Sam Ormerod
2020-Aug-19
Foundations of Programming: Python
Assignment 06

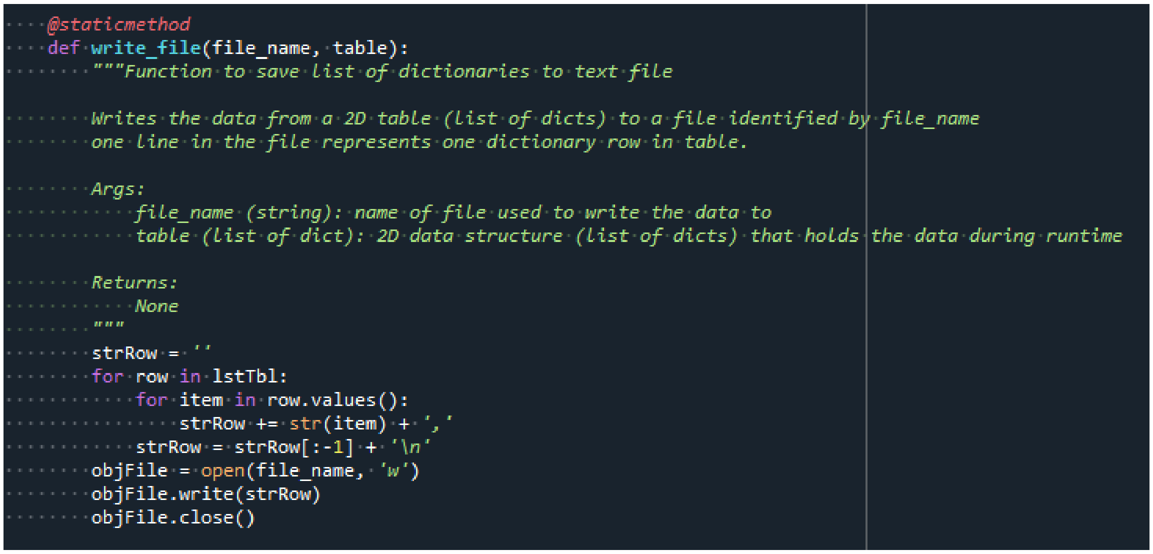
CD Inventory Round 3

Intro

For round 3 we establish a new way to organize the code. This is done by moving the code into defined functions and using call statements to access the said functions. These new concepts were easy enough to comprehend during the labs however incorporating them into our CD inventory program proved to be more challenging than anticipated.

Functions

In module06 we began with identifying what functions[1] and parameters[2] are. In order to utilize them in our programs we will need to organize the code differently because Python requires functions to be defined before they are able to be called.



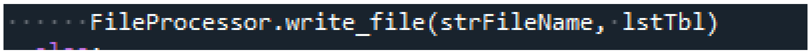
```
    @staticmethod
    def write_file(file_name, table):
        """Function to save list of dictionaries to text file

        Writes the data from a 2D table (list of dicts) to a file identified by file_name
        one line in the file represents one dictionary row in table.

        Args:
            file_name (string): name of file used to write the data to
            table (list of dict): 2D data structure (list of dicts) that holds the data during runtime

        Returns:
            None
        """
        strRow = ''
        for row in lstTbl:
            for item in row.values():
                strRow += str(item) + ','
            strRow = strRow[:-1] + '\n'
        objFile = open(file_name, 'w')
        objFile.write(strRow)
        objFile.close()
```

Figure 1 - Example of a defined function including docstring

[1] FDN_Py_Module_06.pdf page 2
[2] FDN_Py_Module_06.pdf page 3

When defining the functions we also incorporate function document headers, also known as docstrings[3], giving a brief explanation of its purpose. Please see Figure 1 for an example of a defined function with docstring.

In order for us to process that code we simply need to reference a path back to the function and define any arguments if needed, also known as calling the function. Please see Figure 2 for an example of a call referencing our defined function from Figure 1.

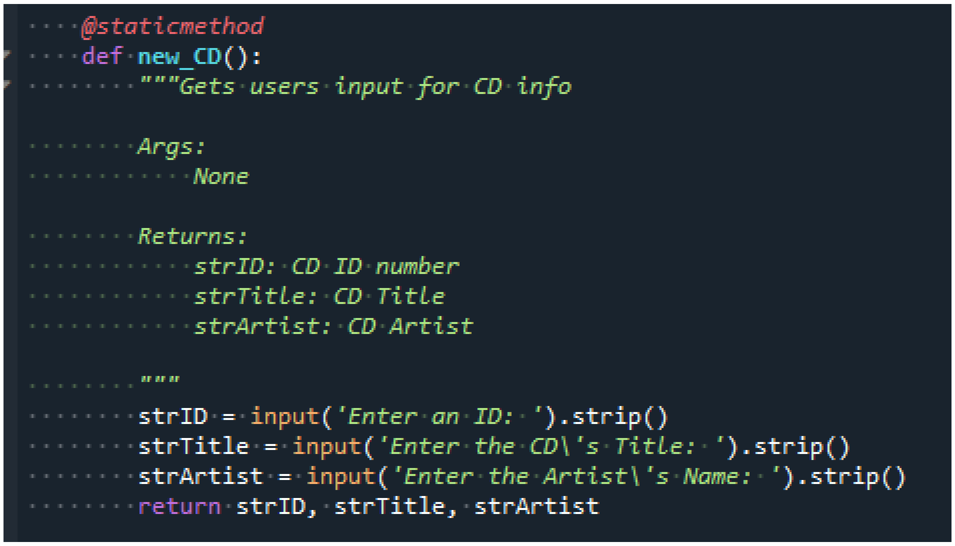


```
      FileProcessor.write_file(strFileName, lstTbl)
```

Figure 2 - Example of a function being called

Assignment

We were given a starter code for our assignment which was similar to our solutions for assignment05. This was intentional for us to experience working in another programmer's code. We needed to reorganize the code by defining functions under predefined classes and utilize calls to execute the code. Comprehension versus execution of the task at hand proved to be the main struggle for me. The first function I attempted to relocate was the user inputs for the CD info. I seemed to overthink it and was unable to execute the code without error. This resulted in me testing it for fifteen to twenty minutes with little to no improvement, rethinking how I wrote the code and repeating my testing process. After my third attempt I revisited and recompleted the labs. This allowed me to simplify what I was attempting to write and execute the code without error. Please see Figure 3.



```
    @staticmethod
    def new_CD():
        """Gets users input for CD info

        Args:
            None

        Returns:
            strID: CD ID number
            strTitle: CD Title
            strArtist: CD Artist

        """
        strID = input('Enter an ID: ').strip()
        strTitle = input('Enter the CD\'s Title: ').strip()
        strArtist = input('Enter the Artist\'s Name: ').strip()
        return strID, strTitle, strArtist
```

Figure 3 - Function to gather user inputs for the CD Info

[3] FDN_Py_Module_06.pdf page 18

I followed the same process to relocate the write function, delete function, and the CD info processing function that converts the CD info into a dictionary and adds that dictionary to the list table.

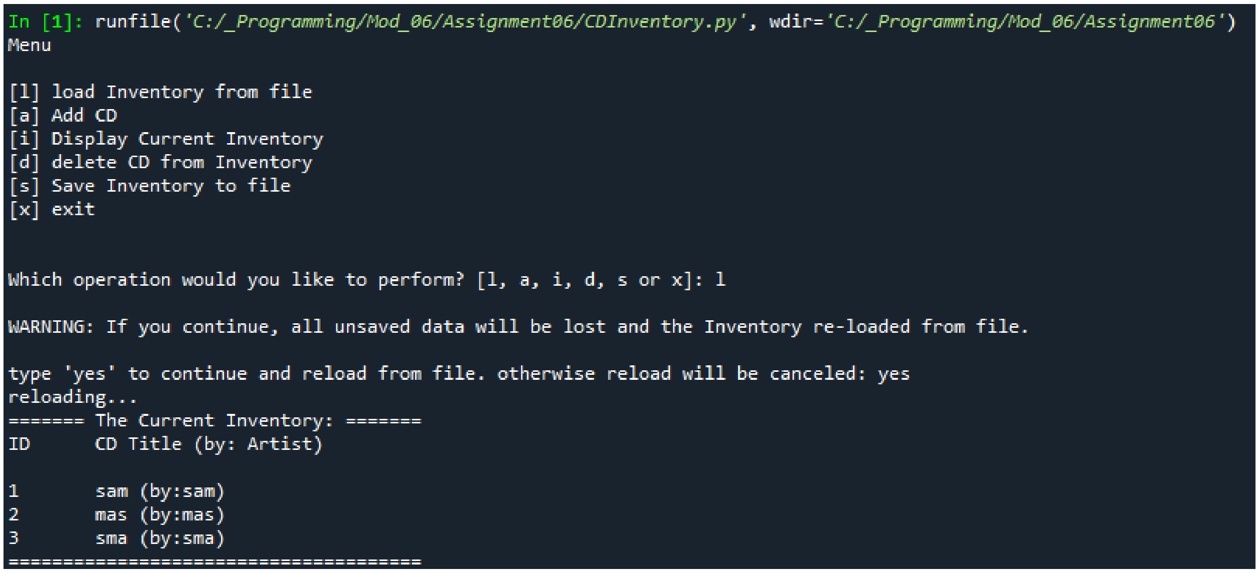
Summary

There are a few benefits when structuring a program in this manner. It can save time when adding functionality in the future, but mainly being able to execute any of these functions wherever and as frequent as I like by implementing a simple call command. It didn't take me long to comprehend organizing my program this way. The execution however took me a good amount of time before I could process without error.

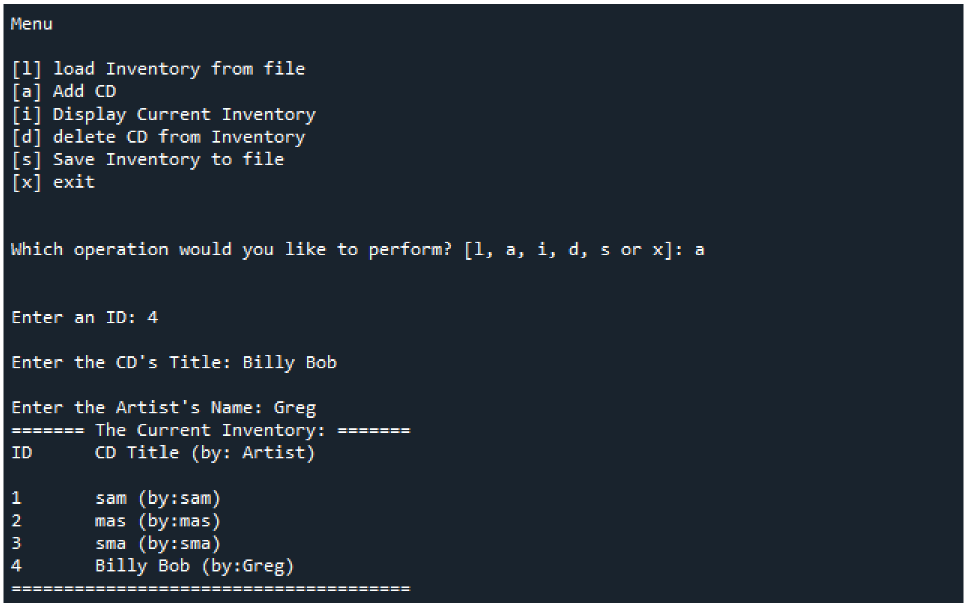
Appendix

Full program available on GitHub

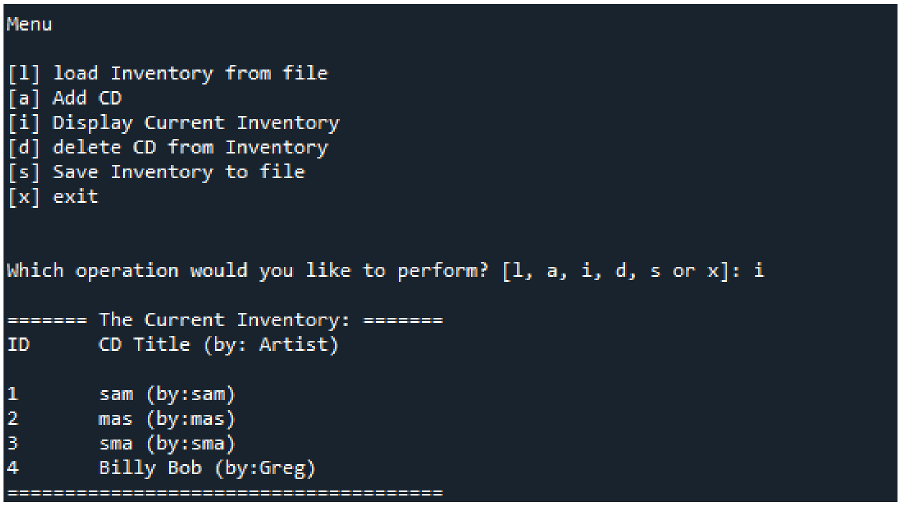
Spyder Screenshots:



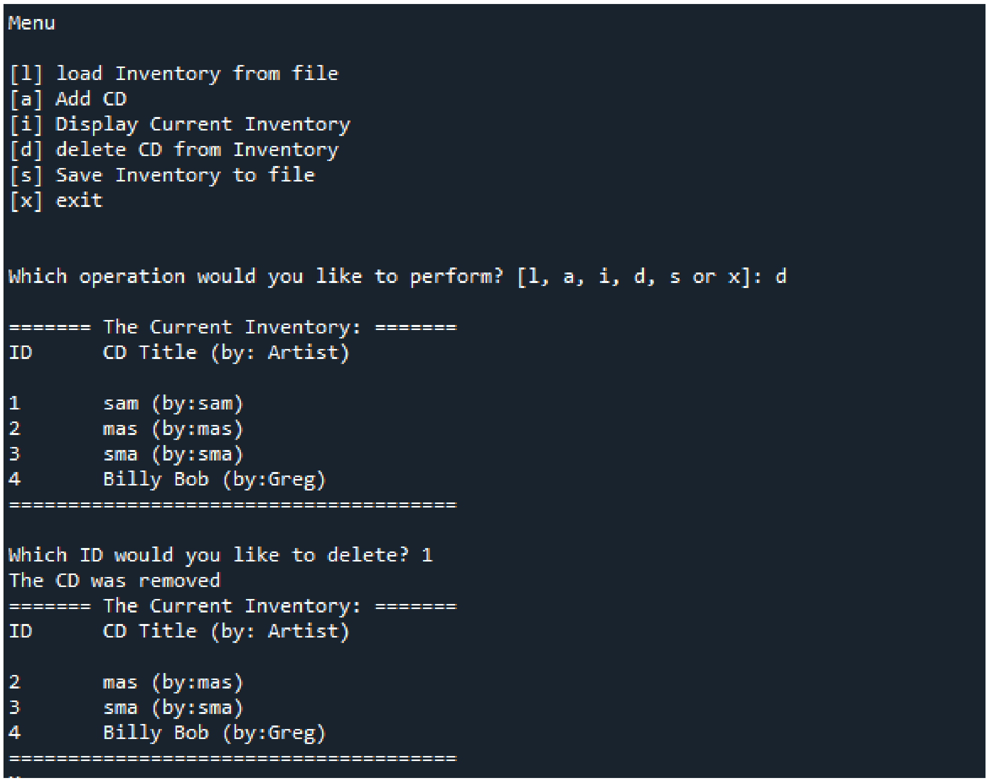
```
In [1]: runfile('C:/_Programming/Mod_06/Assignment06/CDInventory.py', wdir='C:/_Programming/Mod_06/Assignment06')
Menu

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


Which operation would you like to perform? [l, a, i, d, s or x]: l

WARNING: If you continue, all unsaved data will be lost and the Inventory re-loaded from file.

type 'yes' to continue and reload from file. otherwise reload will be canceled: yes
reloading...
======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       sam (by:sam)
2       mas (by:mas)
3       sma (by:sma)
======================================
```



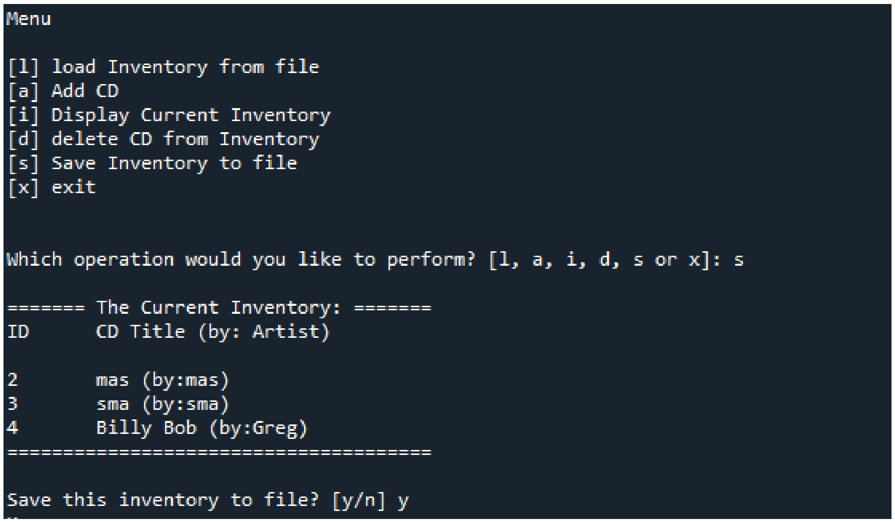
```
Menu

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


Which operation would you like to perform? [l, a, i, d, s or x]: a


Enter an ID: 4

Enter the CD's Title: Billy Bob

Enter the Artist's Name: Greg
======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       sam (by:sam)
2       mas (by:mas)
3       sma (by:sma)
4       Billy Bob (by:Greg)
======================================
```



```
Menu

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


Which operation would you like to perform? [l, a, i, d, s or x]: i

======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       sam (by:sam)
2       mas (by:mas)
3       sma (by:sma)
4       Billy Bob (by:Greg)
======================================
```

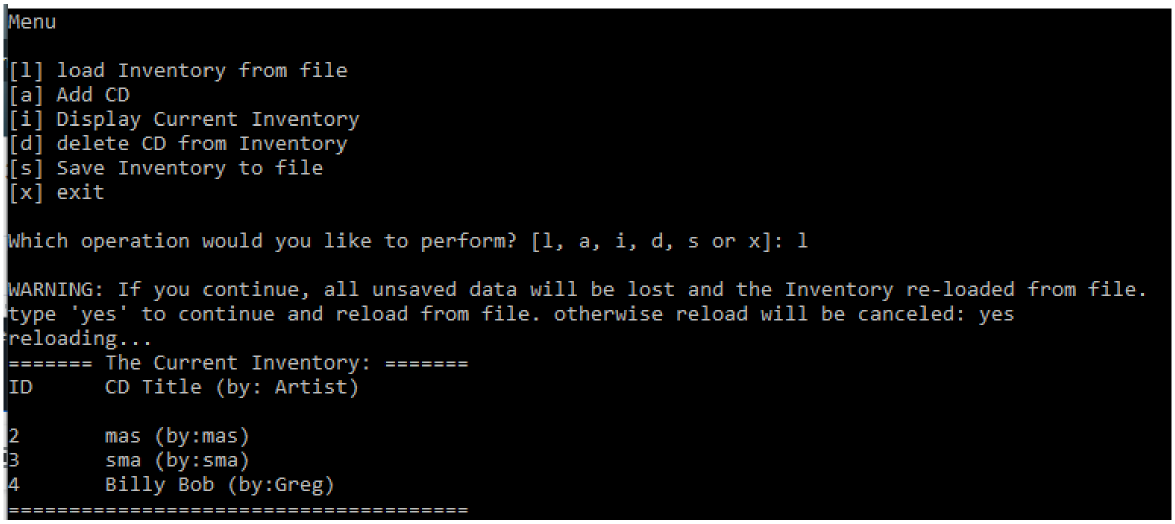


```
Menu

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


Which operation would you like to perform? [l, a, i, d, s or x]: d

======= The Current Inventory: =======
ID      CD Title (by: Artist)

1       sam (by:sam)
2       mas (by:mas)
3       sma (by:sma)
4       Billy Bob (by:Greg)
======================================

Which ID would you like to delete? 1
The CD was removed
======= The Current Inventory: =======
ID      CD Title (by: Artist)

2       mas (by:mas)
3       sma (by:sma)
4       Billy Bob (by:Greg)
======================================
```

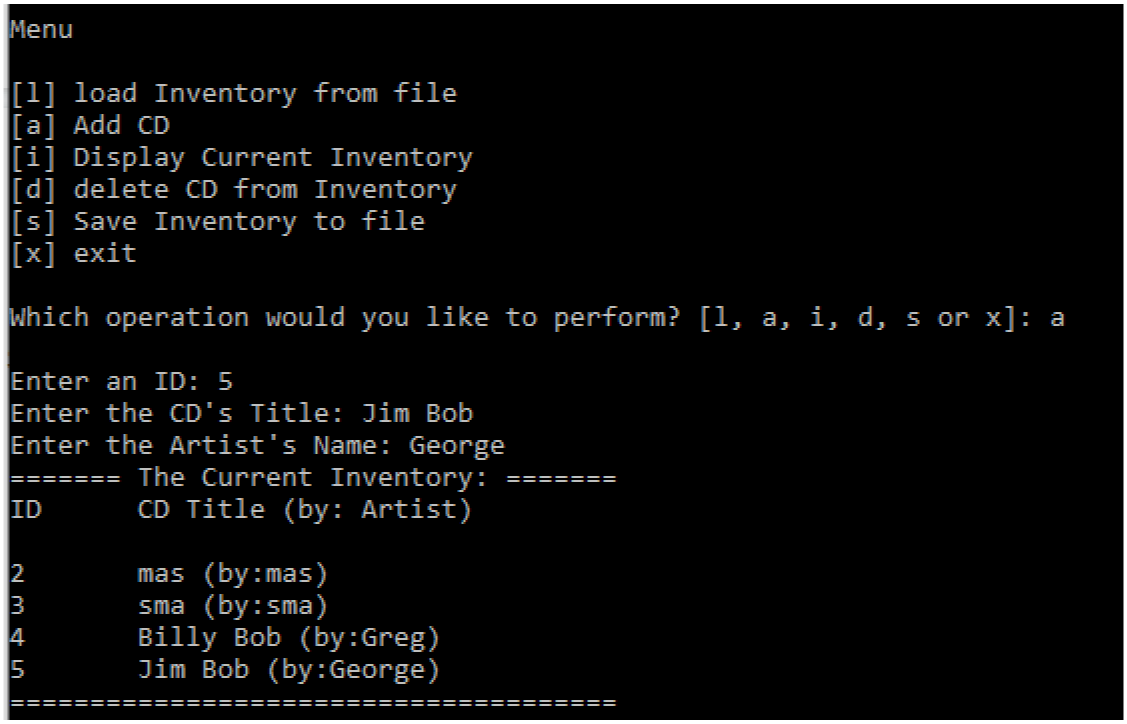


```
Menu

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


Which operation would you like to perform? [l, a, i, d, s or x]: s

======= The Current Inventory: =======
ID      CD Title (by: Artist)

2       mas (by:mas)
3       sma (by:sma)
4       Billy Bob (by:Greg)
======================================

Save this inventory to file? [y/n] y
```

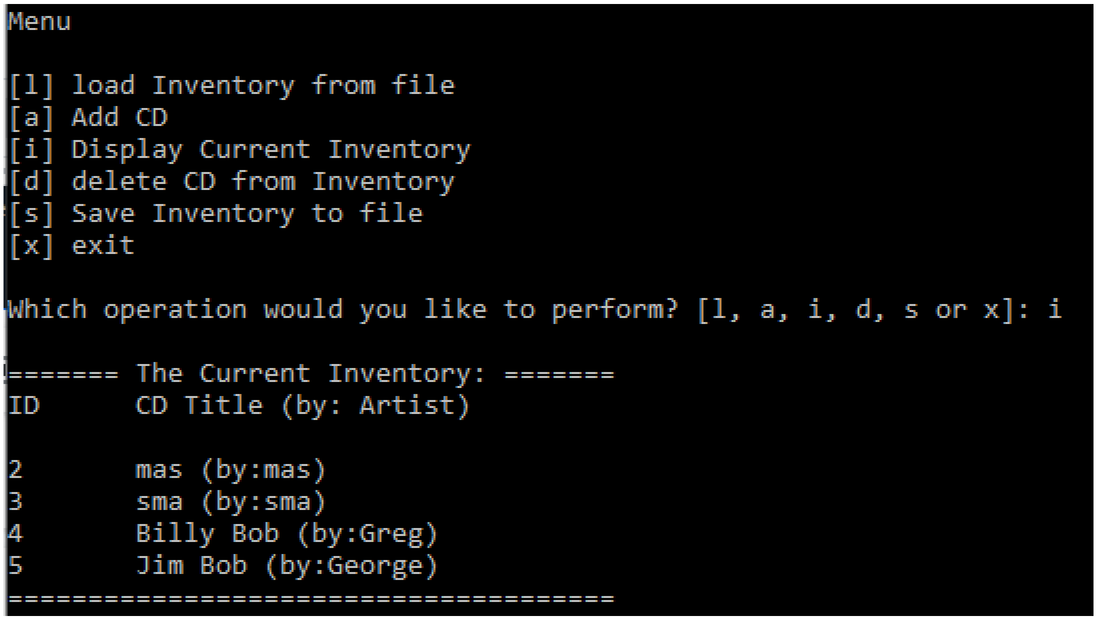
Terminal Screenshots:



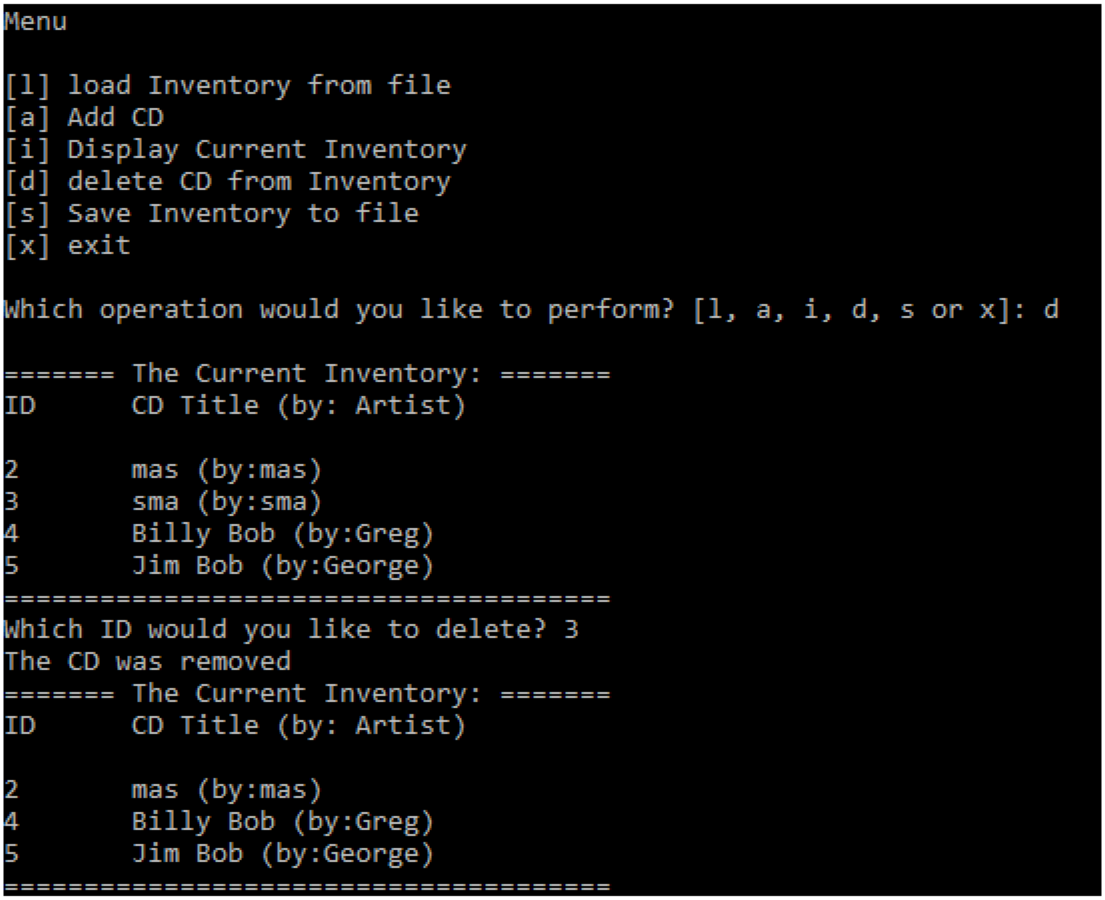
```
Menu

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

Which operation would you like to perform? [l, a, i, d, s or x]: l

WARNING: If you continue, all unsaved data will be lost and the Inventory re-loaded from file.
type 'yes' to continue and reload from file. otherwise reload will be canceled: yes
reloading...
======= The Current Inventory: =======
ID      CD Title (by: Artist)

2       mas (by:mas)
3       sma (by:sma)
4       Billy Bob (by:Greg)
======================================
```



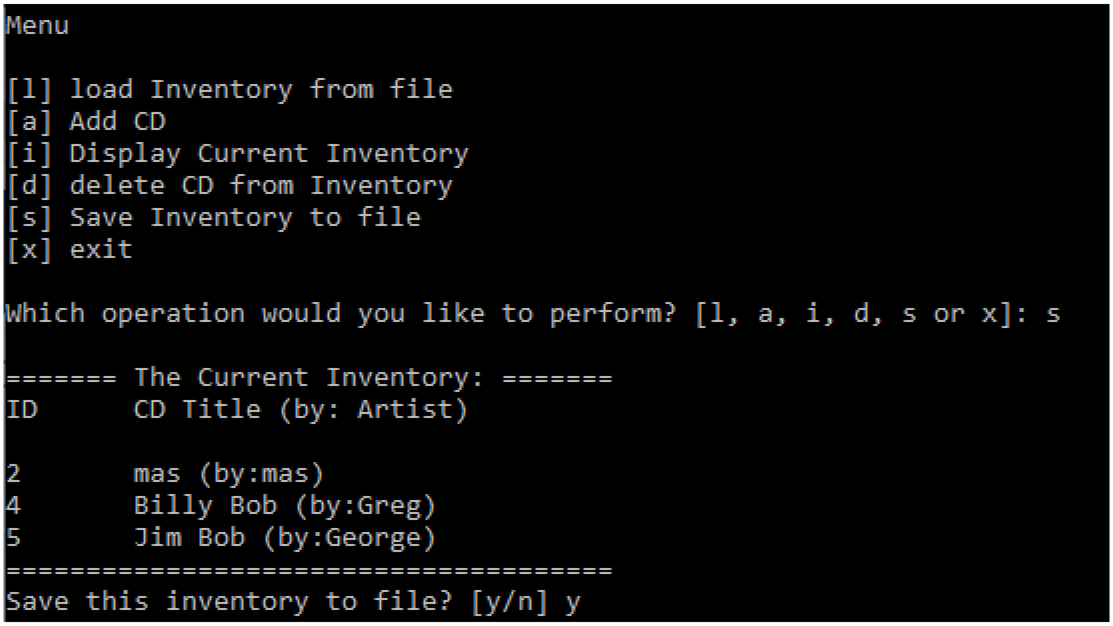
```
Menu

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

Which operation would you like to perform? [l, a, i, d, s or x]: a

Enter an ID: 5
Enter the CD's Title: Jim Bob
Enter the Artist's Name: George
======= The Current Inventory: =======
ID      CD Title (by: Artist)

2       mas (by:mas)
3       sma (by:sma)
4       Billy Bob (by:Greg)
5       Jim Bob (by:George)
======================================
```



```
Menu

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

Which operation would you like to perform? [l, a, i, d, s or x]: i

======= The Current Inventory: =======
ID      CD Title (by: Artist)

2       mas (by:mas)
3       sma (by:sma)
4       Billy Bob (by:Greg)
5       Jim Bob (by:George)
======================================
```



```
Menu

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

Which operation would you like to perform? [l, a, i, d, s or x]: d

======= The Current Inventory: =======
ID      CD Title (by: Artist)

2       mas (by:mas)
3       sma (by:sma)
4       Billy Bob (by:Greg)
5       Jim Bob (by:George)
======================================
Which ID would you like to delete? 3
The CD was removed
======= The Current Inventory: =======
ID      CD Title (by: Artist)

2       mas (by:mas)
4       Billy Bob (by:Greg)
5       Jim Bob (by:George)
======================================
```



```
Menu

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

Which operation would you like to perform? [l, a, i, d, s or x]: s

======= The Current Inventory: =======
ID      CD Title (by: Artist)

2       mas (by:mas)
4       Billy Bob (by:Greg)
5       Jim Bob (by:George)
======================================
Save this inventory to file? [y/n] y
```