Sabanci University

Faculty of Engineering and Natural Sciences
CS204 Advanced Programming
Summer 2017-2018

Take-Home Exam 2
Due: 30 July 2021 11.55pm (Sharp Deadline)

DISCLAIMER:

Your program should be a robust one such that you have to consider all relevant user mistakes and extreme cases; you are expected to take actions accordingly!

Only checking the sample run cases might not be sufficient as your solution will be checked against a variety of samples different from the provided samples; however checking these cases is highly encouraged and recommended.

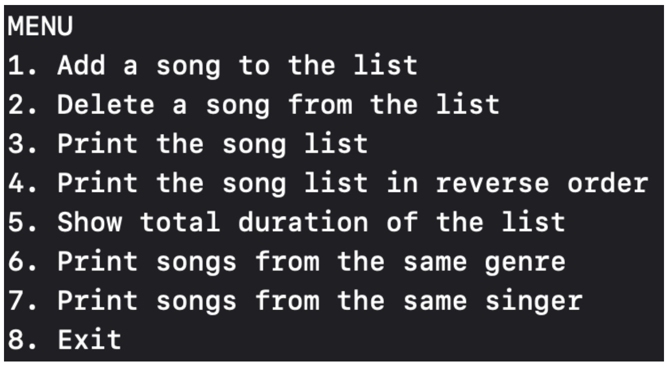
You can NOT collaborate with your friends and discuss your solutions with each other. You have to write down the code on your own. Plagiarism will not be tolerated AND cooperation is not an excuse!

Introduction

The aim of this assignment is to practice on linked lists structures. In this assignment, you will implement a dynamic list by using a **circular doubly linked list**. Several operations will be available to the user through a recurring menu.

Input to Your Program

All the inputs to your program will be obtained through an option menu which looks like:



```
MENU
1. Add a song to the list
2. Delete a song from the list
3. Print the song list
4. Print the song list in reverse order
5. Show total duration of the list
6. Print songs from the same genre
7. Print songs from the same singer
8. Exit
```

There are 8 options in the menu given, where each option has a different functionality. Options in the menu will be asked from the user recurrently until the option 8 is selected. When the option 8 is selected, ***all the dynamically allocated memory must be deallocated*** and the program should then terminate.

Format of the Inputs

In the main menu, a valid input can be any digit from "1" to "8". Trying to read this input as a string may save you a lot of headaches. Be reminded to check if the user enters anything other than the valid option numbers; in such cases your program should output **"Not a valid option."**, examples of which exist in the sample runs.

- *Option 1*: In the first option, there are 5 more inputs to be asked to the user in order to gather information about the song. 3 of these are string inputs; song name, singer name, and genre. Remaining 2 inputs are integer inputs; listener count and the duration of the song in minutes. You may assume that these last 2 inputs will always be integers for the sake of simplicity; thus, you do not have to perform any input checks for these inputs.
- *Option 2*: In the second option, if the list is not empty then the name of the song should be asked to the user as a string input. Else song name input will not be taken.
- *Options 3, 4, 5, 8*: These options do not ask for any other inputs.
- *Options 6, 7*: In the sixth and seventh options, a string input will be asked to the user, which will be either the genre of the song or the singer, depending on the selected menu option. This input will be taken unless the list is not empty.

All string inputs might consist of multiple words and there can be multiple spaces or tabs between the words. Your program should handle these cases and the final string should have single spaces between the words (and no space in the end :)).

Moreover, you should store all the inputs in uppercase. Therefore, all the letters of the string inputs should be capitalized.

Besides the string inputs, as mentioned above, for each song, there will be duration and listening count data. These two fields should be considered as integers. You are free to assume that the user enters a positive integer for those fields all the time.

Data Structure to be Used

You **must** use a **circular doubly linked list** for the implementation of the song list. Apart from the automated grading, we will also inspect your code in detail. Faking to use a linked list will not help you and you will end up with a grade of 0 directly. So, please do not try to get away with a vector/array implementation or any other data structures.

You may encapsulate the name, singer, duration, listener and genre information regarding one item in a struct, as we did for all the linked list examples in this course. Then, you can basically form a circular doubly linked list of these struct elements. Remember that you should also have some kind of pointer**s** in these structs for connecting them to each other. You are also allowed to keep extra fields in the struct if you feel like so, but these data fields (and some pointer**s**) should be kept minimal.

Tasks and Outputs

- ***Option 1***
    - Task to perform in this option is to add a song to the list. You may assume that every given song name will be unique; meaning that, one song name can not belong to more than one singer. That is why your program needs to check if the song name asked to be added is already in the list. If so, the program should output the message; **"The song *song_name* is already in the song list."**. If not, then the program should continue with asking the singer name, genre, the listener count and the duration. After adding the song and it's information to the list, your program should output the message saying; **"The song *song_name* from the singer is added to the song list."**.

- ***Option 2***
    - Task to perform in this option is to delete a song from the song list. Before performing delete operation your program should check if the list is empty and output **"The song list is empty."** message. Again, two cases might occur for delete operation: the song may exist in the list or not. If the song name could be found in the list, then it's node should be deleted from the list appropriately and a message saying **"The song *song_name* is deleted from the list."**. If the song name could not be found in the list, then your program should output **"The song *song_name* could not be found in the list."**.

For the first two tasks, there are two points to be clarified. First of all, after getting the song name, your program should consider their uppercase versions with a single space between the words and no space at the end; otherwise you might have duplicate entries in your list or your deletion operations might fail unexpectedly.

- ***Options 3 and 4***
    - Tasks to perform in options 3 and 4 are to print the stored song list in order (descending) and to print it in reverse order (ascending). The list of songs should be printed according to their listener count. Songs with higher counts of listeners should come first while printing in order (descending). If the program encounters songs with the same listener count, then it should consider the alphabetical order of the song names.
    - These tasks are basic linked lists traversals. In order for your program to print the correct results in these steps, it's more practical to keep the list sorted (descending wrt listener count and ascending wrt song name if necessary) after each action taken in first and second tasks. Please check the sample runs given below for the format which your program will show songs in the console.

- ***Option 5***
    - Task to perform in option 5 is to calculate the total duration of the song list. This task is also a basic linked lists traversal.

- ***Option 6 and 7***
    - Before performing any operation your program should check if the list is empty and output **"The song list is empty."** message.
    - Task to perform in option 6 is to print the songs with the given genre, while task to perform in option 7 is to print the songs with the given singer name.
    - Keeping the list organized is very important for these options as well, since your program should print out all the songs from the given genre/singer sorted according to their listener count (and alphabetic order if necessary).

- ***Option 8***
    - The main menu will be printed after every action, until the user selects the sixth option, which is "Exit". Once the user selects to exit, your program should print an appropriate acknowledgement message, **clear the circular doubly linked list from the memory**, and exit the program. Not returning dynamic memory back is known to cause "memory leak", which is a serious issue. If you do not clear your allocations from the memory, it may cause a leak in your grade as well.

Sample Runs

Below, we provide some sample runs of the program that you will develop. The *italic* and **bold** phrases are the standard inputs (cin) taken from the user (i.e., like ***this***). You have to display the required information in the same order and with the same words as here.

Sample Run 1

```
This program helps to create a music list and suggest you songs that you might like!
---
MENU
1. Add a song to the list
2. Delete a song from the list
3. Print the song list
4. Print the song list in reverse order
5. Show total duration of the list
6. Print songs from the same genre
7. Print songs from the same singer
8. Exit
---
Enter your choice: 0
---
Not a valid option.
---
Enter your choice: 9
---
Not a valid option.
---
Enter your choice: three
---
Not a valid option.
---
Enter your choice: 2
---
The song list is empty.
---
Enter your choice: 3
---
```

```
The song list is empty.
---
Enter your choice: 4
---
The song list is empty.
---
Enter your choice: 5
---
The song list is empty.
---
Enter your choice: 6
---
The song list is empty.
---
Enter your choice: 7
---
The song list is empty.
---
Enter your choice: 8
---
Deleting song list...
Exiting the program...
```

Sample Run 2

```
This program helps to create a music list and suggest you songs that you might like!
---
MENU
1. Add a song to the list
2. Delete a song from the list
3. Print the song list
4. Print the song list in reverse order
5. Show total duration of the list
6. Print songs from the same genre
7. Print songs from the same singer
8. Exit
---
Enter your choice: 1
---
Enter the name of the song: baka baka
Enter singer name of the song: emir taha
Enter genre of the song: pop
Enter the listener count of the song: 100
Enter the duration of the song: 3
---
The song BAKA BAKA from EMIR TAHA is added to the song list.
---
Enter your choice: 1
---
Enter the name of the song: baka baka
---
The song BAKA BAKA is already in the song list.
---
Enter your choice: 1
---
Enter the name of the song: CrY BaBy
Enter singer name of the song: Janis Joplin
Enter genre of the song: roCk
Enter the listener count of the song: 1010
Enter the duration of the song: 5
---
The song CRY BABY from JANIS JOPLIN is added to the song list.
---
Enter your choice: 3
---
Song Name: CRY BABY
Singer Name: JANIS JOPLIN
Genre: ROCK
Duration: 5
Listener Count: 1010
---
Song Name: BAKA BAKA
Singer Name: EMIR TAHA
Genre: POP
Duration: 3
Listener Count: 100
---
Enter your choice: 1
---
Enter the name of the song: TrY
Enter singer name of the song: janis joPLIN
Enter genre of the song: blueS
Enter the listener count of the song: 500
Enter the duration of the song: 4
---
The song TRY from JANIS JOPLIN is added to the song list.
---
Enter your choice: 1
---
Enter the name of the song: hole in my soul
Enter singer name of the song: aerosmith
Enter genre of the song: rock
Enter the listener count of the song: 7000
Enter the duration of the song: 3
---
The song HOLE IN MY SOUL from AEROSMITH is added to the song list.
---
Enter your choice: 3
---
Song Name: HOLE IN MY SOUL
Singer Name: AEROSMITH
Genre: ROCK
Duration: 3
Listener Count: 7000
---
Song Name: CRY BABY
Singer Name: JANIS JOPLIN
Genre: ROCK
Duration: 5
Listener Count: 1010
---
Song Name: TRY
Singer Name: JANIS JOPLIN
Genre: BLUES
Duration: 4
Listener Count: 500
---
Song Name: BAKA BAKA
Singer Name: EMIR TAHA
Genre: POP
```

```
Duration: 3
Listener Count: 100
---
Enter your choice: 4
---
Song Name: BAKA BAKA
Singer Name: EMIR TAHA
Genre: POP
Duration: 3
Listener Count: 100
---
Song Name: TRY
Singer Name: JANIS JOPLIN
Genre: BLUES
Duration: 4
Listener Count: 500
---
Song Name: CRY BABY
Singer Name: JANIS JOPLIN
Genre: ROCK
Duration: 5
Listener Count: 1010
---
Song Name: HOLE IN MY SOUL
Singer Name: AEROSMITH
Genre: ROCK
Duration: 3
Listener Count: 7000
---
Enter your choice: 5
---
Enjoy your next 15 minutes with the songs in the list.
---
Enter your choice: 6
---
Please enter the genre of the songs you want to list: pOp
---
List of All the Songs from Genre POP
---
- BAKA BAKA from EMIR TAHA
```

```
---
Enter your choice: 6
---
Please enter the genre of the songs you want to list: rock
---
List of All the Songs from Genre ROCK
---
- HOLE IN MY SOUL from AEROSMITH
- CRY BABY from JANIS JOPLIN
---
Enter your choice: 7
---
Please enter the singer name of the songs you want to list: janis joplin
---
List of All the Songs from Singer JANIS JOPLIN
---
- CRY BABY
- TRY
---
Enter your choice: 8
---
Deleting song list...
Exiting the program...
```

Sample Run 3

```
This program helps to create a music list and suggest you songs that you might like!---
MENU
1. Add a song to the list
2. Delete a song from the list
3. Print the song list
4. Print the song list in reverse order
5. Show total duration of the list
6. Print songs from the same genre
7. Print songs from the same singer
8. Exit
---
Enter your choice: 1
---
Enter the name of the song: ONE
Enter singer name of the song: metaLLiCa
Enter genre of the song: heavy metal
Enter the listener count of the song: 9000
Enter the duration of the song: 10
---
The song ONE from METALLICA is added to the song list.
---
Enter your choice: 1
---
Enter the name of the song: rock you like a            hurricane
Enter singer name of the song: scorpions
Enter genre of the song: heavy    metal
Enter the listener count of the song: 9000
Enter the duration of the song: 5
---
The song ROCK YOU LIKE A HURRICANE from SCORPIONS is added to the song list.
---
Enter your choice: 3
---
Song Name: ONE
Singer Name: METALLICA
Genre: HEAVY METAL
Duration: 10
Listener Count: 9000
---
Song Name: ROCK YOU LIKE A HURRICANE
Singer Name: SCORPIONS
Genre: HEAVY METAL
Duration: 5
Listener Count: 9000
---
Enter your choice: 1
---
Enter the name of the song: edge
Enter singer name of the song: rezz
Enter genre of the song: dark techno
Enter the listener count of the song: 9000
Enter the duration of the song: 2
---
The song EDGE from REZZ is added to the song list.
```

```
---
Enter your choice: 3
---
Song Name: EDGE
Singer Name: REZZ
Genre: DARK TECHNO
Duration: 2
Listener Count: 9000
---
Song Name: ONE
Singer Name: METALLICA
Genre: HEAVY METAL
Duration: 10
Listener Count: 9000
---
Song Name: ROCK YOU LIKE A HURRICANE
Singer Name: SCORPIONS
Genre: HEAVY METAL
Duration: 5
Listener Count: 9000
---
Enter your choice: 6
---
Please enter the genre of the songs you want to list: heavy metal
---
List of All the Songs from Genre HEAVY METAL
---
- ONE from METALLICA
- ROCK YOU LIKE A HURRICANE from SCORPIONS
---
Enter your choice: 1
---
Enter the name of the song: Enter sandman
Enter singer name of the song: metallica
Enter genre of the song: heavy metal
Enter the listener count of the song: 9000
Enter the duration of the song: 4
---
The song ENTER SANDMAN from METALLICA is added to the song list.
---
```

```
Enter your choice: 7
---
Please enter the singer name of the songs you want to list: metallica
---
List of All the Songs from Singer METALLICA
---
- ENTER SANDMAN
- ONE
---
Enter your choice: 8
---
Deleting song list...
Exiting the program...
```

Some Important Rules

Although some of the information is given below, please also read the assignment submission and grading policies from the lecture notes of the first week. In order to get a full credit, your program must be efficient, modular (with the use of functions), well commented and indented. Besides, you also have to use understandable identifier names. Presence of any redundant computation, bad indentation, meaningless identifiers or missing/irrelevant comments may decrease your grade in case that we detect them.

When we grade your assignments, we pay attention to these issues. Moreover, in order to observe the real performance of your codes, we are going to run your programs in Release mode and **we may test your programs with very large test cases**. Hence, take into consideration the efficiency of your algorithms other than correctness.

How to get help?

You may ask your questions to TAs or to the instructor. Information regarding the office hours of the TAs are available at SUCourse+. For the instructor, you may get an appointment via email.

YOU CAN USE GRADE CHECKER FOR THIS THE!

You can use GradeChecker (https://learnt.sabanciuniv.edu/GradeChecker/) to check your expected grade. Just a reminder, you will see a character ¶ which refers to a newline in your expected output.

Make sure you upload any supporting files that you uses, too.

Grade Checker and the automated grading system use a different compiler than MS Visual Studio does. Hence, you should check the “Common Errors” page to see some extra situations to consider while doing your assignment. If you do not consider these situations, you may get a lower score (even zero) even if your program works correctly with MS Visual Studio.

Common Errors Page: https://learnt.sabanciuniv.edu/GradeChecker/commonerrors.jsp

Grade Checker can be pretty busy and unresponsive during the last day of the submission. Due to this fact, leaving the assignment for the last day generally is not a good idea. You may wait for hours to test your assignment or make an untested submission, sadly..

Grade Checker and Sample Runs together give a good estimate of how correct your implementation is, however we may test your programs with different test cases and **your final grade may conflict with what you have seen on Grade Checker.** We will also **manually** check your code, indentations and so on, hence do not object to your grade based on the Grade Checker results, but rather, consider every detail on this documentation. **So please make sure that you have read this documentation carefully and covered all possible cases, even some other cases you may not have seen on Grade Checker or Sample Runs**. The cases that you *do not need* to consider are also given throughout this documentation.

Submit via SUCourse ONLY! **Grade Checker is not considered as a submission**. Paper, e-mail or any other methods are not acceptable, either.

The internal clock of SUCourse might be a couple of minutes skewed, so make sure you do not leave the submission to the last minute. In the case of failing to submit your assignment on time:

"No successful submission on SUCourse on time = A grade of 0 directly."

What and where to submit (PLEASE READ, IMPORTANT)

You should test your program using Grade Checker. We will use the same UNIX based C++ compiler that Grade Checker uses for grading your THE.

It'd be a good idea to write your name and lastname in the program (as a comment line of course). Do not use any Turkish characters anywhere in your code (not even in comment parts). If your full name is "Duygu Karaoğlan Altop", and if you want to write it as comment; then you must type it as follows:

// Duygu Karaoglan Altop

Submission guidelines are below. Since the grading process will be automatic, you are expected to strictly follow these guidelines. If you do not follow these guidelines, your grade will be zero. The lack of even one space character in the output will result in your grade being zero, so please test your programs yourself and with the Grade Checker tool explained above.

- Name your cpp file that contains your program as follows:

  "SUCourseUserName_the2.cpp"

  Your SUCourse user name is actually your SUNet username which is used for checking sabanciuniv e-mails. Do NOT use any spaces, non-ASCII and Turkish characters in the file name. For example, if your SU e-mail address is **atam@sabanciuniv.edu**, then the file name must be: **"atam_the2.cpp"**

- Please make sure that this file is the latest version of your THE program.

- You should upload all necessary files to SUCourse as well.

- Do not zip any of the documents but upload them as separate files only.

- Submit your work **through SUCourse only**! You can use the Grade Checker only to see if your program can produce the correct outputs both in the correct order and in the correct format. It will not be considered as the official submission. You must submit your work to SUCourse.

You may visit the office hours if you have any questions regarding submissions.

Plagiarism

Plagiarism is checked by automated tools and we are very capable of detecting such cases. Be careful with that...

Exchange of abstract ideas are totally okay but once you start sharing the code with each other, it is very probable to get caught by plagiarism. So, do NOT send any part of your code to your friends by any means or you might be charged as well, although you have done your assignment by yourself. Assignments are to be done personally and you have to submit your own work. **Cooperation will NOT be counted as an excuse.**

In case of plagiarism, the rules on the Syllabus apply.

Good Luck!
Elif Pınar Ön