MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG213 Data Structures – Programming Assignment 3

Date handed out: 30 December 2021, Thursday
Date submission due: 13 January 2021, Thursday 23:00 (Cyprus Time)

Indexing Songs

This assignment aims to help you practice the AVL Tree ADT. You will write a program that can be used to analyse songs available in the given playlist. You will use this application to index the songs and find out the songs belonging to the most popular artist and genre in a given song playlist.

Requirements:
In this assignment, you are given a text file called "songs.txt" that includes the data about the songs in the playlist. This file includes the following information:

- **Song Name**: This is the name of the song.
- **Song Genre**: This is the genre of the song.
- **Album Name**: This is the name of the album that the song belongs.
- **Artist Name**: This is the name of the artist that the song belongs.
- **Release Year**: It shows the year that the song is released.
- **Date:** It shows the date and time of the song that is added to your playlist.

And sample data is as follows where each part is separated by a ";" character:

```
Unutmamali;Pop;Aacayipsin;Tarkan;1994;22/10/2019 14:11
Huzurum Kalmadi;Arabesque;Huzurum Kalmadi;Ferdi Tayfur;1976;22/12/2018 15:15
Sorma Kalbim;Pop;Dudu;Tarkan;2003;3/6/2005 15:12
Gemiler;Rock;O;Teoman;1998; 15/4/2002 08:30
```

Your task here is to process this file, and generate an AVL tree based on the **artist name** data. If there is more than one post with the same artist name, they should then be stored together in the same node. Based on this data representation, you need to write a program that provides the following functionalities to the user:

1. **Display the full index:** This will display the full AVL tree constructed. For traversal, you need to display the songs alphabetically sorted based on their artist name.

2. **The artist with the maximum number of songs:** This will display the details (Song Name, Song Genre, Album Name, Artist Name, Release Year and Date) of the songs of artist which has maximum number of songs.

3. **The oldest song:** This will display the details (Song Name, Song Genre, Album Name, Artist Name, Release Year and Date) of the oldest song (i.e. song with minimum Release Year).

Programming Requirements:
You will start by taking the file name as a command line argument and then you will need to implement at least the following functions:

- **`readData()`:** This function will mainly process the external file. As an input, it will take the file name and it will return an AVL tree.
- **`insertSong()`:** This function will take an AVL tree, and the details of a song, and then it will insert the song to the AVL tree. The song will be interested into the tree based on the artist name. You cannot make assumptions about the number of songs that belongs to the same artist name. Therefore, if the node with the given artist name exists then you will add your song details to that node.
- **`displaySongs()`:** This function will mainly take an AVL tree and display the data in the tree in alphabetical order according to the artist name.
- **`popularArtist()`:** This function will mainly take an AVL tree, will find and display the details of the songs of the artist name which has maximum number of songs. In the comment part of this function, discuss the complexity of this function based on your current representation of data. You also need to discuss if there is a way you could improve this.
- **`oldestSong()`:** This function will mainly take an AVL tree, will find and display the details of the oldest song (song with minimum Release Year). In the comment part of this function, discuss the complexity of this function based on your current representation of data. You also need to discuss if there is a way you could improve this.

Please note that in this assignment, you can make use of the functions in the **string.h** library and similar external libraries. You cannot assume about the number of songs in this external file.

Submission Requirements:
In this assignment, you need to have a header file (avltree.h) which includes the major functionality of the AVL Tree ADT. If you will use other ADTs, you need to create a separate header file for each of them. You also need to have a C source file (songIndexing.c) that includes the main function and other functions. You need put all these files into the "cng213a3" folder and then submit the compressed version of the folder to ODTU-CLASS. If you do not follow this structure, you will lose %10 from the overall grade.

Programming Style Tips!
Please follow the modular programming approach. In C programming, we use functions referred to modules to perform specific tasks that are determined/guided by the solution. Remember the following tips!
• Modules can be written and tested separately!
• Modules can be reused!
• Large projects can be developed in parallel by using modules!
• Modules can reduce the length of the program!
• Modules can also make your code more readable!

Important Notes:
- Remember to have good programming style (Appropriate comments, variable names, formulation of selection statements and loops, reusability, extensibility etc.). Each of the items above will include10% for good programming style.
- Read rules regarding to assignments from the Syllabus carefully.

- If your code does not compile due to syntax errors, you will automatically get zero.
- If your code includes a variable declaration inside a for loop such as for(int i=0; i<5;i++), you will automatically get zero.
- If your code includes global varibles, you will automatically get zero.

Grading:
Your program will be graded as follows:

Grading Point Mark	**(out of 100)**
AVL Tree Data Structure	5
Processing data file (**`readData()`**)	15
Inserting/Updating a node in the tree (**`insertSong()`**)	25
Displaying the songs(**`displaySongs()`**)	10
Displaying the details of the songs of the artist name which has maximum number of songs (**`(popularArtist)`**) (complexity discussion – 2pts)	15
Displaying the details of the song with minimum Release Year (**`oldestSong()`**) (complexity discussion – 2pts)	15
The main function	15

Sample run:
You can find the sample run of the program for the first 4 entries in the above text file.

```
>>songIndexing songs.txt

Welcome to Song Indexing
--------------------------------------------------------------------------------
1. Display the full index of songs
2. Display the songs of popular artist
3. Display the oldest song
4. Exit

Option: 1

Huzurum Kalmadi;Arabesque;Huzurum Kalmadi;Ferdi Tayfur;1976;22/12/2018 15:15

Unutmamali;Pop;Aacayipsin;Tarkan;1994;22/10/2019 14:11

Sorma Kalbim;Pop;Dudu;Tarkan;2003;3/6/2005 15:12

Gemiler;Rock;O;Teoman;1998; 15/4/2002 08:30

--------------------------------------------------------------------------------
1. Display the full index of songs
2. Display the songs of popular artist
3. Display the oldest song
4. Exit

Option: 2

Unutmamali;Pop;Aacayipsin;Tarkan;1994;22/10/2019 14:11

Sorma Kalbim;Pop;Dudu;Tarkan;2003;3/6/2005 15:12
```

```
--------------------------------------------------------------------------------
1. Display the full index of songs
2. Display the songs of popular artist
3. Display the oldest song
4. Exit

Option: 3

Huzurum Kalmadi;Arabesque;Huzurum Kalmadi;Ferdi Tayfur;1976;22/12/2018 15:15
--------------------------------------------------------------------------------
1. Display the full index of songs
2. Display the songs of popular artist
3. Display the oldest song
4. Exit

Option: 4
```