♫ Assignment #3 - musicalLL ♫

Expected outputs and Execution Guide

Deliverables:

1) **makefile**
2) **Src folder** *containing all function-specific files plus mainA3.c (total 9)*
3) **Include folder** *containing givenA3.h file*
4) **Bin folder** *to store executable file (i.e. musicalLL)*

For more information, refer to section *4.0 Submission* of Assignment 3 description.

NOTE:

1) There will be exactly ***21* notes** for each song in the dataset.
2) Song names need ***not*** to be unique.
3) Song ID must be ***unique***.
4) Remember to use srand() with rand().
5) Bin can be *empty* before compilation.
6) The main target in your makefile must be called *musicalLL*.
7) Recurring menu similar to A2 *(But with a **level up** ⬆, see scenarios)*.
8) ***createPlayList*.c** is acceptable (same applies to every other file name).
9) ***CreatePlayList*.c** is **not** acceptable ((same applies to every other file name).
10) Understanding of ***pointers***, *double* pointers and ***linked lists*** is required.

Code Compilation:

```
dkumar07@linux-05:~/cis2500A3$ make
```

Code execution:

```
dkumar07@linux-05:~/cis2500A3$ ./bin/musicalLL musicalNotes.csv
```

SCENARIOS

After Execution, Menu should be displayed:

```
=====================================================
1. Create a new playlist
2. Add a new song to an existing playlist
3. Play all songs in the given playlist
4. Play a song from the playlist, given its id
5. Play a song from the playlist, given its name
6. Count the number of occurrences of a note in a given song
7. Delete a song from the playlist, given its id
8. Delete the entire playlist
9. Exit
=====================================================
Enter your choice: 1
```

As per the Assignment ethics, User always attempts to choose option #1 on any circumstances before moving on to the rest.

```
Enter your choice: 1
Number of songs in the playlist: 100

=====================================================
1. Create a new playlist
2. Add a new song to an existing playlist
3. Play all songs in the given playlist
4. Play a song from the playlist, given its id
5. Play a song from the playlist, given its name
6. Count the number of occurrences of a note in a given song
7. Delete a song from the playlist, given its id
8. Delete the entire playlist
9. Exit
=====================================================
Enter your choice: |
```

NOTE:

- You can assume that choice #1 will **not** be run again in this assignment.
- Make sure to have proper implementation of **Linked List.**

Menu Choice 2:

Adds a new song to the playlist. *beginOrEnd* = [1,2].
Here, print statements are optional - have only been added for understanding purposes.
The Function's job is to return *True* for success or *False* when failure.

```
=====================================================
Enter your choice: 2
Enter the value for beginOrEnd: 2
Song Name: Not like us

Song ID: 275
Song Name: Not like us
Notes: fa do fa do re sol ti la sol mi mi ti fa fa la re do ti la
do sol

Song added at the end!

=====================================================
1. Create a new playlist
2. Add a new song to an existing playlist
3. Play all songs in the given playlist
4. Play a song from the playlist, given its id
5. Play a song from the playlist, given its name
6. Count the number of occurrences of a note in a given song
7. Delete a song from the playlist, given its id
8. Delete the entire playlist
9. Exit
=====================================================
Enter your choice: 2
Enter the value for beginOrEnd: 1
Song Name: Replay

Song added at the beginning!
```

Required Validations:

```
=====================================================
Enter your choice: 2
Enter the value for beginOrEnd: 3
Song Name: Saturn
```

```
Invalid beginOrEnd value passed!!.

=====================================================
1. Create a new playlist
2. Add a new song to an existing playlist
3. Play all songs in the given playlist
4. Play a song from the playlist, given its id
5. Play a song from the playlist, given its name
6. Count the number of occurrences of a note in a given song
7. Delete a song from the playlist, given its id
8. Delete the entire playlist
9. Exit
=====================================================
Enter your choice:
```

When a user attempts to add a new song, same as a pre-existing song name.
The song Id generated must be different.

```
=====================================================
Enter your choice: 2
Enter the value for beginOrEnd: 2
Song Name: Not like us

Song ID: 92
Song Name: Not like us
Song Notes: la re sol sol la sol la do mi do do re ti fa sol ti fa
sol fa la do

Song added at the end

=====================================================
1. Create a new playlist
2. Add a new song to an existing playlist
3. Play all songs in the given playlist
4. Play a song from the playlist, given its id
5. Play a song from the playlist, given its name
6. Count the number of occurrences of a note in a given song
7. Delete a song from the playlist, given its id
8. Delete the entire playlist
9. Exit
=====================================================
Enter your choice:
```

Level up ⬆️

Level up your defensive-programming by validating the menu choice.
Inputs other than *int* datatype are **Invalid**.
Make sure to indicate whether it is ***Invalid Choice*** or ***Invalid Type***.

```
=====================================================
1. Create a new playlist
2. Add a new song to an existing playlist
3. Play all songs in the given playlist
4. Play a song from the playlist, given its id
5. Play a song from the playlist, given its name
6. Count the number of occurrences of a note in a given song
7. Delete a song from the playlist, given its id
8. Delete the entire playlist
9. Exit
=====================================================
Enter your choice: one
Invalid Type!!
=====================================================
1. Create a new playlist
2. Add a new song to an existing playlist
3. Play all songs in the given playlist
4. Play a song from the playlist, given its id
5. Play a song from the playlist, given its name
6. Count the number of occurrences of a note in a given song
7. Delete a song from the playlist, given its id
8. Delete the entire playlist
9. Exit
=====================================================
Enter your choice: music
Invalid Type!!
=====================================================
1. Create a new playlist
2. Add a new song to an existing playlist
3. Play all songs in the given playlist
4. Play a song from the playlist, given its id
5. Play a song from the playlist, given its name
6. Count the number of occurrences of a note in a given song
7. Delete a song from the playlist, given its id
8. Delete the entire playlist
9. Exit
=====================================================
Enter your choice: 12
```

```
Invalid Choice!!
=====================================================
1. Create a new playlist
2. Add a new song to an existing playlist
3. Play all songs in the given playlist
4. Play a song from the playlist, given its id
5. Play a song from the playlist, given its name
6. Count the number of occurrences of a note in a given song
7. Delete a song from the playlist, given its id
8. Delete the entire playlist
9. Exit
=====================================================
Enter your choice:
```

Menu Choice 3:

Prints all the songs present in the playlist.
Recommended way to print the songs.
The Notes should be printed out on one line and separated by a “.” character.
Below are some songs picked from the playlist.

```
Song ID: 245
Song Name: Whispering Symphony
Song Notes:
sol.mi.mi.do.sol.mi.mi.do.do.fa.sol.mi.mi.mi.do.sol.ti.ti.mi.la.sol
.sol.la.sol.la.la.re.sol

Song ID: 597
Song Name: Rustic Saga
Notes:
re.fa.mi.la.fa.ti.fa.do.sol.do.do.ti.do.fa.re.sol.mi.mi.la.do.do.fa
.mi

Song ID: 110
Song Name: Distant Memories
Notes:
mi.do.re.sol.ti.ti.ti.sol.sol.do.sol.do.do.mi.fa.sol.mi.mi.sol.ti.t
i.do.mi.do.ti.ti.sol

Song ID: 554
Song Name: Rustic Symphony
Notes:
la.do.re.re.fa.do.re.re.ti.sol.do.do.do.mi.do.la.ti.re.la.do.ti.fa
```

```
Song ID: 275
Song Name: Not like us
Notes:
fa.do.fa.do.re.sol.ti.ti.la.sol.mi.mi.mi.ti.fa.fa.la.re.do.ti.la.do
.sol

Song ID: 92
Song Name: Not like us
Notes:
la.re.sol.sol.la.sol.la.la.sol.la.la.do.mi.do.do.re.ti.fa.sol.ti.ti
.fa.sol.fa.fa.la.do
```

Menu Choice 4:

Play a song given its id. You can print the song's details in the function itself.

```
Enter your choice: 4
Enter the id of the song you want to play: 110

Song ID: 110
Song Name: Distant Memories
Song Notes:
mi.do.re.sol.ti.ti.ti.sol.sol.do.sol.do.do.mi.fa.sol.mi.mi.sol.ti.
ti.do.mi.do.ti.ti.sol

========================================================
1. Create a new playlist
2. Add a new song to an existing playlist
3. Play all songs in the given playlist
4. Play a song from the playlist, given its id
5. Play a song from the playlist, given its name
6. Count the number of occurrences of a note in a given song
7. Delete a song from the playlist, given its id
8. Delete the entire playlist
9. Exit
========================================================
Enter your choice: 554
Invalid Choice!!

========================================================
1. Create a new playlist
```

```
2. Add a new song to an existing playlist
3. Play all songs in the given playlist
4. Play a song from the playlist, given its id
5. Play a song from the playlist, given its name
6. Count the number of occurrences of a note in a given song
7. Delete a song from the playlist, given its id
8. Delete the entire playlist
9. Exit
=====================================================
Enter your choice: 4
Enter the id of the song you want to play: 554

Song ID: 554
Song Name: Rustic Symphony
Song Notes:
la.do.re.re.fa.do.re.re.ti.sol.do.do.do.mi.do.la.ti.re.la.do.ti.fa

=====================================================
```

Required Validation:

You **Don't** necessarily need to ask the user again if the entered song ID is not present. Return the value accordingly.

```
=====================================================
Enter your choice: 4
Enter the id of the song you want to play: 23

No song found!!

=====================================================
1. Create a new playlist
2. Add a new song to an existing playlist
3. Play all songs in the given playlist
4. Play a song from the playlist, given its id
5. Play a song from the playlist, given its name
6. Count the number of occurrences of a note in a given song
7. Delete a song from the playlist, given its id
8. Delete the entire playlist
9. Exit
=====================================================
Enter your choice:
```

Menu Choice 5:

Play the song from the playlist, given its **name**.
If there are two songs present with the **same name** then, it should play the song which is **first** in the list.
Validation for song name should be done in a ***case-insensitive*** manner. See below.

```
=========================================================
Enter your choice: 5
Enter the name of the song you want to play: not like us

Song ID: 275
Song Name: Not like us
Song Notes:
la.re.sol.mi.mi.ti.do.re.re.re.mi.ti.ti.ti.fa.fa.re.ti.re.mi.
fa.la

=========================================================
1. Create a new playlist
2. Add a new song to an existing playlist
3. Play all songs in the given playlist
4. Play a song from the playlist, given its id
5. Play a song from the playlist, given its name
6. Count the number of occurrences of a note in a given song
7. Delete a song from the playlist, given its id
8. Delete the entire playlist
9. Exit
=========================================================
Enter your choice: 5
Enter the name of the song you want to play: distant memories

Song ID: 110
Song Name: Distant Memories
Song Notes:
mi.do.re.sol.ti.ti.ti.sol.sol.do.sol.do.do.mi.fa.sol.mi.mi.so
l.ti.ti.do.mi.do.ti.ti.sol

=========================================================
1. Create a new playlist
2. Add a new song to an existing playlist
3. Play all songs in the given playlist
```

```
4. Play a song from the playlist, given its id
5. Play a song from the playlist, given its name
6. Count the number of occurrences of a note in a given song
7. Delete a song from the playlist, given its id
8. Delete the entire playlist
9. Exit
```

When a song is not on the playlist →

```
=========================================================
Enter your choice: 5
Enter the name of the song you want to play: luther

No song found!!

=========================================================
1. Create a new playlist
2. Add a new song to an existing playlist
3. Play all songs in the given playlist
4. Play a song from the playlist, given its id
5. Play a song from the playlist, given its name
6. Count the number of occurrences of a note in a given song
7. Delete a song from the playlist, given its id
8. Delete the entire playlist
9. Exit
=========================================================
Enter your choice:
```

Again, no need to ask the user repeatedly if the song is not present.

Menu Choice 6:

Count the number of occurrences of a note in a given song.
Song Id can be different on your system. Here in this scenario, we have the song '*Serene Adventure'* with id ***28*** and holds ***3*** occurrences of note '*do'.*

```
=========================================================
Enter your choice: 6
Enter the id of the song you want to count the notes in: 28
Enter the note you want to count: do
```

```
Number of times the note appears: 3

=====================================================
1. Create a new playlist
2. Add a new song to an existing playlist
3. Play all songs in the given playlist
4. Play a song from the playlist, given its id
5. Play a song from the playlist, given its name
6. Count the number of occurrences of a note in a given song
7. Delete a song from the playlist, given its id
8. Delete the entire playlist
9. Exit
=====================================================
Enter your choice: 6
Enter the id of the song you want to count the notes in: 110
Enter the note you want to count: sol

Number of times the note appears: 6
```

Required Validation:

Entered song note and song id should be ***validated***.
Below, in the first case, it returns -1 because of **invalid** song note (i.e. sd) and in the other case it also **returns -1** because, song id does not exist.
Message should be printed in the main.

```
=====================================================
Enter your choice: 6
Enter the id of the song you want to count the notes in: 732
Enter the note you want to count: sd

Song not found!!

=====================================================
1. Create a new playlist
2. Add a new song to an existing playlist
3. Play all songs in the given playlist
4. Play a song from the playlist, given its id
5. Play a song from the playlist, given its name
6. Count the number of occurrences of a note in a given song
7. Delete a song from the playlist, given its id
8. Delete the entire playlist
9. Exit
```

```
=====================================================
Enter your choice: 6
Enter the id of the song you want to count the notes in: 34
Enter the note you want to count: sol

Song not found!!
```

Menu Choice 7:

Your program should be able to delete the song (all related information such as ID, name and notes) using the given id.
Get help from the *playPlayList* function to manually check if the song is deleted or not.

Required Validation:

If the song id entered does not match with any existing song id then, function should return -1, 1 otherwise. Print statement should be in main.

```
=====================================================
Enter your choice: 7
Enter the id of the song you want to delete: 811

Song not found!!

=====================================================
1. Create a new playlist
2. Add a new song to an existing playlist
3. Play all songs in the given playlist
4. Play a song from the playlist, given its id
5. Play a song from the playlist, given its name
6. Count the number of occurrences of a note in a given song
7. Delete a song from the playlist, given its id
8. Delete the entire playlist
9. Exit
=====================================================
Enter your choice: 7
Enter the id of the song you want to delete: 810

Song Deleted!!
```

Menu Choice 8 :

Delete the entire playlist. Printing of ♫ characters is *optional*.
Make sure to have a ***valgrind check*** to verify the successful deletion without any *memory leak*.
Always run *choice 8* to ***free*** the memory **allocated** throughout the execution, before *choice 9 (*EXIT*)*.

```
======================================================
Enter your choice: 8

Deleting Playlist ♫♫♫

======================================================
1. Create a new playlist
2. Add a new song to an existing playlist
3. Play all songs in the given playlist
4. Play a song from the playlist, given its id
5. Play a song from the playlist, given its name
6. Count the number of occurrences of a note in a given song
7. Delete a song from the playlist, given its id
8. Delete the entire playlist
9. Exit
======================================================
Enter your choice: 9

♫ Exiting the program ♫

==203723==
==203723== HEAP SUMMARY:
==203723==     in use at exit: 0 bytes in 0 blocks
==203723==   total heap usage: 105 allocs, 105 frees, 19,544 bytes
allocated
==203723==
==203723== All heap blocks were freed -- no leaks are possible
==203723==
==203723== For lists of detected and suppressed errors, rerun
with: -s
==203723== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0
from 0)

dkumar07@linux-05:~/cis2500A3$ |
```

Valgrind check: Recall lab #4 for better understanding of valgrind.

Make sure to ***Push*** all the required ***folders*** on the gitlab. Object files and executable target files need not to be pushed. You **must** run your code on a linux server. **Don't forget** to clean your working directory using make clean before final submission. **Don't forget** to Check *FAQs* regularly.

Best wishes ,
CIS2500 TAs