CS115 Homework 10: Music Recommender+

Overview

In this assignment you will create a music recommender system similar to the one described in chapter 5 of the textbook. The learning 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.

Notes & Suggestions

This is the most rigorous assignment yet, so start early, and read the *requirements* section of this document carefully. It provides exact specifications that you *must* follow in order to recieve full marks on this assignment.

The requirements section is broken down into the exact functionality your assignment must provide, and the output which it must produce. However, you must decide on the implementation details yourself.

As this assignment is meant to help you focus on while loops, for loops, lists, and dictionaries, you should think about how you might use these tools for each task you complete. Reviewing your ideas and think about alternatives and improvements before proceeding with your plan can help you create better code. However, if you are feeling indecisive, you can always code small things and write tests to figure out how the direction you need to take your code in next.

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. These can be assert statements, print outs, pyunit tests, or anything else you fancy, but remember to comment out these tests before submitting so that your program is able to be graded by the auto-grader.

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 following format:

```
FirstName1 LastName1:Artist1,Artist2,Artist3,...
FirstName2 LastName2:Artist4,Artist5,Artist6,...
```

Note - Artist names may be the same - `Artist1` may equal `Artist5` or something similar. Furthermore, the saved artist list must be in alphabetical order according to Python’s sort. For instance in Python, `'a' < 'b'` is `True`. Therefore, the artist `a` should come before the artist `b` in the user preferences. Furthermore, each line of the file should be sorted by the user’s name. User `Arby Aardvark` should come before `Bethy Bee$` because `'Arby Aardvark' < 'Bethy Bee$'` is `True` in Python. Users who put a `$` as the last character of their name should have that `$` saved to their file even though this denotes a private user. It is expected that the users of this program will not be able to access the save file.

The user should be prompted for thier name. 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 to the `menu`. If the user is not new, they should not be asked their preferences and should immediately be shown the `menu`

The user should be prompted for their name with the phrase `Enter your name (put a $ symbol after your name if you wish your preferences to remain private):` - if the user adds a `$` as the last character of their name, they are in private mode, which affects the recommendations phase.

The `Menu`

The menu should display as follows (exactly):

```
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 the user enters their one character choice, the following describes all possible behavior.

Enter Preferences: The user enters their preferences (artists they like) one at a time, similar to the basic program in the textbook, subject to the following requirements:

- This will replace the preferences already saved in the database.
- The user should be prompted by the text `Enter an artist that you like (Enter to finish):` during the preference entering phase.
- You should keep asking the user for preferences until the user enters the empty string.
- After the user finishes entering preferences, they should be returned to the `menu`.

An example of this interaction is shown at the end.

Get recommendations: This option will also work similar to the basic program in the textbook. However, it is a separate menu option and must follow the following requirements.

- The recommendations should come from the users with the most similarity to the current user.
    - If there are no differences between the user and another client’s preferences, that/those client(s) should be discarded before similarity is rated.
    - If two or more other client’s preferences match the user, all the artists not currently liked by the user should be included in the result.
- The recommendations should appear one per line.
- Artists names should not be included more than once in a single list of recommendations.
- The user should not be recommended any artists they already have entered a prefernce for.
- Recommendations should be sorted according to the rule that if `a` is < than `b` (`a < b` is `True`) than `a` should be recommended before `b`.
- Users in private mode (with a $ at the end of their name) should be excluded from these calculations.
- If there are no preferences available for recommendation, the user should be told `No recommendations available at this time`.
- After the recommendations are done printing, the user should be returned to the `menu`.

An example of this interaction is shown at the end.

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. This is subject to the following requirements:

- Artists should be printed one per line.
- Users in private mode (with a $ at the end of their name) should be excluded from these calculations.
- If there are no artists found by this operation, display to the user `Sorry, no artists found`.
- Artists names should appear only once in these results.
- Artists should be sorted according to the rule that if `a` is < than `b` (`a < b` is `True`) than `a` should be printed before `b`.
- After the artists are done printing, the user should be returned to the `menu`.

An example of this interaction is shown at the end.

How popular is the most popular: Print the number of likes the most popular artist(s) received according to the following requirements.

- Print the number of likes the most popular artist received.
- Users in private mode (with a $ at the end of their name) should be excluded from these calculations.
- If there are no artists found by this operation, display to the user `Sorry, no artists found`.
- In the event of a tie, still only print this number once.
- Print only the number.
- After the artists are done printing, the user should be returned

to the `menu`.

An example of this interaction is shown at the end.

Which user has the most likes: Print the full name(s) of the user(s) who likes the most artists subject to the following requirements:

- Print only one user name per line
- User names should be sorted according to the rule that if `a` is < than `b` (`a < b` is `True`) than `a` should be printed before `b`.
- If there are no artists found by this operation, display to the user `Sorry, no user found`.
- Exclude any user whose name ends in a `$` from these computations.
- If there is a tie, print all tied users.
- The current user should be included in this computation.
- After the user names are done printing, the user should be returned to the `menu`.

An example of this interaction is shown at the end.

Save and quit: When the user chooses to quit, the current database should be written to the `musicrecplus.txt`, replacing old contents (if any). This is subject to the following requirements.

- The name of the file saved to must be `musicrecplus.txt`.
- You must save according to the file format described at the beginning of the requirements section.

- If the file exists, overwrite the contents.
- If the file does not exist, you should create it with the proper contents.
- After this saving occurrs, the program should exit without crashing.

An example of this interaction is shown at the end.

If no option is chosen, the menu should reprint and request input until the user enters a valid option.

All text in this document may be considered a requirement. At no point should your program crash.

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

Do exactly as instructed in this assignment to receive full marks. Do no more than asked. Do exactly as asked. You may import from the cs115 module if needed. This assignment is meant to let you use your (potentially new-found) knowledge of iterative programming.

Extra credit may be associated with this assignment due to it’s difficulty. The decision on this will be announced after the assignment due date. You may work with up to one partner on this assignment.

As always include your id and pledge on the top of the document and the id of your partner if applicable. The definition of your id is (as always) available on canvas. The full text of the honor pledge, as always, is available here: `https://web.stevens.edu/honor/`.

Example Interaction

Before this interaction, `musicrecplus.txt` does not exist.

```
Enter your name (put a $ symbol after your name if you wi
sh your preferences to remain private):
Steph Oro
Enter an artist that you like (Enter to finish):
Fun.
Enter an artist that you like (Enter to finish):
TMBG
Enter an artist that you like (Enter to finish):
Gotye
Enter an artist that you like (Enter to finish):

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
r
No recommendations available at this time
```

```
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
p
Fun.
Gotye
TMBG
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
h
1
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
m
```

```
Steph Oro
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
q
```

After this interaction, `musicrecplus.txt` should now contain:

```
Steph Oro:Fun.,Gotye,TMBG
```

For our next interaction we assume `musicrecplus.txt` contains:

```
Anne Adamant:50 cent,eminem,lil wayne,snoop dog
Bacon Bryant$:Britney Spears,Gotye,Kesha,TMBG
Caesar Zeppeli:Fun.,Gotye,Sara Bareilles
Hidden Powers:Baby Metal,FLOW,Spyair,Vipera
Sappho of Lesbos:Anna Kendrick,Kerkylas of Andros,Sara Bareilles
Steph Oro:Fun.,Gotye,Sara Bareilles
```

In this context, an interaction:

```
Enter your name (put a $ symbol after your name if you wi
sh your preferences to remain private):
Steph Oro
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
r
Anna Kendrick
Kerkylas of Andros
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
p
Sara Bareilles
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
h
3
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
m
Anne Adamant
Hidden Powers
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
e
Enter an artist that you like (Enter to finish):
Vipera
Enter an artist that you like (Enter to finish):
TMBG
Enter an artist that you like (Enter to finish):

```

```
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
r
Baby Metal
FLOW
Spyair
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
p
Sara Bareilles
Vipera
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
```

```
h
2
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
m
Anne Adamant
Hidden Powers
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
q
```

After this, the `musicrecplus.txt` should now read:

```
Anne Adamant:50 cent,eminem,lil wayne,snoop dog
Bacon Bryant$:Britney Spears,Gotye,Kesha,TMBG
Caesar Zeppeli:Fun.,Gotye,Sara Bareilles
Hidden Powers:Baby Metal,FLOW,Spyair,Vipera
```

```
Sappho of Lesbos:Anna Kendrick,Kerkylas of Andros,Sara Bareilles
Steph Oro:TMBG,Vipera
```

Good luck.