Assignment 10 – Dictionaries

Learning Objective

Create a dictionary to manage data. Allow the user to insert, update, and delete data from the dictionary.

Assignment Description

Write a program that manages a user's music library. The music library is represented using a dictionary. The **keys** in the dictionary are artists' names (strings), and each **value** is a list of albums (strings) by that artist.

We have provided a file named **music_library.dat**, which contains a dictionary with a few artists and their albums. This file is a binary file and is not human readable. You need to add this file to your directory.

We have provided a helper file called **MusicLibraryHelper.py** that you will need to add to your directory. This file contains functions to load a binary file containing a dictionary and then return a dictionary as well as save a dictionary to a binary file. These functions use a module called Pickle to load and write objects, specifically dictionaries. Here are descriptions of the two functions that you will need to use:

- MusicLibraryHelper.loadLibrary(libraryFileName)
  - Parameter: libraryFileName is the name of the file containing the music library dictionary
  - Return value: a dictionary representing the music library where each key is an artist (string) and each value is a list of albums (string) by that artist
  - Loads a dictionary from the specified binary file
- MusicLibraryHelper.saveLibrary(libraryFileName, musicLibDct)
  - Parameter 1: libraryFileName is the name of the file the music library will be saved to
  - Parameter 2: musicLibDct is the dictionary representing the music library
  - Return value: None
  - Writes a dictionary to the specified binary file

Your program will allow a user to interact with their music library in different ways. When the user is done, you will call the appropriate function to save it to a binary file.

Steps

1. In PyCharm (Community Edition), open an existing project (such as ITP115) or create a new project.
   - If you open an existing project, then create a new directory (probably under the Assignments directory) named **a10_*last_first*** where *last* is your last/family name and *first* is your preferred first name.
   - If you create a new project, then name it **a10_*last_first*** where *last* is your last/family name and *first* is your preferred first name.
2. In the project or directory, create a new Python file called **assignment10.py**. At the top of the file, put comments in the following format and replace the name, email, and section with your actual information:

   ```
   # Name, USC email
   # ITP 115, Spring 2022
   # Section: number or nickname
   # Assignment 10
   # Description:
   # Describe what this program does.
   ```

3. Put the **music_library.dat** file and the **MusicLibraryHelper.py** file in your a10_*last_first* directory.
   - Download the files from Blackboard under the item for this assignment.
   - Drag them onto your a10_*last_first* directory in PyCharm.
4. Place the following line at the top of your code below your top comment block:

   ```
   import MusicLibraryHelper
   ```

5. Import the **random** module.
6. Define the **displayMenu()** function.
   - Parameter: None
   - Return value: None
   - Print out the menu options to the user:

```
Manage Your Music Library
   a) Display library
   b) Display artists
   c) Add an artist/album
```

```
   d) Delete an album
   e) Delete an artist
   f) Generate a random playlist
   g) Exit
```

7. Define the displayLibrary(dictionary) function.
   - Parameter: 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 the following format:

```
Artist: The Beatles
   Albums:
      Abbey Road
      Let It Be
Artist: Bruno Mars
   Albums:
      24K Magic
Artist: Adele
   Albums:
      19
      21
      25
```

8. Define the displayArtists(dictionary) function.
   - Parameter: a dictionary representing the music library
   - Return value: None
   - Print out the artists in the music library.

```
Artists:
   The Beatles
   Bruno Mars
   Adele
```

9. Define the addAlbum(dictionary) function.
   - Parameter: a dictionary representing the music library
   - Return value: None
   - Get input from the user for the name of the artist and the name of the album that they want to add.

```
Enter artist: SELENA gomez
Enter album: Revelacion
```

- The user may not enter the artist and album in the correct case. Use the str.title() method to convert both of them to Title Case.
- Check if the artist is already in the dictionary. If so, add the album to the artist's existing list of albums. If the album already exists, then do not add it. 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).

10. Define the **deleteAlbum(dictionary)** function.
    - Parameter: a 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
    - Get input from the user for the name of the artist and the name of the album that they want to delete.

```
Enter artist: adele
Enter album: 21
```

- The user may not enter the artist and album in the correct case. Use the str.title() method to convert both of them to Title Case.
- 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.

11. Define the **deleteArtist(dictionary)** function.
    - Parameter: a 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
    - Get input from the user for the name of the artist and the name of the album that they want to delete.

```
Enter artist to delete: bruno MARS
```

- The user may not enter the artist in the correct case. Use the str.title() method to convert it to Title Case.
- Check that the artist is in the dictionary before removing the artist, and then return True.
- If the artist is not in the dictionary, do not modify the dictionary and return False.

12. Define the **generateRandomPlaylist(dictionary)** function.
    - Parameter: a 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 the following format:

```
Here is your random playlist:
   Abbey Road by The Beatles
   19 by Adele
   Revelacion by Selena Gomez
```

13. Define and call the **main()** function.
    - Create a dictionary by calling the MusicLibraryHelper.loadLibrary() function which returns a dictionary. For the argument, use "music_library.dat". Capture the return value in a variable, which you will using throughout the main() function.
    - Using a while loop, display the menu to the user by calling the displayMenu() function and get user input using "Choice: " for the prompt. Here's an example:

```
Manage Your Music Library
   a) Display library
   b) Display artists
   c) Add an artist/album
   d) Delete an album
   e) Delete an artist
   f) Generate a random playlist
   g) Exit
Choice: B
```

- Based on the user's input, call the corresponding function. Allow the user to enter their choice in upper or lower case.

- The deleteAlbum() and deleteArtist() functions have return values. Based on the return value (boolean), print out an appropriate message about whether or not the delete was successful.

```
Delete album success
Delete album failed
Delete artist success
Delete artist failed
```

- If the enter enters an invalid choice, print the following message:

```
Invalid entry
```

- When the user enters the appropriate choice to quit, ask the user for a file name for the music library.

```
Enter music library filename: my_music.dat
```

- Save the music library using the MusicLibraryHelper.saveLibrary() function and print out a message to the user.

```
Saved music library to my_music.dat
```

14. Be sure to comment your code. This means that there should be comments throughout your code. Put a comment block before each function stating the parameters, return values, and what that function does. Points will be deducted for not having comments.
15. Follow coding conventions. You should use lowerCamelCase or snake_case for variable names. You are welcome to create any variables that you need.
16. Test the program. Look at the Sample Output below. Assignments that do not run are subject to 20% penalty.
17. Prepare your submission:
    - Find the **a10_*last_first*** folder on your computer and compress it. This cannot be done within PyCharm.
    - On Windows, use **File Explorer** to select the folder. Right click and select the Send to -> Compressed (zipped) folder option. This will create a zip file.
    - On Mac OS, use **Finder** to select the folder. Right click and select the Compress "*FolderName*" option. This will create a zip file.

18. Upload the zip file to your Blackboard section:
    - On Blackboard, navigate to the appropriate item.
    - Click on the specific item for this assignment.
    - Click on the **Browse Local Files** button and select the zip file.
    - Click the **Submit** button.

Grading

- This assignment is worth 40 points.
- Make sure that you the program runs. Points will be taken off if the graders have to edit the source code to test your program.
- Make sure to submit your assignment correctly as described above. Points will be taken off for improper submission.

Item	Points
music_library.dat and MusicLibraryHelper.py in directory	2
displayLibrary()	4
displayArtists()	4
addAlbum()	6
deleteAlbum()	6
deleteArtist()	6
generateRandomPlaylist()	6
main()	6
Total	**40**

Sample Output

```
Manage Your Music Library
   a) Display library
   b) Display artists
   c) Add an artist/album
   d) Delete an album
   e) Delete an artist
   f) Generate a random playlist
   g) Exit
Choice: 1
Invalid entry

Manage Your Music Library
   a) Display library
   b) Display artists
   c) Add an artist/album
   d) Delete an album
   e) Delete an artist
   f) Generate a random playlist
   g) Exit
Choice: a
Artist: The Beatles
   Albums:
      Abbey Road
      Let It Be
Artist: Bruno Mars
   Albums:
      24K Magic
Artist: Adele
   Albums:
      19
      21
      25

Manage Your Music Library
   a) Display library
   b) Display artists
   c) Add an artist/album
   d) Delete an album
   e) Delete an artist
   f) Generate a random playlist
   g) Exit
Choice: B
Artists:
   The Beatles
   Bruno Mars
   Adele

Manage Your Music Library
   a) Display library
   b) Display artists
   c) Add an artist/album
   d) Delete an album
   e) Delete an artist
   f) Generate a random playlist
   g) Exit
Choice: c
Enter artist: the BEATles
Enter album: Yellow submarine

Manage Your Music Library
   a) Display library
   b) Display artists
   c) Add an artist/album
   d) Delete an album
   e) Delete an artist
   f) Generate a random playlist
   g) Exit
Choice: c
Enter artist: SELENA gomez
Enter album: Revelacion

Manage Your Music Library
   a) Display library
   b) Display artists
   c) Add an artist/album
   d) Delete an album
   e) Delete an artist
   f) Generate a random playlist
   g) Exit
Choice: c
Enter artist: selena Gomez
Enter album: Rare

Manage Your Music Library
   a) Display library
   b) Display artists
   c) Add an artist/album
   d) Delete an album
   e) Delete an artist
   f) Generate a random playlist
   g) Exit
Choice: a
Artist: The Beatles
   Albums:
      Abbey Road
      Let It Be
      Yellow Submarine
Artist: Bruno Mars
   Albums: 24K Magic
Artist: Adele
   Albums:
      19
      21
      25
Artist: Selena Gomez
   Albums:
      Revelacion
      Rare

Manage Your Music Library
   a) Display library
   b) Display artists
   c) Add an artist/album
   d) Delete an album
   e) Delete an artist
   f) Generate a random playlist
   g) Exit
Choice: d
Enter artist: adele
Enter album: 21
Delete album success

Manage Your Music Library
   a) Display library
   b) Display artists
   c) Add an artist/album
   d) Delete an album
   e) Delete an artist
   f) Generate a random playlist
   g) Exit
Choice: e
Enter artist to delete: Drake
Delete artist failed

Manage Your Music Library
```

```
   a) Display library
   b) Display artists
   c) Add an artist/album
   d) Delete an album
   e) Delete an artist
   f) Generate a random playlist
   g) Exit
Choice: e
Enter artist to delete: bruno MARS
Delete artist success

Manage Your Music Library
   a) Display library
   b) Display artists
   c) Add an artist/album
   d) Delete an album
   e) Delete an artist
   f) Generate a random playlist
   g) Exit
Choice: a
Artist: The Beatles
   Albums:
      Abbey Road
      Let It Be
      Yellow Submarine
Artist: Adele
   Albums:
      19
      25
Artist: Selena Gomez
   Albums:
      Revelacion
      Rare

Manage Your Music Library
   a) Display library
   b) Display artists
   c) Add an artist/album
   d) Delete an album
   e) Delete an artist
   f) Generate a random playlist
   g) Exit
Choice: f
Here is your random playlist:
   Yellow Submarine by The Beatles
   19 by Adele
   Rare by Selena Gomez

Manage Your Music Library
   a) Display library
   b) Display artists
   c) Add an artist/album
   d) Delete an album
   e) Delete an artist
   f) Generate a random playlist
   g) Exit
Choice: f
Here is your random playlist:
   Let It Be by The Beatles
   19 by Adele
   Revelacion by Selena Gomez

Manage Your Music Library
   a) Display library
   b) Display artists
   c) Add an artist/album
   d) Delete an album
   e) Delete an artist
   f) Generate a random playlist
   g) Exit
Choice: x
Invalid entry

Manage Your Music Library
   a) Display library
   b) Display artists
   c) Add an artist/album
   d) Delete an album
   e) Delete an artist
   f) Generate a random playlist
   g) Exit
Choice: g
Enter music library filename: my_music.dat
Saved music library to my_music.dat
```

Extra Credit

Add an additional function and menu option to generate a custom playlist. Define and call the generateCustomPlaylist(dictionary) function:

- 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.