

Journal/Blog Module 06

1. Summarize what you've learned about the topics in Module 6.
2. Did you try to draw the connections of your linked lists? Was it easier to trace?
3. Why should you not use the type-definition declaration of a structure when creating self-referential structures for linked lists?
4. Why is the head pointer important?
5. What will happen if you forget to assign NULL to an empty list or if you don't put NULL at the last node of the linked list?
6. Do you think linked lists are better than arrays? Why or why not?
7. What problems have you encountered in answering the exercise? How were you able to resolve them?
8. Have you encountered segmentation fault in linked lists? If you haven't, list some reasons (at least two) why a segmentation fault would occur.

Exercise 06

Create a program that implements a music playlist using linked lists that is sorted by playlist name. The program must continue to run until the user picks exit.

Functionalities:

1. Each playlist and song node is defined by the structure below:

```
typedef struct song_tag {
    char title[20];
    char artist[20];
    char album[20];
    struct song_tag *nextSong;
} song;

typedef struct playlist_tag {
    char name[50];
    int songCount;
    song *songHead;
    struct playlist_tag *next;
} playlist;
```

NOTE: You can add additional members that you think may help you finish the exercise.

2. The program must load the data each time it is ran and must save the data each time the user picks exit.
 - (a) **Save Data**
 - You must check whether there is a data to be saved. If there are none, simply print a prompt saying so. Also, save count as 0 so that you will be able to keep track if file is empty.
 - You can use whatever format of data you want as long as you will be able to load the data properly.
 - (b) **Load Data**
 - You must check whether the file exists. If the file exists, check if the count is 0. If the count is not 0, read each of the data in the file and save it in the linked lists.
3. Create a menu that shows:
 - (a) **Add Playlist**
 - Create an instance of the playlist and ask for the details. Make sure to initialize the members properly. Remember to check that the name of the playlist is unique. If it already exists, it musn't be added to the system.
 - The new playlist node must be added to the linked list in such a way that the list remains **sorted by playlist name (alphabetical)**.
 - (b) **Add Song to Playlist**
 - Make sure to check if there a playlist to add to. Ask which playlist to add the song to. If the playlist is found, create an instance of the song and ask for the details. Make sure to initialize the members properly.
 - The new song node must be added to the linked list in such a way that the list remains **sorted by song title (alphabetical)**.

(c) **Remove Song from Playlist**

- Make sure to check if there is a playlist to remove from. Ask which playlist to remove songs from. If the playlist exists, and it isn't empty, ask which song to remove using the song title and artist. If found, remove the song. Else, print the appropriate prompts.

(d) **View a Playlist**

- Make sure to check if there is a playlist to print. Ask which playlist to print. If the playlist exists, print all the details about it along with the songs. Else, print the appropriate prompts.

(e) **View all Data**

- This simply prints all of the available data in the system. Both the playlists and songs. Make sure to check if there is data to print.

(f) **Exit**

- When the user picks this, the program terminates. However, before ending the program, **all dynamically-allocated variables must be freed first**.

4. Each choices except for the exit must be implemented inside a function.

5. Do not forget to put documentation in your code.

Sample run:

```

===== MENU =====
[1] Add Playlist
[2] Add Song to Playlist
[3] Remove Song from Playlist
[4] View a Playlist
[5] View All Data
[6] Exit

Enter choice: 3

There are no playlists yet!
-----
===== MENU =====
[1] Add Playlist
[2] Add Song to Playlist
[3] Remove Song from Playlist
[4] View a Playlist
[5] View All Data
[6] Exit

Enter choice: 1

Enter playlist name: Taylor <3

Successfully added playlist!
-----
===== MENU =====
[1] Add Playlist
[2] Add Song to Playlist
[3] Remove Song from Playlist
[4] View a Playlist
[5] View All Data
[6] Exit

Enter choice: 1

Enter playlist name: Favorites <3

Successfully added playlist!
-----

```

```
===== MENU =====
[1] Add Playlist
[2] Add Song to Playlist
[3] Remove Song from Playlist
[4] View a Playlist
[5] View All Data
[6] Exit

Enter choice: 1

Enter playlist name: Taylor <3

Playlist name already exists!
-----
===== MENU =====
[1] Add Playlist
[2] Add Song to Playlist
[3] Remove Song from Playlist
[4] View a Playlist
[5] View All Data
[6] Exit

Enter choice: 5

PLAYLIST: Taylor <3
SONG COUNT: 0

PLAYLIST: Favorites <3
SONG COUNT: 0
-----
===== MENU =====
[1] Add Playlist
[2] Add Song to Playlist
[3] Remove Song from Playlist
[4] View a Playlist
[5] View All Data
[6] Exit

Enter choice: 2

THE PLAYLISTS AVAILABLE ARE:
    [1] Taylor <3
    [2] Favorites <3

Enter playlist name: KPOP

Invalid playlist!
-----
===== MENU =====
[1] Add Playlist
[2] Add Song to Playlist
[3] Remove Song from Playlist
[4] View a Playlist
[5] View All Data
[6] Exit

Enter choice: 2

THE PLAYLISTS AVAILABLE ARE:
    [1] Taylor <3
    [2] Favorites <3
```

```
Enter playlist name: Taylor <3

Enter song title: Sparks Fly
Enter song artist: Taylor Swift
Enter song album: Speak Now

Successfully added song to playlist!
-----
===== MENU =====
[1] Add Playlist
[2] Add Song to Playlist
[3] Remove Song from Playlist
[4] View a Playlist
[5] View All Data
[6] Exit

Enter choice: 2

THE PLAYLISTS AVAILABLE ARE:
    [1] Taylor <3
    [2] Favorites <3

Enter playlist name: Taylor <3

Enter song title: Invisible String
Enter song artist: Taylor Swift
Enter song album: folklore

Successfully added song to playlist!
-----
===== MENU =====
[1] Add Playlist
[2] Add Song to Playlist
[3] Remove Song from Playlist
[4] View a Playlist
[5] View All Data
[6] Exit

Enter choice: 5

PLAYLIST: Taylor <3
SONG COUNT: 2
    SONG TITLE: Sparks Fly
    SONG ARTIST: Taylor Swift
    SONG ALBUM: Speak Now

    SONG TITLE: Invisible String
    SONG ARTIST: Taylor Swift
    SONG ALBUM: folklore

PLAYLIST: Favorites <3
SONG COUNT: 0
-----
===== MENU =====
[1] Add Playlist
[2] Add Song to Playlist
[3] Remove Song from Playlist
[4] View a Playlist
[5] View All Data
[6] Exit
```

```
Enter choice: 4

THE PLAYLISTS AVAILABLE ARE:
  [1] Taylor <3
  [2] Favorites <3

Enter playlist name: Taylor <3

PLAYLIST: Taylor <3
SONG COUNT: 2
  SONG TITLE: Sparks Fly
  SONG ARTIST: Taylor Swift
  SONG ALBUM: Speak Now

  SONG TITLE: Invisible String
  SONG ARTIST: Taylor Swift
  SONG ALBUM: folklore
-----
===== MENU =====
[1] Add Playlist
[2] Add Song to Playlist
[3] Remove Song from Playlist
[4] View a Playlist
[5] View All Data
[6] Exit

Enter choice: 3

THE PLAYLISTS AVAILABLE ARE:
  [1] Taylor <3
  [2] Favorites <3

Enter playlist name: Favorites <3
Playlist is still empty!
-----
===== MENU =====
[1] Add Playlist
[2] Add Song to Playlist
[3] Remove Song from Playlist
[4] View a Playlist
[5] View All Data
[6] Exit

Enter choice: 3

THE PLAYLISTS AVAILABLE ARE:
  [1] Taylor <3
  [2] Favorites <3

Enter playlist name: Taylor <3

THE SONGS AVAILABLE ARE:
  [1] Sparks Fly by Taylor Swift
  [2] Invisible String by Taylor Swift

Enter song title to delete: Sparks Fly
Enter song artist to delete: Taylor Sweep

Song not found!
-----
```

```
===== MENU =====
[1] Add Playlist
[2] Add Song to Playlist
[3] Remove Song from Playlist
[4] View a Playlist
[5] View All Data
[6] Exit

Enter choice: 3

THE PLAYLISTS AVAILABLE ARE:
  [1] Taylor <3
  [2] Favorites <3

Enter playlist name: Taylor <3

THE SONGS AVAILABLE ARE:
  [1] Sparks Fly by Taylor Swift
  [2] Invisible String by Taylor Swift

Enter song title to delete: Sparks Fly
Enter song artist to delete: Taylor Swift

Successfully deleted song!
-----
===== MENU =====
[1] Add Playlist
[2] Add Song to Playlist
[3] Remove Song from Playlist
[4] View a Playlist
[5] View All Data
[6] Exit

Enter choice: 3

THE PLAYLISTS AVAILABLE ARE:
  [1] Taylor <3
  [2] Favorites <3

Enter playlist name: Taylor <3

THE SONGS AVAILABLE ARE:
  [1] Invisible String by Taylor Swift

Enter song title to delete: Invisible String
Enter song artist to delete: Taylor Swift

Successfully deleted song!
-----
===== MENU =====
[1] Add Playlist
[2] Add Song to Playlist
[3] Remove Song from Playlist
[4] View a Playlist
[5] View All Data
[6] Exit

Enter choice: 5

PLAYLIST: Taylor <3
SONG COUNT: 0
```

```
PLAYLIST: Favorites <3
```

```
SONG COUNT: 0
```

```
-----
```