Greg Eisele

Nov 13, 2022

IT FDN 110 B

Assignment 5

Introduction

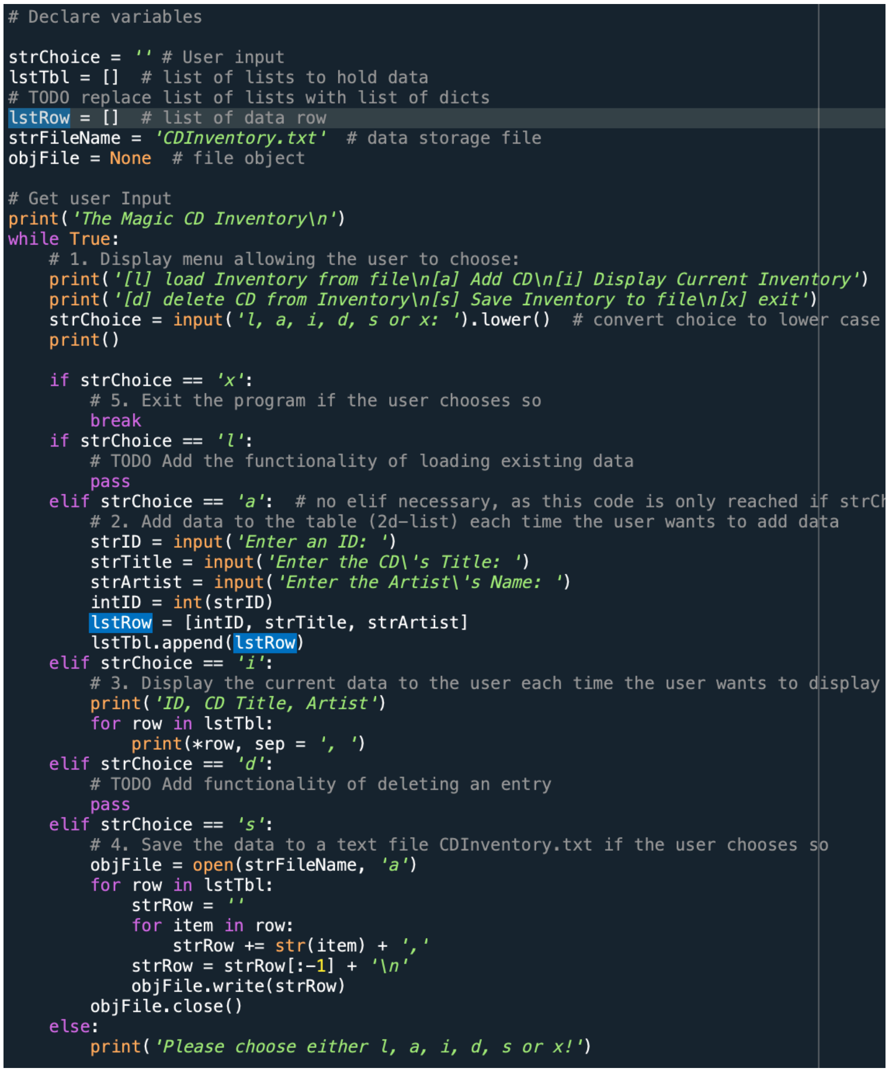
Week 5's lesson was a continuation of Lists, and an introduction to Dictionaries. Additionally, there was focus on real world application of modifying another person's code to change a programs functionality.

Assignment

For this weeks assignment, we were provided with starter code "CDInventory_Starter.py". We were asked to modify the script as required to replace the inner data structure by dictionaries, add the functionality of loading existing data, and add functionality of deleting an entry. In this section, I will walk through the steps I took to accomplish these goals.

<u>Modify the script as required to replace the inner data structure by dictionaries</u>

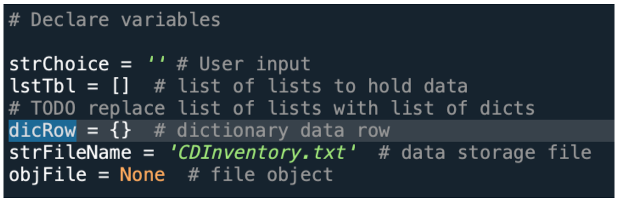
The inner data structure of the starter code uses lists, highlighted as "lstRow" in figure 1. To accomplish the assignment, the "lstRow" must be removed from the code, and replaced with dictionaries as the inner data structure.



```
# Declare variables

strChoice = '' # User input
lstTbl = []  # list of lists to hold data
# TODO replace list of lists with list of dicts
lstRow = []  # list of data row
strFileName = 'CDInventory.txt'  # data storage file
objFile = None  # file object

# Get user Input
print('The Magic CD Inventory\n')
while True:
    # 1. Display menu allowing the user to choose:
    print('[l] load Inventory from file\n[a] Add CD\n[i] Display Current Inventory')
    print('[d] delete CD from Inventory\n[s] Save Inventory to file\n[x] exit')
    strChoice = input('l, a, i, d, s or x: ').lower()  # convert choice to lower case
    print()

    if strChoice == 'x':
        # 5. Exit the program if the user chooses so
        break
    if strChoice == 'l':
        # TODO Add the functionality of loading existing data
        pass
    elif strChoice == 'a':  # no elif necessary, as this code is only reached if strCh
        # 2. Add data to the table (2d-list) each time the user wants to add data
        strID = input('Enter an ID: ')
        strTitle = input('Enter the CD\'s Title: ')
        strArtist = input('Enter the Artist\'s Name: ')
        intID = int(strID)
        lstRow = [intID, strTitle, strArtist]
        lstTbl.append(lstRow)
    elif strChoice == 'i':
        # 3. Display the current data to the user each time the user wants to display
        print('ID, CD Title, Artist')
        for row in lstTbl:
            print(*row, sep = ', ')
    elif strChoice == 'd':
        # TODO Add functionality of deleting an entry
        pass
    elif strChoice == 's':
        # 4. Save the data to a text file CDInventory.txt if the user chooses so
        objFile = open(strFileName, 'a')
        for row in lstTbl:
            strRow = ''
            for item in row:
                strRow += str(item) + ','
            strRow = strRow[:-1] + '\n'
            objFile.write(strRow)
        objFile.close()
    else:
        print('Please choose either l, a, i, d, s or x!')
```

Figure 1 - Starter Code with "lstRow" Highlighted

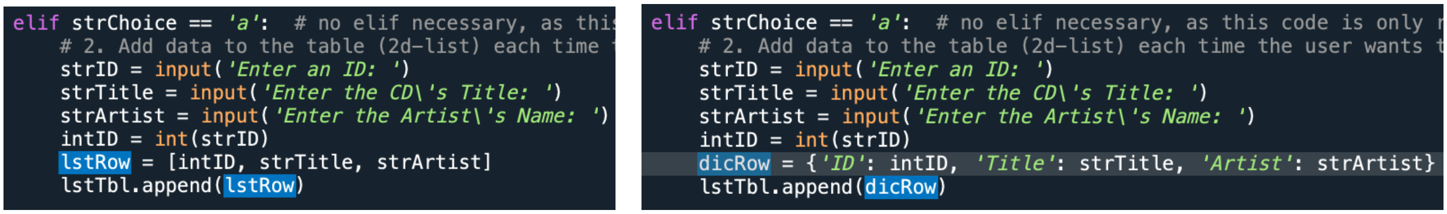
In the variable declaration block of code, this was a simple change - “lstRow=[]” became “dicRow={}”, highlighted in code in figure 2.



```
# Declare variables

strChoice = '' # User input
lstTbl = []  # list of lists to hold data
# TODO replace list of lists with list of dicts
dicRow = {}  # dictionary data row
strFileName = 'CDInventory.txt'  # data storage file
objFile = None  # file object
```

Figure 2 - “dicRow” Shown in Variables Block

Within the remaining code, it was not as simple as replacing “lstRow=[]” with “dicRow={}”, since it requires different code to handle each data structure. To add content to the 2d list, “lstRow” was replaced with “dicRow” and the dictionary structure had to be added, using keys and values. See figure 3 for comparison (before - left, after - right).



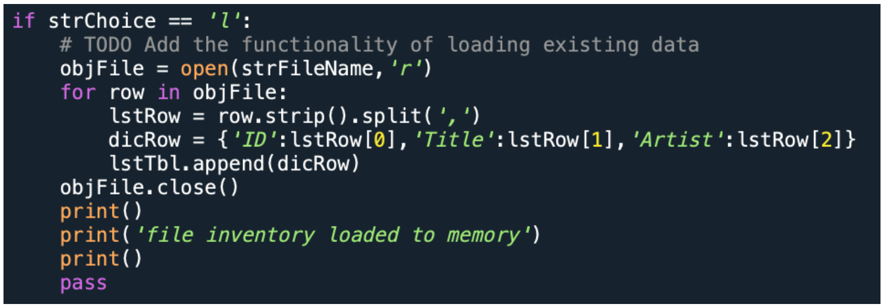
```
elif strChoice == 'a':  # no elif necessary, as thi
    # 2. Add data to the table (2d-list) each time t
    strID = input('Enter an ID: ')
    strTitle = input('Enter the CD\'s Title: ')
    strArtist = input('Enter the Artist\'s Name: ')
    intID = int(strID)
    lstRow = [intID, strTitle, strArtist]
    lstTbl.append(lstRow)
```

```
elif strChoice == 'a':  # no elif necessary, as this code is only r
    # 2. Add data to the table (2d-list) each time the user wants t
    strID = input('Enter an ID: ')
    strTitle = input('Enter the CD\'s Title: ')
    strArtist = input('Enter the Artist\'s Name: ')
    intID = int(strID)
    dicRow = {'ID': intID, 'Title': strTitle, 'Artist': strArtist}
    lstTbl.append(dicRow)
```

Figure 3 - List Structure (left), Dictionary Structure (right)

Add the functionality of loading existing data

With dictionary data structure established, we now have to add functionality to the code, to load data from an existing file. This section of the starter code was a blank block under an ‘if’ loop. To accomplish this code, we have to implement the ‘open()’ function. For each row in row in the file, the data is read into a list row, then the list content is stored in a dictionary row according to its index in the list. Each dictionary row is then stored into the 2d list (table).

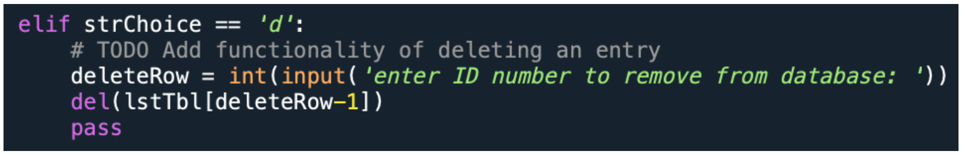


```
if strChoice == 'l':
    # TODO Add the functionality of loading existing data
    objFile = open(strFileName,'r')
    for row in objFile:
        lstRow = row.strip().split(',')
        dicRow = {'ID':lstRow[0],'Title':lstRow[1],'Artist':lstRow[2]}
        lstTbl.append(dicRow)
    objFile.close()
    print()
    print('file inventory loaded to memory')
    print()
    pass
```

Figure 4 - Loading Existing Data Code

Add functionality of deleting an entry

To delete a row from our database, we use the del() function. Each dictionary row has an associated ID number, as well as an associated row index in the table. As long as the list is complete and has an uninterrupted numerical order, we can use ‘deleteRow - 1’ to delete the correct row of data. We have to subtract 1 from the user input, because the row index starts at zero.

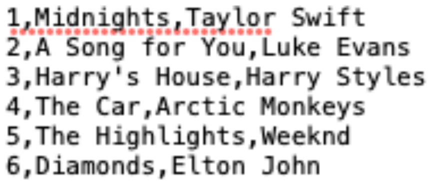


```
elif strChoice == 'd':
    # TODO Add functionality of deleting an entry
    deleteRow = int(input('enter ID number to remove from database: '))
    del(lstTbl[deleteRow-1])
    pass
```

Figure 5 - Deletion Functionality Block of Code

Now, I understand that this methodology will be problematic down the line. Once a row of data is deleted, the numerical order of ID's becomes interrupted, and any subsequent attempts to delete rows of data will most likely not yield the desired result. I didn't have time to figure out how to re-sequence the list after every deletion, and the explicit assignment was only to introduce the ability to delete content.

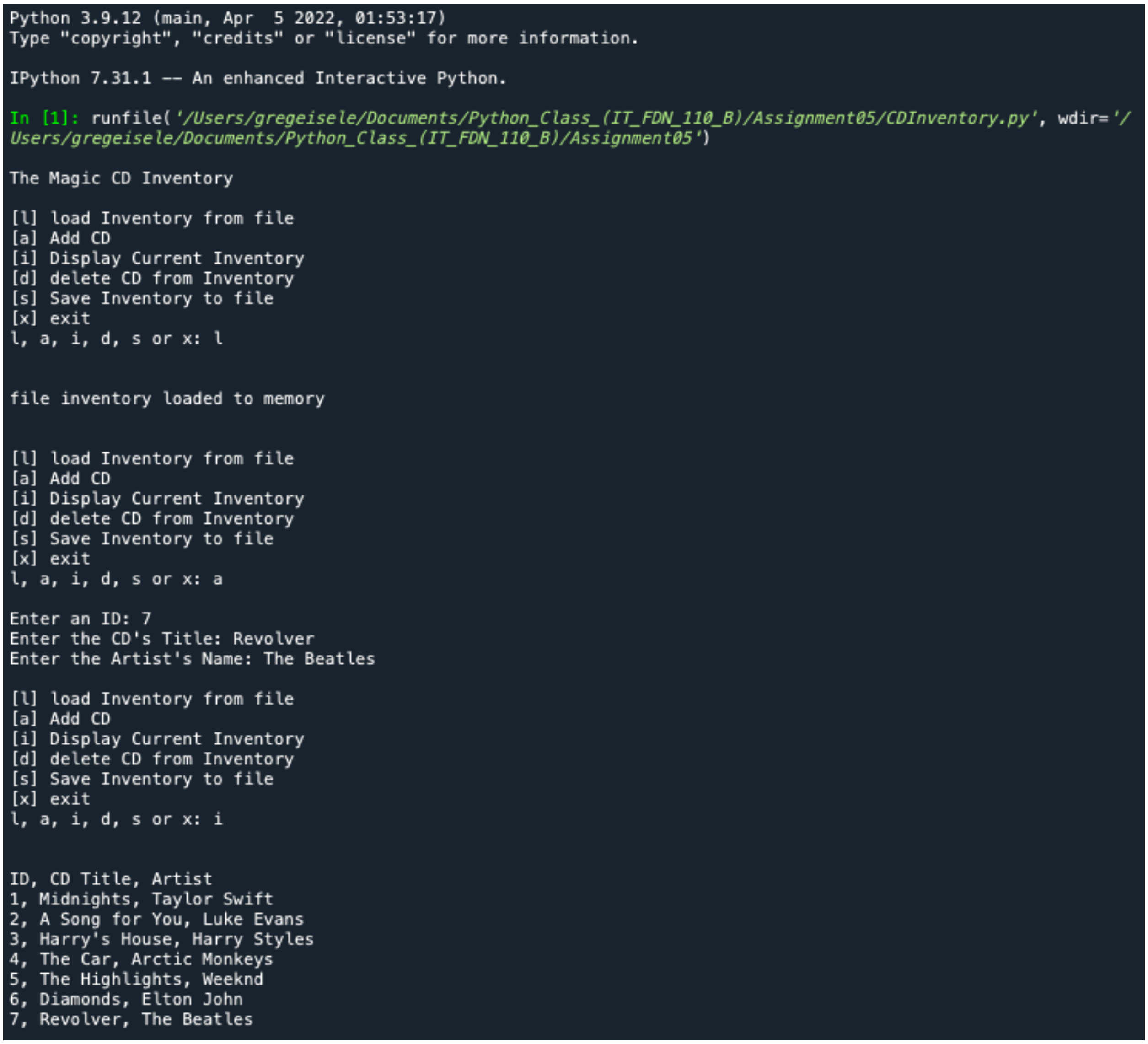
Now - lets see how it works! We'll start with the following list in a text file, and prove out program functionality.



```
1,Midnights,Taylor Swift
2,A Song for You,Luke Evans
3,Harry's House,Harry Styles
4,The Car,Arctic Monkeys
5,The Highlights,Weeknd
6,Diamonds,Elton John
```

Figure 6 - "CDInvetory.txt" Starter List

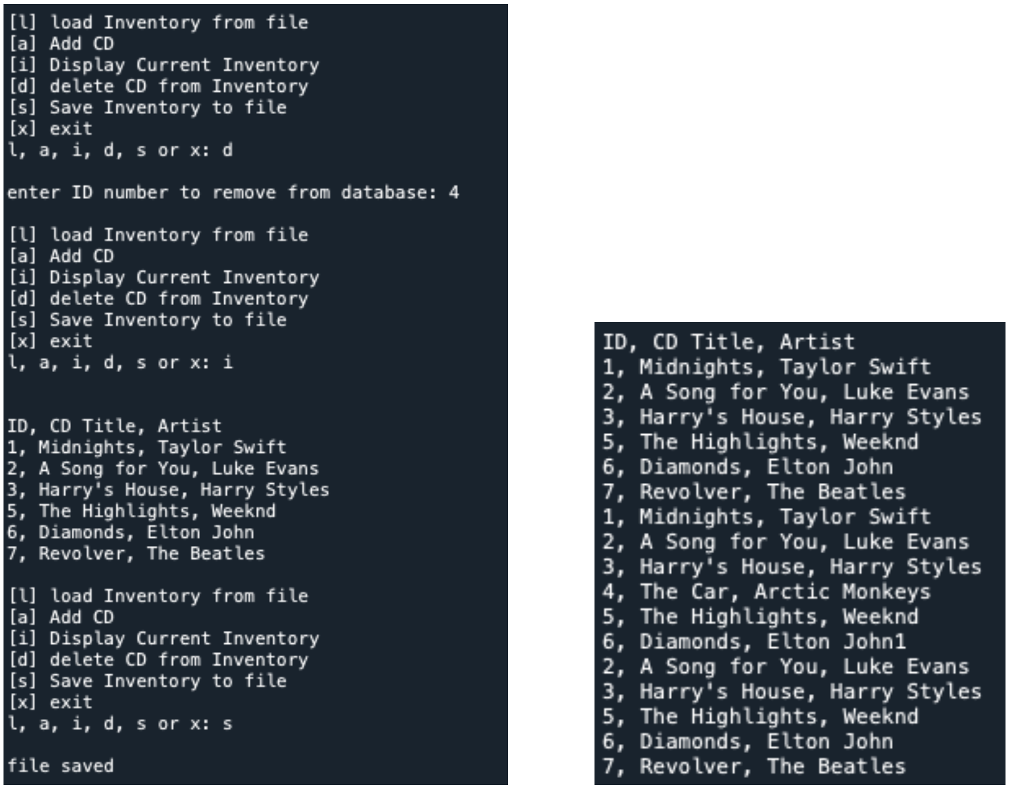
Figure 7 shows the load, add, and display functions of the program working correctly in python.



```
Python 3.9.12 (main, Apr  5 2022, 01:53:17)
Type "copyright", "credits" or "license" for more information.

IPython 7.31.1 -- An enhanced Interactive Python.

In [1]: runfile('/Users/gregeisele/Documents/Python_Class_(IT_FDN_110_B)/Assignment05/CDInventory.py', wdir='/
Users/gregeisele/Documents/Python_Class_(IT_FDN_110_B)/Assignment05')

The Magic CD Inventory

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
l, a, i, d, s or x: l


file inventory loaded to memory


[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
l, a, i, d, s or x: a

Enter an ID: 7
Enter the CD's Title: Revolver
Enter the Artist's Name: The Beatles

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
l, a, i, d, s or x: i


ID, CD Title, Artist
1, Midnights, Taylor Swift
2, A Song for You, Luke Evans
3, Harry's House, Harry Styles
4, The Car, Arctic Monkeys
5, The Highlights, Weeknd
6, Diamonds, Elton John
7, Revolver, The Beatles
```

Figure 7 - Spyder Screen Shot of Load, Add, and Display Functions

Figure 8 shows the delete functionality working, removing ID 4 from the database. Also, the file was saved (appended) with the new data. The saving function works, but the coding needs improvement, to prevent duplicate entries in the list. Currently, it just adds to whats already there, and if you use the load and save function a few times, the list becomes a huge list of redundant entries (see figure 8, right).

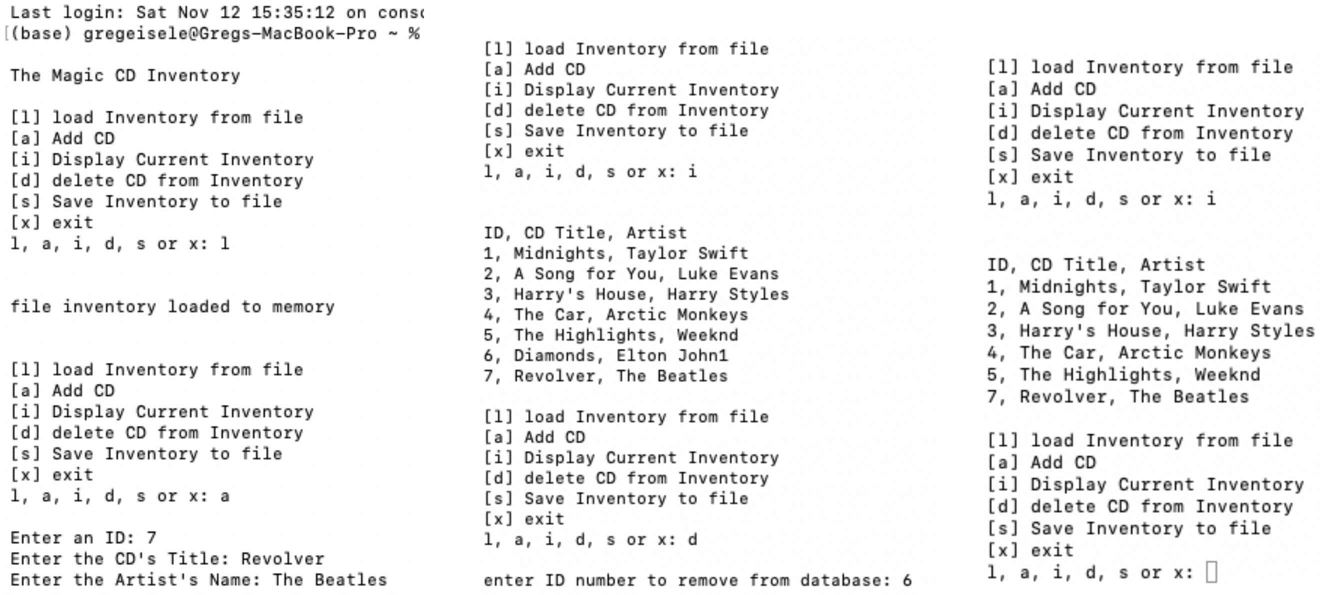


```
[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
l, a, i, d, s or x: d

enter ID number to remove from database: 4

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
l, a, i, d, s or x: i


ID, CD Title, Artist
1, Midnights, Taylor Swift
2, A Song for You, Luke Evans
3, Harry's House, Harry Styles
5, The Highlights, Weeknd
6, Diamonds, Elton John
7, Revolver, The Beatles

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
l, a, i, d, s or x: s

file saved
```

```
ID, CD Title, Artist
1, Midnights, Taylor Swift
2, A Song for You, Luke Evans
3, Harry's House, Harry Styles
5, The Highlights, Weeknd
6, Diamonds, Elton John
7, Revolver, The Beatles
1, Midnights, Taylor Swift
2, A Song for You, Luke Evans
3, Harry's House, Harry Styles
4, The Car, Arctic Monkeys
5, The Highlights, Weeknd
6, Diamonds, Elton John1
2, A Song for You, Luke Evans
3, Harry's House, Harry Styles
5, The Highlights, Weeknd
6, Diamonds, Elton John
7, Revolver, The Beatles
```

Figure 8 - Spyder Screen Shot of Delete and Save Functions (left), Redundant Entries Example (right)

As requested, figure 9 shows the same program functioning in Terminal.



```
Last login: Sat Nov 12 15:35:12 on cons
[(base) gregeisele@Gregs-MacBook-Pro ~ %

The Magic CD Inventory

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
l, a, i, d, s or x: l


file inventory loaded to memory


[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
l, a, i, d, s or x: a

Enter an ID: 7
Enter the CD's Title: Revolver
Enter the Artist's Name: The Beatles
```

```
[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
l, a, i, d, s or x: i


ID, CD Title, Artist
1, Midnights, Taylor Swift
2, A Song for You, Luke Evans
3, Harry's House, Harry Styles
4, The Car, Arctic Monkeys
5, The Highlights, Weeknd
6, Diamonds, Elton John1
7, Revolver, The Beatles

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
l, a, i, d, s or x: d

enter ID number to remove from database: 6
```

```
[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
l, a, i, d, s or x: i


ID, CD Title, Artist
1, Midnights, Taylor Swift
2, A Song for You, Luke Evans
3, Harry's House, Harry Styles
4, The Car, Arctic Monkeys
5, The Highlights, Weeknd
7, Revolver, The Beatles

[l] load Inventory from file
[a] Add CD
[i] Display Current Inventory
[d] delete CD from Inventory
[s] Save Inventory to file
[x] exit
l, a, i, d, s or x: 
```

Figure 9 - Program Working in Terminal

Summary

There wasn’t a huge step chance in difficulty between week 4 and 5 like there was between week 3 and 4, but this assignment was still challenging in its own right. At this point in my coding development, I feel like I liked starting with someone elses code rather than a blank sheet of paper. I’m still learning how to think like a coder, and with frame work provided, it helps remove road blocks I normally have when starting.

Since we’ve now learned tuples, lists, and dictionaries, I’m not quite sure that dictionary data storage is appropriate type for our CD Inventory. It feels much harder to work with as opposed to tuples or lists.

I’m looking forward to learning more tools to deal with the sequencing problem my program currently has when deleting rows, as well as a method to prevent duplicate row content from saving to the data base or file.

Github link

https://github.com/Greg6874/Assignment_05