every insert, delete and edit. No command other than display all, or search and display should print anything on the terminal.
- No command other than display all, or search and display should print anything on the terminal.
- The output for 'search and display' and 'display all' should be coma separated words in the same sequence as the format of the csv.
- The commands should be as follows
    - Insert Contact
    - Edit Contact
    - Display all Contacts
    - Search and Display
    - Delete Contact

The system should run based on flag-based command line arguments.
A flag –C would be used to first specify the command (which would be one of 'insert', 'edit', 'display', 'search', 'delete') The flags for each command are as follows:

- Insert **(10 Marks)**
    - -f for entering first name
    - -l for entering last name
    - -n for entering contact number
    - -o for entering company name
- Edit contact **(10 Marks)**
    - -k give the first name of the contact to be edited (assume no 2 contacts have same first name)
    - -f for entering updated first name
    - -l for entering updated last name
    - -n for entering updated contact number
    - -o for entering updated company name
- Display all contacts **(5 Marks)**
    - -a to sort in ascending order of first name
    - -d to sort in descending order of first name
- Search and display **(5 Marks)**
    - -c for entering the column on which it is to be searched (has to be one of first name, last name or number)
    - -v for the search value of that column
- Delete contact **(5 Marks)**
    - -c for entering the column on which it is to be searched (has to be one of first name, last name or number)

- -v for the search value of that column of the record to be deleted

Please note that you do not have to worry about any validation checks and the only valid input (both flags and values) would be provided while testing.