Assignment 7 – Music Library

Goals

- Practice using dictionaries
- Continue working with functions

Requirements

- You will create a program that simulates a user's music library. You will be importing an already existing music library from a file we have provided for you, and saving the music library with any of the user's changes back to the same file. Your program will allow a user to interact with their music library in different ways.
- The music library will be represented as a dictionary. The **`keys`** in the dictionary are artists' names (**`strings`**), and each **`value`** is a **`list`** of albums by that artist.

Functions provided for you

- In order to store and load more complex data (such as dictionaries and lists) to and from a file we can use binary files as opposed to normal text files.
- Binary files store data in a computer-readable format which is not human readable. Python uses a module called **`Pickle`** which allows us to do this (optional: read more about Pickling).
- We have provided a helper file which will help you read a dictionary from a binary file as well as save a dictionary to a binary file.
- Download the file called **`MusicLibraryHelper.py`** from Blackboard and include it in the same directory as your Assignment 7 python file. To use the helper functions, include the line "**`import MusicLibraryHelper`**" at the top of your code to access the functions. You will also need to download and include the starter binary file in your directory ("**`musicLibrary.dat`**")
- Note: sometime the "**`import MusicLibraryHelper`**" line will be underlined in red saying it could not import the library even if it is in the same directory. Often times that red underline is incorrect, though, and you have successfully imported MusicLibraryHelper. Before trying to fix the import error, try to call the loadLibrary() function (see below). If your program runs successfully then you have correctly imported MusicLibraryHelper
- To use the load library function, use the following syntax:

```
MusicLibraryHelper.loadLibrary(libraryFileName)
```

  - Input: the name of the file containing the music library dictionary

        - Return value: a dictionary representing the music library, where each key is an artist and each value is a **`list`** of albums by that artist
        - Loads a dictionary from the specified binary file
- To use the save library function, use the following syntax:
  `MusicLibraryHelper.saveLibrary(libraryFileName, musicLibDct)`
        - Input: the name of the file that the music library will be saved to; the dictionary representing the music library
        - Return value: none
        - Writes a dictionary to the specified output file

Functions you will create

- Define the functions below that have the specified inputs and return values:
    - **`displayMenu()`**
        - Input: none
        - Return value: none
        - Print out the menu options to the user (see sample output)
    - **`displayLibrary(musicLibDictionary)`**
        - Input: a dictionary representing the music library
        - Return value: none
        - Print out the entire music library. The user should be able to see all the albums associated with each artist in their library in a nice, readable format (see sample output).
    - **`displayArtists(musicLibDictionary)`**
        - Input: a dictionary representing the music library
        - Return value: none
        - Print out all the artists currently in the user's music library.
    - **`addAlbum(musicLibDictionary)`**
        - Input: a dictionary representing the music library
        - Return value: none
        - Ask the user for the name of an artist and the name of the album they would like to add.
        - Check if the artist is already in the dictionary, and if so, add the album to the artist's existing list of albums. If the artist is not in the dictionary, add a new key (artist) to the dictionary along with the new value (list containing the new album).
        - Error checking: the user's input should not be case sensitive – if they enter **`"adele"`** or **`"ADELE"`** and the artist **`"Adele"`** already exists in their library, this should add the new album to the already existing key.

If the user is adding an album that already exists for an artist, do nothing (no duplicate albums). This should also be case insensitive.

- **`deleteAlbum(musicLibDictionary)`**
    - Input: the dictionary representing the music library
    - Return value: a **`boolean`** value – **`True`** if an album was successfully deleted, or **`False`** if the album was not successfully deleted.
    - Ask the user for the name of an artist and the name of the album they would like to remove.
    - Check that both the artist and the album are in the dictionary before removing the album, and then return **`True`**.
    - If the artist or the album are not in the dictionary, do not modify the dictionary and return **`False`**.
    - Error checking: the album name and artist names should not be case sensitive. If the album being deleted is the last album associated with the artist, be sure to remove the artist from the dictionary as well (do not leave the artist key with an empty list value).
- **`deleteArtist(musicLibDictionary)`**
    - Input: the dictionary representing the music library
    - Return value: a **`boolean`** value – **`True`** if an artist was successfully deleted, or **`False`** if the artist was not successfully deleted.
    - Ask the user for the name of an artist. Check that the artist is in the dictionary before deleting the artist. Return **`True`** if the artist was deleted, or **`False`** if the artist was not found in the music library.
    - Error checking: the artist's name should not be case sensitive.
- **`searchLibrary(musicLibDictionary)`**
    - Input: the dictionary representing the music library
    - Return value: none
    - Ask the user for a search term (note that this search term could be an artist or an album)
    - Search through all of the artists and print out the names of any artists containing the search term.
    - Search through all of the albums in the dictionary and print out any album names containing the search term.
    - Note that your search should not be case sensitive. In other words, searching for **`"ADELE"`** or **`"adele"`** should still return to you the artist **`"Adele"`**
- **`generateRandomPlaylist(musicLibDictionary)`**
    - Input: the dictionary representing the music library
    - Return value: none

        - Generate a random playlist for the user by randomly selecting one album from every artist in the library. Print the library out to the user in a nice format.
    - **`main()`**
        - Input: none
        - Return value: none
        - Start by importing the music library from the binary data file using **`MusicLibraryHelper.loadLibrary(…)`** to create the music library dictionary where the keys are artists and the values are a list of albums by that artist.
        - In a **`while`** loop, print the menu out to the user and allow them to select a menu option. Be sure to handle invalid menu choices.
        - Based on the user's selection, call the corresponding function.
            - For **`deleteAlbum`** and **`deleteArtist`**, be sure to store the return values in a variable. Based on the return value (a **`boolean`**), print out an appropriate message about whether or not the delete was successful.
        - When the user decides to exit, save their music library to the same file it was loaded in from using **`MusicLibraryHelper.saveLibrary(…)`**.

Extra Credit

Add an additional function and menu option to generate a custom playlist.

- **`generateCustomPlaylist(musicLibDictionary)`**
    - Input: the dictionary representing the music library
    - Return value: none
    - Generate a custom playlist by letting the user select the artists and their albums
    - As long as the user wants to add another item to the playlist:
        - Print the current playlist (at first it should be empty)
        - Print the artists in the library and a number next to each artist's name. Hint: store a list of the artists to make the next step easier.
        - Ask the user to select an artist by selecting their corresponding number, be sure to include error checking. The selected artist can be determined by using the user's input as an index position in your artist list.
        - Using the selected artist, print out a list of the artist's albums with a number next to each album name.

    - Ask the user to select an album number, be sure to include error checking.
    - Keep track of the playlist by using a list of strings, where each string contains the album name and artist (as one string).
    - Ask the user if they want to continue adding to their playlist. Be sure to include error checking
- Once the user is done building their playlist, print it out one final time.

Sample Output

```
Welcome to Your Music Library
Options:
    1) Display library
    2) Display all artists
    3) Add an album
    4) Delete an album
    5) Delete an artist
    6) Search library
    7) Generate a random playlist
    8) Make your own playlist
    9) Exit
> 1

Artist: The Who
Albums:
      - Tommy
Artist: Adele
Albums:
      - 19
      - 21
      - 25
Artist: The Beatles
Albums:
      - Abbey Road
      - Let it Be

Welcome to Your Music Library
Options:
    1) Display library
    2) Display all artists
    3) Add an album
    4) Delete an album
    5) Delete an artist
    6) Search library
    7) Generate a random playlist
    8) Make your own playlist
    9) Exit
> 2

Displaying all artists:
- The Who
- Adele
- The Beatles
```

```
Welcome to Your Music Library
Options:
    1) Display library
    2) Display all artists
    3) Add an album
    4) Delete an album
    5) Delete an artist
    6) Search library
    7) Generate a random playlist
    8) Make your own playlist
    9) Exit
> 3

Enter artist: the BEAtles
Enter album: Yellow Submarine

Welcome to Your Music Library
Options:
    1) Display library
    2) Display all artists
    3) Add an album
    4) Delete an album
    5) Delete an artist
    6) Search library
    7) Generate a random playlist
    8) Make your own playlist
    9) Exit
> 1

Artist: The Who
Albums:
      - Tommy
Artist: Adele
Albums:
      - 19
      - 21
      - 25
Artist: The Beatles
Albums:
      - Abbey Road
      - Let it Be
      - Yellow Submarine

Welcome to Your Music Library
```

```
Options:
    1) Display library
    2) Display all artists
    3) Add an album
    4) Delete an album
    5) Delete an artist
    6) Search library
    7) Generate a random playlist
    8) Make your own playlist
    9) Exit
> 4

Enter artist: adele
Enter album: 21
Delete album success!

Welcome to Your Music Library
Options:
    1) Display library
    2) Display all artists
    3) Add an album
    4) Delete an album
    5) Delete an artist
    6) Search library
    7) Generate a random playlist
    8) Make your own playlist
    9) Exit
> 5

Enter artist to delete: Drake
Delete artist failed.

Welcome to Your Music Library
Options:
    1) Display library
    2) Display all artists
    3) Add an album
    4) Delete an album
    5) Delete an artist
    6) Search library
    7) Generate a random playlist
    8) Make your own playlist
    9) Exit
> 6
```

```
Please enter a search term: the
Artists containing 'the'
- The Who
- The Beatles
Albums containing 'the'
      No results

Welcome to Your Music Library
Options:
    1) Display library
    2) Display all artists
    3) Add an album
    4) Delete an album
    5) Delete an artist
    6) Search library
    7) Generate a random playlist
    8) Make your own playlist
    9) Exit
> 7

Here is your random playlist:
- Tommy by The Who
- 25 by Adele
- Yellow Submarine by The Beatles

Welcome to Your Music Library
Options:
    1) Display library
    2) Display all artists
    3) Add an album
    4) Delete an album
    5) Delete an artist
    6) Search library
    7) Generate a random playlist
    8) Make your own playlist
    9) Exit
> 7

Here is your random playlist:
- Tommy by The Who
- 19 by Adele
- Let it Be by The Beatles

Welcome to Your Music Library
Options:
    1) Display library
    2) Display all artists
    3) Add an album
    4) Delete an album
    5) Delete an artist
    6) Search library
    7) Generate a random playlist
    8) Make your own playlist
    9) Exit
> 8

Your playlist so far:
----
0) The Who
1) Adele
2) The Beatles
Select an artist from the list by entering its number: 0
0) Tommy
Select an album from the list by entering its number: 12
*Error, please try again.
Select an album from the list by entering its number: 0
Would you like to continue building your playlist? (y/n) y
Your playlist so far:
- Tommy by The Who
----
0) The Who
1) Adele
2) The Beatles
Select an artist from the list by entering its number: 1
0) 19
1) 25
Select an album from the list by entering its number: 1
Would you like to continue building your playlist? (y/n) n
Your completed playlist:
- Tommy by The Who
- 25 by Adele

Welcome to Your Music Library
Options:
    1) Display library
    2) Display all artists
    3) Add an album
    4) Delete an album
    5) Delete an artist
    6) Search library
    7) Generate a random playlist
    8) Make your own playlist
    9) Exit
> 9

Saving music library...
Goodbye!
```

Deliverables and Submission Instructions

- Create a folder on your computer called
  `ITP115_a#_lastname_firstname`
  (replace # with this lab number)
- Inside the folder, include your python source code
- Compress the folder (make a zip file) called
  `ITP115_a#_lastname_firstname.zip`
  (replace # with this assignment number)
- Upload zip file to Blackboard site for our course

Grading

Item	Points
Music Library	30
Total*	**30**

** Points will be deducted for poor code style, or improper submission.*