IF100
Files & Dictionary - Practice #3

Introduction

The aim of this example is to practice file operations and dictionaries in Python. The use of file operations is due to the nature of the problem; that is, you cannot finish this practicing assignment without using file operations. You may implement the solution without using a dictionary but that would be less efficient.

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 practicing assignment, you are going to process a small portion of such data to find out some statistics about the given hotels.

Description

In this exercise, 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, which 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). Please note that you cannot make any assumptions on the number of lines of 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 shared 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 of "quit" is given by the user.

Processing and Outputs

Initially, your program should read and parse both of the files and store the related information in some dictionaries that you will define. 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 the 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 of "quit" is given by the user.

We highly recommend the use of dictionaries.

We are very aware that this homework can also be done by:

1) Using lists instead, which makes the implementation much complicated.

2) Using files only by parsing several times, which is very inefficient.

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

We also highly recommend the use of functions.

Sample Runs

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

Sample Run

```
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.