

Project “Songs”

Themes: “File Handling”, “Singly Linked List”

The Playlist is represented as a Singly Linked List where every Song is an Element of this List.

The next library used:

```
#include <stdio.h>
#include <stdlib.h> -- for malloc
#include <string.h> -- for string operations
#include <stdbool.h> -- to use boolean variables
#include "header.h" -- to use the header file
```

“header.h”

To simplify the program the file “header.h” was used. It contains functions and structures which are later used in “main” functions.

Struct length

Special structure to represent the length of every song.

Consists of minutes and seconds that cannot be more than 59.

Struct song

Special structure to store the data about a song. “title[50]” - the title of a song, contains 50 empty elements; “singer[50]” - singer of a song, also string with 50 elements; “album[50]” - from which album a song; “year” - year when a song was released (only integer type); “genre[50]” - the genre of a song; “length length” - length of every song.

Struct SongInList

Represents an Element of Singly Linked List or in other words a Song in the Playlist.

Contains Structure Song type variable - “songData”, which in the future will give access to all information regarding the song (title of the song, name of the performer (singer or group), the title of the album it was released on, year of release, the genre of the music (rock, pop, hip-hop, jazz, classical, etc.), length of song (in minutes and seconds)).

Because it is a Linked List it also should contain the point to the next Element, which is “struct SongInList *next” in this program.

printMenu()

Void function which prints the menu. The purpose of usage is to simplify the program.

void displayByCommonArtist(SongInList* n)

Display all songs of a user-selected artist. The parameter is the first Element of the list.

If there are no songs in a playlist, the user is notified and a new cycle begins. If the List is not empty then, there is a traversing loop. So the condition of the loop is true until the element is

not “NULL”. This function compares every song’s artist with the entered one using strcmp() function from <string.h> library. If they are the same the function prints the titles of those songs.

If there is no such artist in the playlist, the user will be notified about it. For that there is a boolean variable “conditionForError” which will be true once there is such artist in the playlist.

void displayByCommonAlbum(SongInList* n)

Display all songs of a user-selected Album. The parameter is the first Element of the list.

If there are no songs in a playlist, the user is notified and a new cycle begins. If the List is not empty then, there is a traversing loop. So the condition of the loop is true until the element is not “NULL”. This function compares every song’s artist and album with the entered data using strcmp() function from <string.h> library. If they are the same the function prints all information about those songs.

If there is no such artist or album in the playlist, the user will be notified about it. For that there is a boolean variable “conditionForError” which will be true once there is such an artist or album in the playlist.

void displayByCommonYear(SongInList* n)

Display all songs of a user-selected year of release. The parameter is the first Element of the list.

If there are no songs in a playlist, the user is notified and a new cycle begins. If the List is not empty then, there is a traversing loop. So the condition of the loop is true until the element is not “NULL”. This function compares every song’s year of release with the entered one. If they are the same the function prints all information about those songs.

If there is no such year of release in the playlist, the user will be notified about it. For that there is a boolean variable “conditionForError” which will be true once there is such an artist or album in the playlist.

void displayByCommonGenre(SongInList* n)

Display all songs of a user-selected Genre. The parameter is the first Element of the list.

If there are no songs in a playlist, the user is notified and a new cycle begins. If the List is not empty then, there is a traversing loop. So the condition of the loop is true until the element is not “NULL”. This function compares every song’s genre with the entered data using strcmp() function from <string.h> library. If they are the same the function prints all information about those songs.

If there are no such songs with such genre in the playlist, the user will be notified about it. For that there is a boolean variable “conditionForError” which will be true once there is such an artist or album in the playlist.

void printToFile(SongInList* n, FILE* fp, char fileName[25], char mode[3])

This function prints and saves information about songs in FILE (Playlist). There is also the parameter “mode” which in the main function can be “w” or “a” which means the file can be opened to rewrite the information in FILE or append information. The parameter “n” is again the first Element of Linked List. The loop allows going through all the elements/songs. fprintf() writes information about the song into FILE. After that, we close that file. The user is also notified about successfully saved data into the file.

SongInList* insertNewSong (SongInList* n)

Insert a new Element to the Linked List or insert a new song at the end of the playlist. In the loop the point n stops at the last element and after that, there is a new element created. The user-entered information about the new song is stored at this element using strcpy() function from <string.h> library. After that, all that information of the last element/new Song will be written in FILE using the already created function “printToFile()”. Then a new Song will be inserted to the original list. This function then returns a new List with that new Song. If a user enters the invalid length, namely minutes or seconds more than 59, the user will be notified.

void freeList(SongInList* n)

This function is to free memory allocated space. It will delete every element of the Linked List.

FinalState.c

In the main function, there are three first elements in the list with the head called “head”. 3 default songs are saved immediately in the FILE “Playlist.txt”.

The user will be greeted by the program and aware of the instructions for using the program. Then there is a menu that appears every time a user selects some option. According to Specification the program works using all foregoing functions. The user will be notified if there is no such an option. If the user selects the “Stop” option, the function **freeList()** will be used to free memory and the program will stop.

Songs to test:

Preacher	OneRepublic	Native	2013	pop-music	5:25
Secrets	OneRepublic	WakingUp	2009	rock	3:45
OnlyGirl	Rihanna	Loud	2010	pop	3:56
Woo	Rihanna	Anti	2016	dance-pop	3:55
Lost	BTS	Wings	2016	pop	4:41