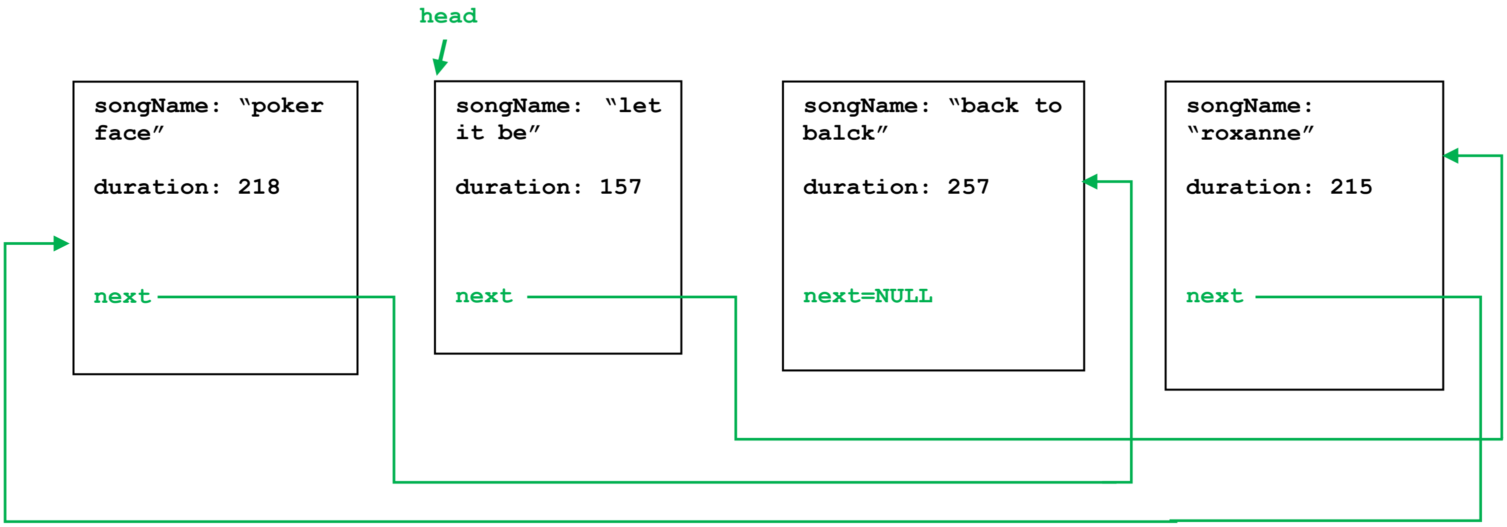
CSE 1242 - COMPUTER PROGRAMMING II
Programming Assignment #5

DUE DATE: 10/06/2023 - 23:59 (No extension)

In this assignment, you will build a linked list data structure for storing and organizing a musical song database that can be iterated/viewed/played in duration time order (sorted based on the duration times of the songs). 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.

Each song struct in your linked list should contain a char array of size **`25`** named, as **`songName`**;   an **`integer,`** named as **`duration`** that represents duration time of the song in seconds; a pointer to a struct with the same type, named as **`next`**; a pointer to a struct with the same type. Below is a picture of the relationships between the nodes:



The example above contains only 4 nodes (i.e., songs) connected to each other with **`next`** pointer. This pointer ties nodes to each other based on the duration-time order (increasingly). There is a head pointer (**`head`**) that shows the starting node.

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.

- We can delete any existing node from the beginning, middle, or end of the list.

Your tasks:

- You should declare a song struct with the mentioned features above.
- You should implement a function **`insertNode`** to insert a new node to your list.
- You should implement a function **`deleteNode`** to delete an existing node from your list.
- You should implement a function **`printList`** to print the content of the linked list.
- You can use the linked list examples covered in lectures to implement these functions.
- 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`**).
- Then, you should print the content of the list to the standard output (**`stdout`**) based on duration-time ordering.
- Then, print a menu to the user containing options such as;
  - insert a new node to the list,
  - delete an existing node from the list,
  - print the content of the list to the standard output,
  - print the content of the list to a given output file, and
  - exit from the program.

- 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 with using a **`TAB`** character.

SAMPLE EXECUTION SCENARIO:

```
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


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 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

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 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

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

--------------------------------
```

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

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

- **It should be noted that only selected parts will be graded in your homework.**

Submission Instructions

Please zip and submit your files using filename YourNumberHW5.zip
(ex: 150713852HW5.zip) to Canvas system (under Assignments tab).

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 did not learn in the course.