Marmara University – Faculty of Engineering – Department of Computer Engineering

Spring 2022 – CSE1242 Computer Programming II
Homework #5

Due: 09.06.2022.Thu 23.59

1) In this homework, you will build a linked list data structure for storing and organizing a musical song database that can be iterated, viewed and played in

a) chronological order (in the order of insertion),

b) alphabetical order (sorted based on the names of the songs),

c) duration time order (sorted based on the duration times of the songs), and

d) random order (known as shuffle mode).

2) The songs with their duration times will be read from an input file (*i.e.*, "**songs.txt**") and they will be organized by using a singly linked list data structure in the program.

3) Each song struct in your linked list should contain

a) a char array of size 25, named as **songName**;

b) an integer, named as **duration** that represents duration time of the song in seconds;

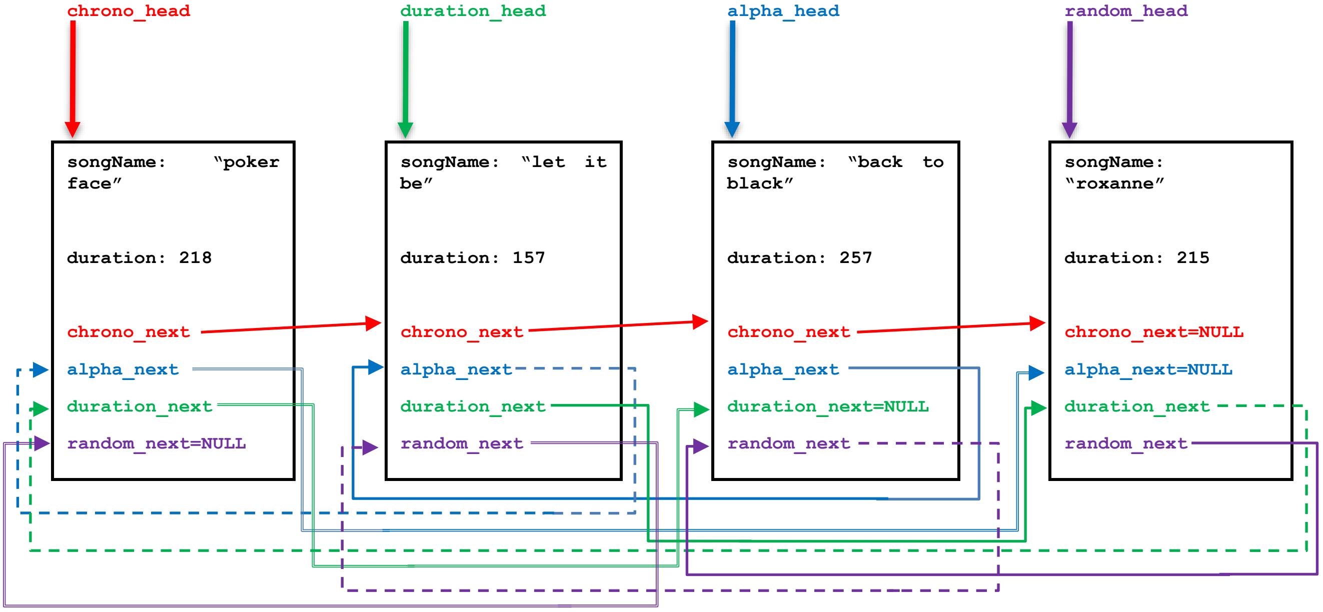
c) a pointer to a struct with the same type, named as **chrono_next**;

d) a pointer to a struct with the same type, named as **alpha_next**;

e) a pointer to a struct with the same type, named as **duration_next**; and

f) a pointer to a struct with the same type, named as **random_next**.

4) Below is a picture of the relationships between the nodes:



5) That's a whole lot of pointers!! The example above contains only 4 nodes (*i.e.*, songs) connected to each other with **`chrono_next`**, **`alpha_next`**, **`duration_next`**, and **`random_next`** pointers. These pointers tie nodes to each other based on the chronological order (the order of insertion), the alphabetical order, the duration-time order (increasingly), and the random order (shuffle mode). As it is seen from the example, there are four different types of orderings of nodes. There are four head pointers (*i.e.*, **`chrono_head`**, **`alpha_head`**, **`duration_head`**, and **`random_head`**) that show the starting node based on the target order. It should be noted that the head pointers can show either different or similar nodes (or songs) based on the target ordering.

6) Firstly, you should construct the linked list (and the connections) based on the given input file. Then, we can insert a new song to the list or delete an existing song from the list.

- **a)** When we add a new node to the list, it must be inserted at the tail of the list for the chronological order. The node should be automatically inserted to the sorted list to maintain the consistent alphabetical and duration-time orderings.
- **b)** We can delete any existing node from the beginning, middle, or end of the list.
- **c)** After insertion and deletion operations, you should carefully rearrange the links among the nodes.
- **d)** Note that the random ordering is constructed in random manner. The random list should be recreated (rerandomized) each time the list is printed.

7) Your tasks are listed as follows:

- **a)** You should declare a song struct with the features mentioned above.
- **b)** You should implement a function **`insertNode`** to insert a new node to your list. You should carefully arrange the pointers to construct different orderings.
- **c)** You should implement a function **`deleteNode`** to delete an existing node from your list. You should carefully arrange the pointers to maintain consistent orderings.
- **d)** You should implement a function **`printList`** to print the content of the linked list based on four different orderings.
- **e)** You can use the linked list examples covered in lectures to implement these functions.
- **f)** Firstly, you should read the names and duration times (based on **`minutes:seconds`** notation) of the songs from a given input file (*i.e.*, **`songs.txt`**).
- **g)** Then, you should print the content of the list to the standard output (**`stdout`**) based on four orderings.
- **h)** Then, print a menu to the user containing options such as;
  - **i)** insert a new node to the list,
  - **ii)** delete an existing node from the list,
  - **iii)** print the content of the list based on four orderings to the standard output,
  - **iv)** print the content of the list based on four orderings to a given output file, and
  - **v)** exit from the program.
- **i)** An example input file (**`songs.txt`**) and the produced output file (**`output.txt`**) are given to you based on the sample execution scenario given below. In the input file, the name of a song is separated by a single space if it contains multiple words, and the last word of the name is separated by the duration time using a **`TAB`** character.

8) SAMPLE EXECUTION SCENARIO:

```
The list in chronological order:
        1.poker face 3:58
        2.let it be 2:37
        3.hotel california 7:58
        4.englishman in newyork 4:28
        5.every breath you take 3:56
        6.shape of my heart 4:33
        7.back to black 4:17
        8.bohemian rhapsody 5:53

The list in alphabetical order:
        1.back to black 4:17
        2.bohemian rhapsody 5:53
        3.englishman in newyork 4:28
        4.every breath you take 3:56
        5.hotel california 7:58
        6.let it be 2:37
        7.poker face 3:58
        8.shape of my heart 4:33

The list in duration-time order:
        1.let it be 2:37
        2.every breath you take 3:56
        3.poker face 3:58
        4.back to black 4:17
        5.englishman in newyork 4:28
        6.shape of my heart 4:33
        7.bohemian rhapsody 5:53
        8.hotel california 7:58

The list in random order:
        1.englishman in newyork 4:28
        2.every breath you take 3:56
        3.let it be 2:37
        4.poker face 3:58
        5.back to black 4:17
        6.shape of my heart 4:33
        7.bohemian rhapsody 5:53
        8.hotel california 7:58

Enter your choice:
   1 to insert a song into the list.
   2 to delete a song from the list.
   3 to print the songs in the list.
   4 to print the songs to an output file.
   5 to end.
? 1

Enter a song name with duration:
brown eyed girl         4:45

Enter your choice:
   1 to insert a song into the list.
   2 to delete a song from the list.
   3 to print the songs in the list.
   4 to print the songs to an output file.
   5 to end.
? 3

The list in chronological order:
        1.poker face 3:58
        2.let it be 2:37
        3.hotel california 7:58
        4.englishman in newyork 4:28
        5.every breath you take 3:56
        6.shape of my heart 4:33
        7.back to black 4:17
        8.bohemian rhapsody 5:53
        9.brown eyed girl  4:45

The list in alphabetical order:
        1.back to black 4:17
        2.bohemian rhapsody 5:53
        3.brown eyed girl  4:45
        4.englishman in newyork 4:28
        5.every breath you take 3:56
        6.hotel california 7:58
        7.let it be 2:37
        8.poker face 3:58
        9.shape of my heart 4:33

The list in duration-time order:
        1.let it be 2:37
        2.every breath you take 3:56
        3.poker face 3:58
        4.back to black 4:17
        5.englishman in newyork 4:28
        6.shape of my heart 4:33
        7.brown eyed girl  4:45
        8.bohemian rhapsody 5:53
        9.hotel california 7:58

The list in random order:
        1.bohemian rhapsody 5:53
        2.poker face 3:58
        3.back to black 4:17
        4.englishman in newyork 4:28
        5.brown eyed girl  4:45
        6.hotel california 7:58
        7.every breath you take 3:56
        8.shape of my heart 4:33
        9.let it be 2:37


Enter your choice:
   1 to insert a song into the list.
   2 to delete a song from the list.
   3 to print the songs in the list.
   4 to print the songs to an output file.
   5 to end.
? 3

The list in choronological order:
        1.poker face 3:58
        2.let it be 2:37
        3.hotel california 7:58
        4.englishman in newyork 4:28
        5.every breath you take 3:56
        6.shape of my heart 4:33
        7.back to black 4:17
        8.bohemian rhapsody 5:53
        9.brown eyed girl  4:45

The list in alphabetical order:
        1.back to black 4:17
        2.bohemian rhapsody 5:53
        3.brown eyed girl  4:45
        4.englishman in newyork 4:28
        5.every breath you take 3:56
        6.hotel california 7:58
        7.let it be 2:37
        8.poker face 3:58
        9.shape of my heart 4:33

The list in duration-time order:
        1.let it be 2:37
        2.every breath you take 3:56
        3.poker face 3:58
        4.back to black 4:17
        5.englishman in newyork 4:28
        6.shape of my heart 4:33
        7.brown eyed girl  4:45
        8.bohemian rhapsody 5:53
        9.hotel california 7:58

The list in random order:
        1.hotel california 7:58
        2.back to black 4:17
        3.let it be 2:37
        4.every breath you take 3:56
        5.brown eyed girl  4:45
        6.bohemian rhapsody 5:53
        7.poker face 3:58
        8.englishman in newyork 4:28
        9.shape of my heart 4:33


Enter your choice:
   1 to insert a song into the list.
   2 to delete a song from the list.
   3 to print the songs in the list.
   4 to print the songs to an output file.
   5 to end.
? 1
Enter a song name with duration:
roxanne         3:35

Enter your choice:
   1 to insert a song into the list.
   2 to delete a song from the list.
   3 to print the songs in the list.
   4 to print the songs to an output file.
   5 to end.
? 3

The list in choronological order:
        1.poker face 3:58
        2.let it be 2:37
        3.hotel california 7:58
        4.englishman in newyork 4:28
        5.every breath you take 3:56
        6.shape of my heart 4:33
        7.back to black 4:17
        8.bohemian rhapsody 5:53
        9.brown eyed girl  4:45
        10.roxanne  3:35

The list in alphabetical order:
        1.back to black 4:17
        2.bohemian rhapsody 5:53
        3.brown eyed girl  4:45
        4.englishman in newyork 4:28
        5.every breath you take 3:56
        6.hotel california 7:58
        7.let it be 2:37
        8.poker face 3:58
        9.roxanne  3:35
        10.shape of my heart 4:33

The list in duration-time order:
        1.let it be 2:37
        2.roxanne  3:35
        3.every breath you take 3:56
        4.poker face 3:58
        5.back to black 4:17
        6.englishman in newyork 4:28
        7.shape of my heart 4:33
        8.brown eyed girl  4:45
        9.bohemian rhapsody 5:53
        10.hotel california 7:58

The list in random order:
        1.hotel california 7:58
        2.poker face 3:58
        3.every breath you take 3:56
        4.roxanne  3:35
        5.shape of my heart 4:33
        6.back to black 4:17
        7.bohemian rhapsody 5:53
        8.englishman in newyork 4:28
        9.let it be 2:37
        10.brown eyed girl  4:45


Enter your choice:
   1 to insert a song into the list.
   2 to delete a song from the list.
   3 to print the songs in the list.
   4 to print the songs to an output file.
   5 to end.
? 2
Enter a song name:
let it be
The song "let it be" is deleted from the list!

Enter your choice:
   1 to insert a song into the list.
   2 to delete a song from the list.
   3 to print the songs in the list.
   4 to print the songs to an output file.
   5 to end.
? 4
Enter a file name:
```

```
playlist.txt
Output is printed to the file playlist.txt

Enter your choice:
   1 to insert a song into the list.
   2 to delete a song from the list.
   3 to print the songs in the list.
   4 to print the songs to an output file.
   5 to end.
? 5
```

9) It should be noted that each song is printed using numbers; however, the song struct should not contain any id fields. Id values are just used for printing purposes.

10) It is highly recommended to implement chronological, alphabetical, and duration-time orderings firstly.

11) Random-ordering can be implemented lastly. In random-ordering it is required to shuffle the content of the linked list in a random manner.

12) You have to use linked lists. Use of arrays to represent the list will not be graded.

13) It should be noted that only selected parts of your homework will be graded.

SUBMISSION INSTRUCTIONS:

Please zip and submit your files using filename YourNumber_HW5.zip (*e.g.*: 150120123_hw5.zip) to http://ues.marmara.edu.tr.

Your program must include necessary comments with your own words to explain your actions!

NOTES:

1) Write a comment at the beginning of each program to explain the purpose of the program.

2) Write your name and student ID as a comment.

3) Include necessary comments to explain your actions.

4) Select meaningful names for your variables and class names.

5) You are allowed to use the materials that you have learned in lectures & labs.

6) Do not use things that you have not learned in the course.

7) **Program submissions** should be done through http://ues.marmara.edu.tr. Do not send program submissions through e-mail. E-mail attachments will not be accepted as valid submissions.

8) You are responsible for making sure you are turning in the right file, and that it is not corrupted in anyway. We will not allow resubmissions if you turn in the wrong file, even if you can prove that you have not modified the file after the deadline.

9) In case of any form of **copying and cheating** on solutions, all parts will get **ZERO** points. You should submit your own work. In case of any forms of cheating or copying, both giver and receiver are equally culpable and suffer equal penalties. **All types of plagiarism will result in zero points from the homework.**

10) No late submission will be accepted.