**IF100 – Fall 2018-2019**

**Practicing Homework 6 - Hotel California**

**Will Not Be Graded - No deadline**

**Introduction**

The aim of this homework is to practice on file operations and dictionary data structure usage.

**Story**

You probably know about the countless websites on accommodation finding and reviewing. Verified users are able to rate the hotels on their experiences and these reviews affect the average rating of the hotels. In this homework, you are going to process a small portion of such data to find out some statistics about the given hotels.   
  
  


**Description**

In this homework, you will write a Python program that reads a bunch of hotels and another bunch of reviews from users on specific hotels. After that, your program will print some statistics from the given data and then will accept hotel queries for finding the average ratings of particular hotels.

**Inputs**

The data of the hotel information and the user reviews are provided in two seperate files, named "hotels.txt" and "reviews.txt" respectively. These file names are fixed, hence your program should not ask the user of the program about the file names, it can directly open the files with given names.

The following is the format of "hotels.txt" file:

*someHotelID someHotelName*

*…*

*…*

The following is the format of "reviews.txt":

*someUsername aHotelID rating*

*…*

*…*

Please note that, in each line of the both files, consecutive items are separated by a single **tab** character. The tab character is represented as "\t" in Python. A hotel name (*someHotelName*) may consist of multiple words, in such a case these words are separated with a single regular space (" "). A username (*someUserName*) is just a single word. Ratings (*rating*) are integer values between 1 and 5 (both inclusive). You may not make any assumptions on the number of lines in these files. However, you may assume that there isn’t any empty line on these files and the files include at least one item.

Please see the attached example files for better understanding of the file formats.

In the next phase of the problem, your program should be taking hotel names as input from the user repeatedly, until the exact phrase “quit” is given by the user.

**What to do and what to output**

Initially your program should read and parse the both files and store the related information in some dictionaries that you will define. We are very aware that this homework can be done also by:

1. Using lists instead, which makes the implementation much complicated.
2. Using files only by parsing several times, which is a very inefficient and bad solution.

Also, we would like to remind that this homework is given for your preparation for the final exam and dictionaries will be an important part of the exam. Doing this homework by avoiding using dictionaries will not help you much in this direction.

After extracting useful relations from the files, your program should output:

1. The name of the user who has submitted the most reviews,
2. The name of the hotel with greatest average rating.

For both, you can assume that there are no more than one user/hotel that achieves the top place in these statistics.

After this, your program proceeds to the second phase and starts collecting user input repeatedly. Each time, the user will input a hotel name and your program should print the average rating of that hotel. Most probably, the average rating will not be a whole number. As most such websites do, your program should print the average rating with one decimal digit (i.e, 4.1). For this, you can use the format function with parameter “.1f”

The given hotel names do not always come perfect from the user. Such hotel may not even exist in the hotel names read from the file or the hotel exists but it has not been reviewed by any users. Your program should be able to detect these two cases and output warning messages. See the sample run for the warning output formats.

Taking input from the user cannot go infinitely; we do not like infinitely running programs. Your program should stop taking user input and terminate when the exact phrase “quit” is given by the user.

**Important Remarks on Working with Files**

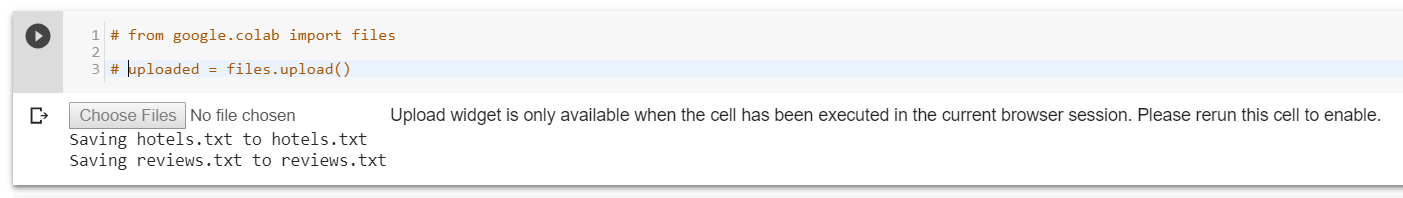
When it comes to working with files, there are some differences in the way we use Google Colab and GradeChecker.

While working in Google Colab, your code does not run on your computer, hence there is no direct access to the files you have on your computer. However, you can upload your text files to Google Colab by running this two line snippet on a separate code cell:

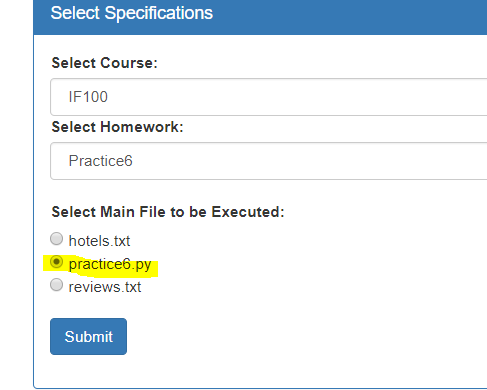
*from google.colab import files*

*uploaded = files.upload()*

Every time you run this code, you will be prompted to upload a file from your computer. Please note that this snippet may not work for certain browsers, so Chrome/Chromium is recommended. Please also be reminded that you need to repeat the upload process every time you reopen your Colab file or you reconnect to the environment.

**If you are going to try your code on GradeChecker, you should remove this code cell (or make them as comment lines as shown below) before you download the .py version of your code, as this snippet only works on Google Colab and it will cause error on GradeChecker.  
  
**

Another note on GradeChecker is that GradeChecker does not have the txt files related to the homeworks already. So, while uploading your code to GradeChecker, you should upload the two txt files together with your .py file (3 files in total) and select your .py file as main file to be executed as shown below. It also means that you can create your custom txt files and test your program on them via GradeChecker, if you wish.

****

**Sample Run**

Below, we provide a sample run of the program that you will develop. The *italic* and **bold** phrases are inputs taken from the user.

The user who posted the most reviews is kookybob  
The best hotel: Taj Diplomatic Enclave  
  
Please enter a hotel name: ***Brewery Gulch Inn***  
Nobody has rated for this hotel yet.  
Please enter a hotel name: ***Alila Manggis***  
The average rating of the hotel: 4.0  
Please enter a hotel name: ***Planters Inn***  
The average rating of the hotel: 4.5  
Please enter a hotel name: ***Hewing Hotel***  
The average rating of the hotel: 4.3  
Please enter a hotel name: ***Taj***  
Hotel name does not exist in the database.  
Please enter a hotel name: ***Sol y Luna***  
The average rating of the hotel: 2.0  
Please enter a hotel name: ***Belmond Grand Hotel Timeo***  
Nobody has rated for this hotel yet.  
Please enter a hotel name: ***Kadikoy Rihtim Otel***  
Hotel name does not exist in the database.  
Please enter a hotel name: ***quit***

**Programming and Coding Advice**

It would be easier for you to implement the algorithm of this problem if you **first** try to **draw the flowchart** or **write the pseudocode** so that you can go over your solution to see if there are any errors.

Additionally, this homework is not short to implement and it will be very difficult and problematic to code the solution without decomposing the process into smaller pieces. Indeed, it is always a good idea to **use decomposition and pattern recognition** in programming, regardless of the length of the problem. Just define subproblems in the whole problem, and try to solve these subproblems before.

**How to get help?**

You may ask your questions to TAs, LAs or instructors. Information regarding the office hours of the TAs, LAs and the instructors are available in our Google Drive Folder.

You can use GradeChecker (<http://sky.sabanciuniv.edu:8080/GradeChecker/index.jsp>) to test your implementation. Just a reminder, you will see a character ¶ which refers to a newline in your expected output.

**What and where to submit?**

Well, we already told you that this homework **will not be** **graded**. It means that you will not submit it anywhere. Just use it for practicing for the exam. As stated before, you may still check your expected score on GradeChecker.

Good Luck :)

Tolga Atam, Duygu K. Altop, İnanç Arın