CS 115 Homework 10: Music Recommender+

Overview

In this assignment you will extend the music recommender system from chapter 5 of the textbook. The objectives of this assignment are:

- Practice with imperative programming (while- and for-loops) and mutable data types (lists and dictionaries).
- Practice designing a program that combines several features and needs to be implemented using several functions.

Suggestions

Start early, reading the assignment and sketching some ideas. Then set it aside for a few hours or days. Review your ideas and think about alternatives and improvements before proceeding with the details.

Plan for unit testing: Early on, make notes on how to test the main functions. Define the test cases so the functions can be tested separately, before combining them.

Review at least section 5.6 of the textbook concerning file I/O and section 5.7 which provides code that you are free to modify and extend.

Requirements

Name your source file musicrecplus.py.

When the program starts, it loads the database from the file named “musicrecplus.txt”, which stores the database using the format in the textbook Section 5.7.

The program should prompt the user for a name and explain an opt-out feature. This feature is that if they add a $ to the end of their name, their name and preferences will not be displayed to other users who use the program.

If the user is a new user (not already in the musicrecplus.txt file), they should be prompted to enter their initial preferences before they can move on.

Once the user is logged in and has preferences, the program should provide a “menu” of the following options:

- **Enter Preferences:** The user enters their preferences (artists they like) one at a time, just like the basic program in the textbook. This will replace the preferences already saved in the database.
- **Get Recommendations:** This option will also work just like the basic program in the textbook; the only difference is that it is a separate menu option.
- **Show Most Popular Artists:** Print the artist that is liked by the most users. If there is a tie, print all artists with the most likes.

- **How Popular Is the Most Popular:** Print the artist that is liked by the most users. If there is a tie, print all artists with the most likes [just like *Show Most Popular Artists*]. Additionally, print how many likes the most popular artist has.
- **Which User Has the Most Likes:** Print the name of the user with the greatest number of artists in their preferences, as well as a list of their preferences and the total of preferences. A user with a dollar sign at the end of their name has opted out of this feature, so even if they have the most preferences, do not list them. For example, "JoeSmith$" has opted out but "SeanJohnson" has not. If there is a tie for most likes, print every user.
- **Quit:** When the user chooses to quit, the current database should be written to the file "musicrecplus.txt," replacing old contents (if any).

The main program should be structured as a while-loop that repeatedly offers the user the choice to do one of the options. When he or she quits, it should halt after saving the file.

The program doesn't provide a graphical menu. Instead, it prompts the user with the following "menu" text:

```
Enter a letter to choose an option:
e - Enter preferences
r - Get recommendations
p - Show most popular artists
h - How popular is the most popular
m - Which user has the most likes
q - Save and quit
```

After performing the chosen option, the program should print the menu again and repeat, except for, of course, quit.

If the user enters any letter other than the ones above, the program should print an error message and print the menu again.

Extras

You're welcome to add additional features, but your program is required to include at least these features, associated with the letters listed above. For extra credit, you can implement the following features:

- **Delete Preferences:** This works similarly to Enter Preferences, the main difference being that you are removing preferences from musicrecplus.txt instead of adding them. This will be "d" in the menu. To start, a list of the user's current preferences should be listed so that the user knows what can be deleted. The details of this implementation are up to you. For example, you could number them and have the user choose a number, or you can just have them type out the name exactly as it is presented. You will receive the same credit either way.

  If you choose to implement this option, your Enter Preferences will need to be a little different. Instead of replacing the old preferences, it should add them to the existing list. (5pts)
- **Show Preferences:** Simply prints the current user's preferences. This will be "s" in the menu. If implemented correctly, this should print the most up-to-date list, even if the user just entered new preferences and has not yet quit the program. (5pts)

- **Fixing the Get Recommendations Bug:** If you only have one user in the database and you try to get recommendations, you will run into an error since there are no users to compare to. Edit the Get Recommendations option to check for this, and present the user with your own error message that doesn't crash the whole system and instead prints the menu again. (5pts)

This extra credit will allow you to exceed 100 points on this assignment and will act as additional points towards your overall homework grade. However, if adding these points would cause you to exceed 100 on your overall homework average, they cannot be counted, nor can they be put towards any other assignment category.