Lab 9 – Python and Shell Commands

This lab may be done either solo or in pairs.

In this lab, you are going to use Python and shell commands to parse a CSV file and print some statistics about its content. The first part of this lab is about using shell commands in python (see: awk, sed, sort, uniq). The second part is about implementing a program to manage the IMDb movies distribution, it is all up to you!

You need to use only the tools/packages that are provided to u.

Task 0

In this task, you are going to download a third-party library for integrating shell commands into Python and use it to extract a list of all the students in the file, and their number. You also need to extract a list of all error-codes (no repetitions) and their number.

Task 0a – Preparation

We are going to download a library that adds shell commands to python and use it during the lab. Do the following:
Download plumbum library:
$>wget
https://pypi.python.org/packages/50/15/f26f60e1bb82aabed7ff86f3fd2976784047f9a291c63ac9019086a69559/plumbum-1.6.3.tar.gz#md5=e0c588ba9271711fae3beb8c0511e8a9

Uncompress the file:
$>tar -xzvf plumbum-1.6.3.tar.gz

Change directory:
$> cd plumbum-1.6.3/

Your python code should be written in this directory.

Task 0b

The file to be parsed contains error-codes describing errors made by different students and how much to reduce for each error code. The format of the file is: student\tab error_code1:1|error_code2:1|error_code3:0.5... where the number after each error_code is used to give partial reduction. 1 means full reduction. Any number less than 1, means a partial reduction for its relevant error_code.
Example:
Danny no_README:1|wrong_file_name:1|code_repetition:0.5|no_task2:1

Use the following file grades_error-codes.

Write procedures in python that receive a grade file in the given format, see above, and calculate the following, one procedure per task:

- A list of all students mentioned in the file.
- The number of students is mentioned in the file.
- A list of all error codes mentioned in the file together with how many times each error code was mentioned.
- The number of unique error-codes found in the file.

All these tasks must be done using shell commands in python, see awk, sed, sort, uniq, wc. Each calculation must be a line of shell commands. Shell commands in python return a 'n' separated list of strings.

For example: in order to view the first field of every line of a file 'test', you can use `awk -F '\t' '{print $1}' test` where `'\t'` is the separator. In the reading material, you can see several links that contain information and additional examples for using awk, sed, sort, wc, and uniq.

Task1 and Task2 can be done either with shell commands in python or regular python commands.

Task 1

You are about to implement the program managing the IMDB movies distribution, it is all up to you! You need to provide data and statistics. You have to implement tasks 1 and 2 as python code. Solving them as GUI with buttons (see Task 3) will be considered as a bonus.

The first file is “IMDb movies.csv” (Download from Moodle) which stores the data regarding the movies in the following format:

imdb_title_id,original_title,year,genre,duration,countries

Use built-in python package csv and not others.

Task 1a

Calculate the number of movies that have been presented in each country (Note, that movie can be presented in more than one country) and output it to ***movies.stats*** in the following format: country_name|number_of_movies.

tt7176472,Vierges,2018,Drama,90,**France, Israel, Belgium**

Task 1b

Calculate the number of movies presented in a specific country that is released after a specific date.

Task 1c

Draw a histogram of the number of movies presented each year, see this page.

If you receive an error when trying to import matplotlib, run the following command in the shell: pip install –user matplotlib

Task 2

Task2a

Calculate the total duration of movies for each genre. Output it to *genre.stats* in the following format: genre|average duration.

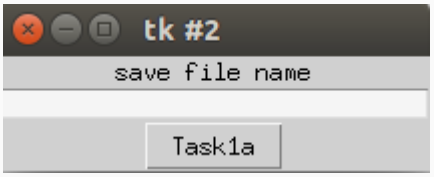
Task2b

For a **specific country** calculate the number of movies for each genre movie that has been published in this country.

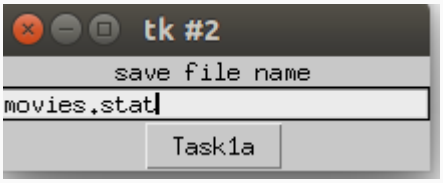
Task 3 - Bonus task

In this task, we have to implement tasks 1 and 2 by building GUI window with buttons. The GUI includes label text that indicates the task section name, text boxes (in case), buttons that print or calculate the result.

For example task1a:



The **button "task1a"** runs a function that calculates and saves the result to the file given in the **text box.**



You have to use this package tkinter