**CHAPTER 1**

**INTRODUCTION**

**1.1 INTRODUCTION**

MUSIC LIBRARY is a program which organizes music and also allows a user to create their own playlist. Since this program is music based, it organizes any music stored by the user into its default classes. Many of the program based on music organization is developed day by day. With the increase in technology, many of the programs are created to suit many types of users. Some prefer music application which just organizes music and some prefer music applications which provide song lyrics. So, many of the music applications are developed to suit different categories of people.

So, this program was created with the aim of organizing music according to the preference of the user. While many of the music applications help in music organization but at the same time, they cost a lot. Many of such application cannot be accessed unless it is purchased and later many disadvantages are seen. What’s the use if it’s not user suitable after purchasing it, so this program is created with the aim of organizing music using object-oriented programming concept in C++.

C++ is an enhanced version of C which provides a greater number of special features and efficiency. While C is the grandfather of most of the programming languages, most of the high-level languages were developed using C. C++ is an object-oriented programming language with many of special features like inheritance, data hiding, polymorphism, etc.

C++ has many of its advantages to C. In C++, variables can be declared anywhere on the scope (at the time of its first usage). It supports both built-in and user-defined data-types. Data is hidden in C++ and is accessible to only its member functions and hence provides more security to the data. C++ provides reference passing, which creates an alias for the parameters passed and it very much suitable than pointers. It provides function overloading and operator overloading. So, C++ is better than C in many ways, giving user wider features making it easier to do difficult operations.

**OBJECTIVES:**

* To allow the user to create their own playlist according to their own preference,
* To organize music according to different classes,
* To provide security to the data provided by the user,
* To utilize C++ in an efficient way to run the program,

**1.2 PROBLEM DEFINITION**

MUSIC LIBRARY is a program based on the organization of music. Using this program, the user can add music into their track and onto their playlists. Usually, many of the users like to create their own playlists and these playlists cannot be inherited, so they have to make another playlist from the beginning. So, this program enables a feature called CLONE to inherit the data of old playlist into a new one. This program is specifically based on creating playlist and organizing music and some additional feature are also added to make the program more preferable for the users.

The outcomes of this program are achieved using the special features of C++. The feature like data hiding/encapsulation helps to keep data secure from unintended access of data. Different playlists are created using classes and objects, and using copy constructor cloning of old playlist is achieved. Data binding helps to make option choosing easier and more efficient.

In this program, the user is allowed to enter into the track lists and can go to main menu. In the main menu, the user can go to the list of playlists, create a new playlist or add music to the track. While creating a new playlist, the user is asked to enter the name of the playlist and the user can only add music to the playlist. In the list of playlists, the user can either open, delete or clone a playlist. Inside a playlist, the user can add or remove music by entering the music name and artist name. While deleting music, the user is asked to enter the music name and artist name. If the following inputs do not match, an error message is displayed. And on clone option, user can inherit the contents of the following playlist to a new playlist.

**1.3 REQUIREMENTS**

**HARDWARE REQUIREMENTS:**

* 1GB RAM (2GB recommended)
* 1024x768 resolution monitor or higher
* PENTIUM 3 (32 bit) or higher

**SOFTWARE REQUIREMENTS:**

* WINDOWS 7 or higher version (32-bit or 64-bit version)
* Turbo C++ or other C++ compiling software
* 5-8GB free space on the hard disk

**CHAPTER 2**

**OBJECT ORIENTED FEATURES**

**2.1 OBJECTS**

Objects are the basic unit of object-oriented programming which is the collection of data members and member functions. Objects are the units having the characteristics defined by a class. These objects are run-time entities. In object-oriented programming, the data are contained inside the objects.

Syntax: classname objectname;

Eg: plant a;

Here, plant is a classname and a is an object.

**2.2 CLASS**

Classes are the blueprints for creating any object. Class is a user-defined data-type which contains its own data members and member functions. These classes do not create any memory when an object is created a memory is allocated for the objects. For example there are many houses with different types, but all of them consists of some common features like windows, doors, pillars, etc. So, here house is the class and windows, doors, pillars are its characteristics.

Syntax: class classname

{

//define;

……

};

Eg: class plant

{

int a;

plant(int b):a(b){} };

**2.3 POLYMORPHISM**

Polymorphism is one of the special features of object-oriented programming which allows an object to perform differently according to the condition. Polymorphism is also considered as the ability to perform different operations using the same function or operator. There are two types of polymorphism:

* Compile time polymorphism: In compile-time polymorphism, two or more functions or operators can be created to perform different operations by passing different data-types or providing different return-types.
* Runtime polymorphism: In run time polymorphism, suppose a base class SHAPE consists of a member function having same name as in the sub-class CIRCLE of class SHAPE. The function that is to be called, is decided at the run-time. This can be achieved using function overriding.

**2.4 DATA ABSTRACTION**

Data abstraction refers to providing essential information hiding the background operation. Here in C++, a class can contain data members and member functions. The members which are made public are visible to the user and the private members are hidden. As implemented in excel, the calculation done is hidden, here the data and operations are also hidden by using private tag. So, a class can decide which member to show outside and which not to. Data abstraction can also be implemented using header files. Data abstraction helps to avoid low-level coding and also increases the security of the data.

**2.5 ENCAPSULATION**

Encapsulation refers to wrapping up data and information in a single unit as in a capsule. In C++, encapsulation is achieved by creating a class. The data members and member functions are enclosed inside a class. By giving private access to the data members, these data can be accessed only inside the class. To access these data, one needs to take permission and that is done by implementing friend function or friend class. So, encapsulation also helps in data abstraction and data security.

DATA

MEMBERS

MEMBER

FUNCTIONS

CLASS

Fig1:encapsulation

**2.6 INHERITANCE**

Inheritance is a special feature of object-oriented programming, by which one class acquires the properties of another class. With the help of inheritance, we can reuse the data and member functions of an existing class. For example there is a class called VEHICLE with data members as wheels, window, door, etc. Another class FERRARI can be created using the class VEHICLE as both of them have same properties.



wheels



windows

wheels

wheels



doors

windows

windows



doors

doors

VEHICLE

(BASE CLASS)

FERRARI

(INERITED CLASS)

90

LAMBORGHINI

(INHERITED CLASS)

Fig2:inheritance

**CHAPTER 3**

**PROJECT DESIGN**

**3.1 ALGORITHM**

THE MUSIC LIBRARY is a program which aims to allow the users to add music which organizes the music into its different genres. In this program, the user is provided with 3 of the options, i.e. to open a list of playlists, to create a new playlist and to add music to the track. By using the option to open list of playlists, the program displays the different playlists available. Here, the user can open, delete or clone the selected playlist. By using the option to create new playlist, a new playlist can be created where music has to be added manually. By using the option to add music to track, the user can add music and the music will be automatically organized into its class.

**Step 1:** Start

**Step 2:** Display the menu

1. LIST OF PLAYLISTS

2. CREATE NEW PLAYLIST

3. ADD NEW SONG TO THE TRACK

**Step 3:**

**Case 1: LIST OF PLAYLISTS:**

In this option, the user is allowed to view the list of available playlists and select a playlist. Upon selecting a playlist, the user is again provided options to open the playlist, delete the playlist and clone the playlist.

i) Open: In opening the playlist, the contents of the selected playlist will be displayed and the user will be allowed to add and remove music to the playlist.

ii) Delete: In delete playlist, the user can delete only the created playlist. Here, the user cannot delete the default playlist.

iii) Clone: Then by using clone option, the user can create a new playlist having the same contents/data of the following playlists. This is called inheritance property of an object-oriented program.

**Case 2: CREATE NEW PLAYLIST:**

By using this option, the user is allowed to create a new playlist. After the playlist is created, the user can add music to the playlist. Here, the user cannot remove music or open music. So, the user is allowed to only add music by using this option.

**Case 3: ADD MUSIC TO THE TRACK:**

By using this option, the user can add music on to the track. And the music is automatically added to its default playlist. Here, the user is asked to enter the music name and its artist’s name, and the program checks the artist’s name on the whole playlists collection list. If it is found, it is added to that following playlist else it is added to the OTHER playlist.

**Case 4: QUIT:**

It allows the user to terminate the program. Without this option, user cannot terminate the program.

**Step 4:** Stop

**3.2 FLOWCHART**

**DISPLAY MENU**

**OPTION**

**RETURN**

**CLONE**

**DELETE**

**OPEN**

**CHOOSE OPTION**

**LIST OF PLAYLISTS**

**DISPLAY PLAYLISTS**

**SELECT A PLAYLIST**

**CREATE PLAYLIST**

**ADD SONGS TO TRACK**

**QUIT**

**LIST OF PLAYLIST**

**fig (1):main()**

**fig(2): list()**

**fig (3): new()**

**RETURN**

**READ SONG NAME & ARTIST**

**ADD SONGS**

**RETURN**

**READ NAME**

**ADD SONGS**

**CREATE NEW PLAYLIST**

**fig (4): add()**

0-:

**3.3 CLASS DAIGRAM**

Read()

**Music class**

Song

Artist

Setsong()

Delsong()

Display()

Create()

**Playlist class**

Playname

Pos

Open()

Delete()

Organize()

Clone()

**Fig1:class diagram**

**3.4 DATAFLOW DIAGRAM**

Main menu

CREATE A NEW PLAYLIST

ADD MUSIC TO THE TRACK

LIS T OF PLAYLIST

NEW PLAYLIST

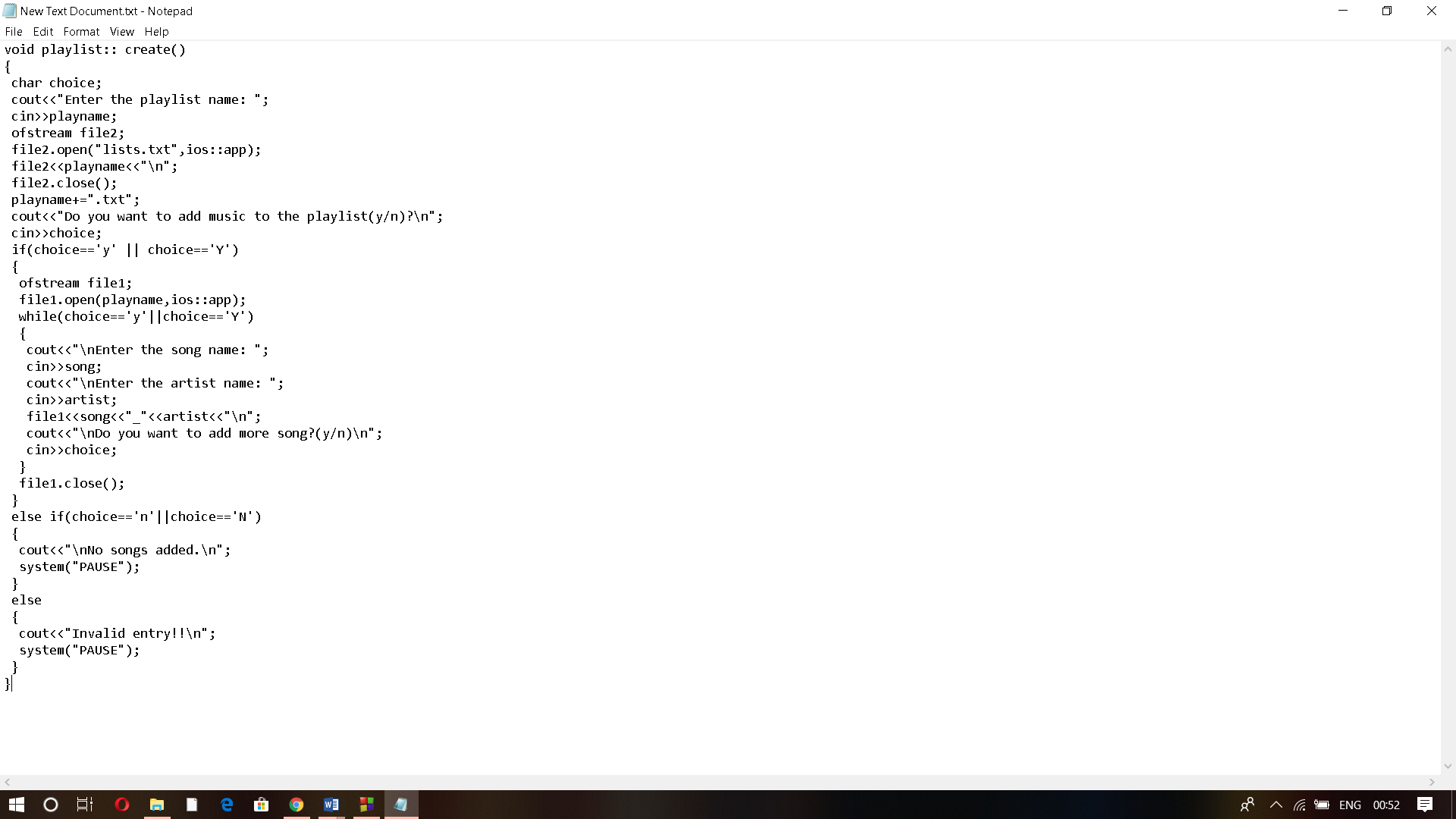
SONGS ADDED TO DEFAULT PLAYLIST

Fig1:data flow diagram

**CHAPTER 4**

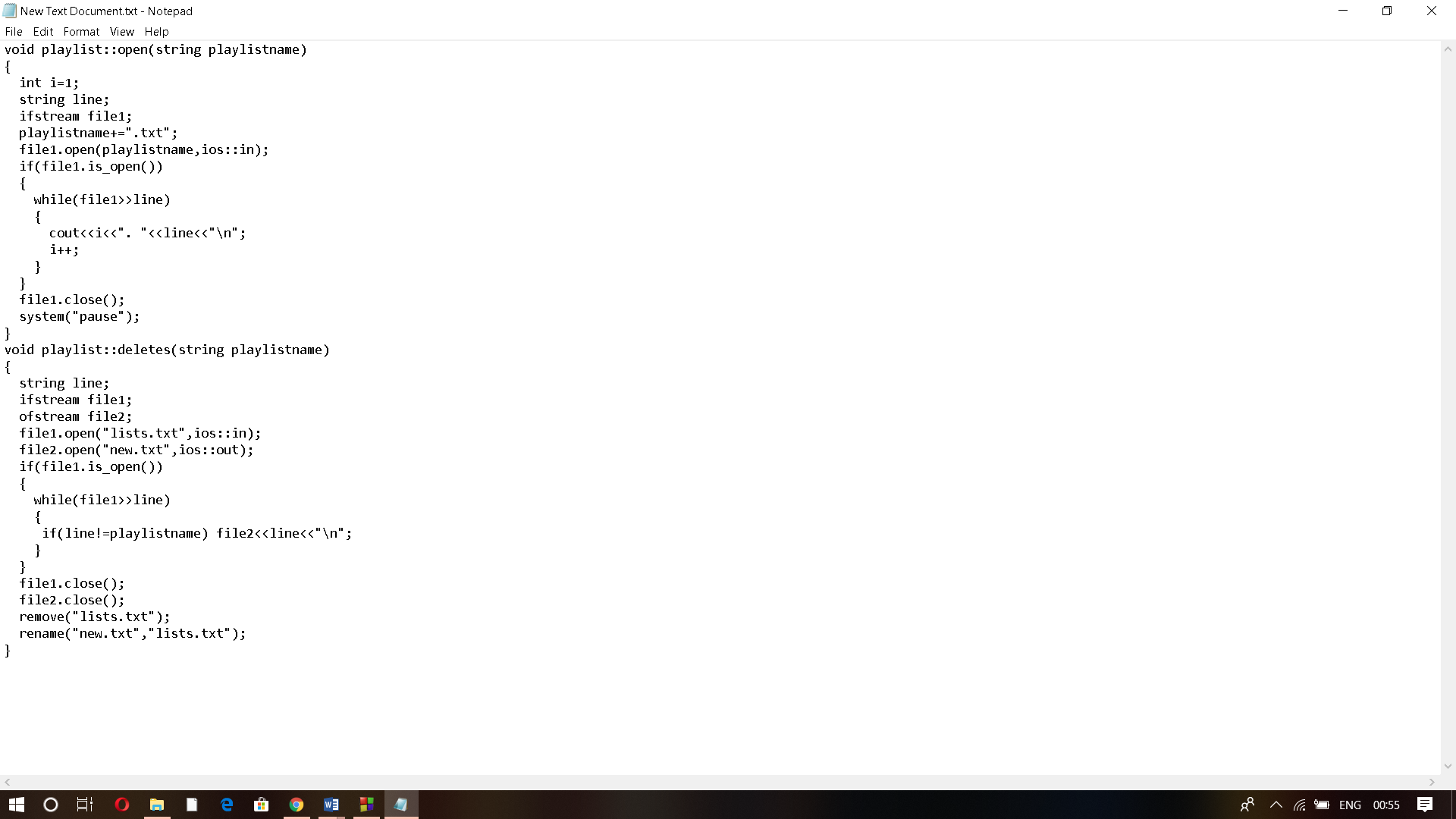
**IMPLEMENTATION**

Creating a new playlist in the Music Library:



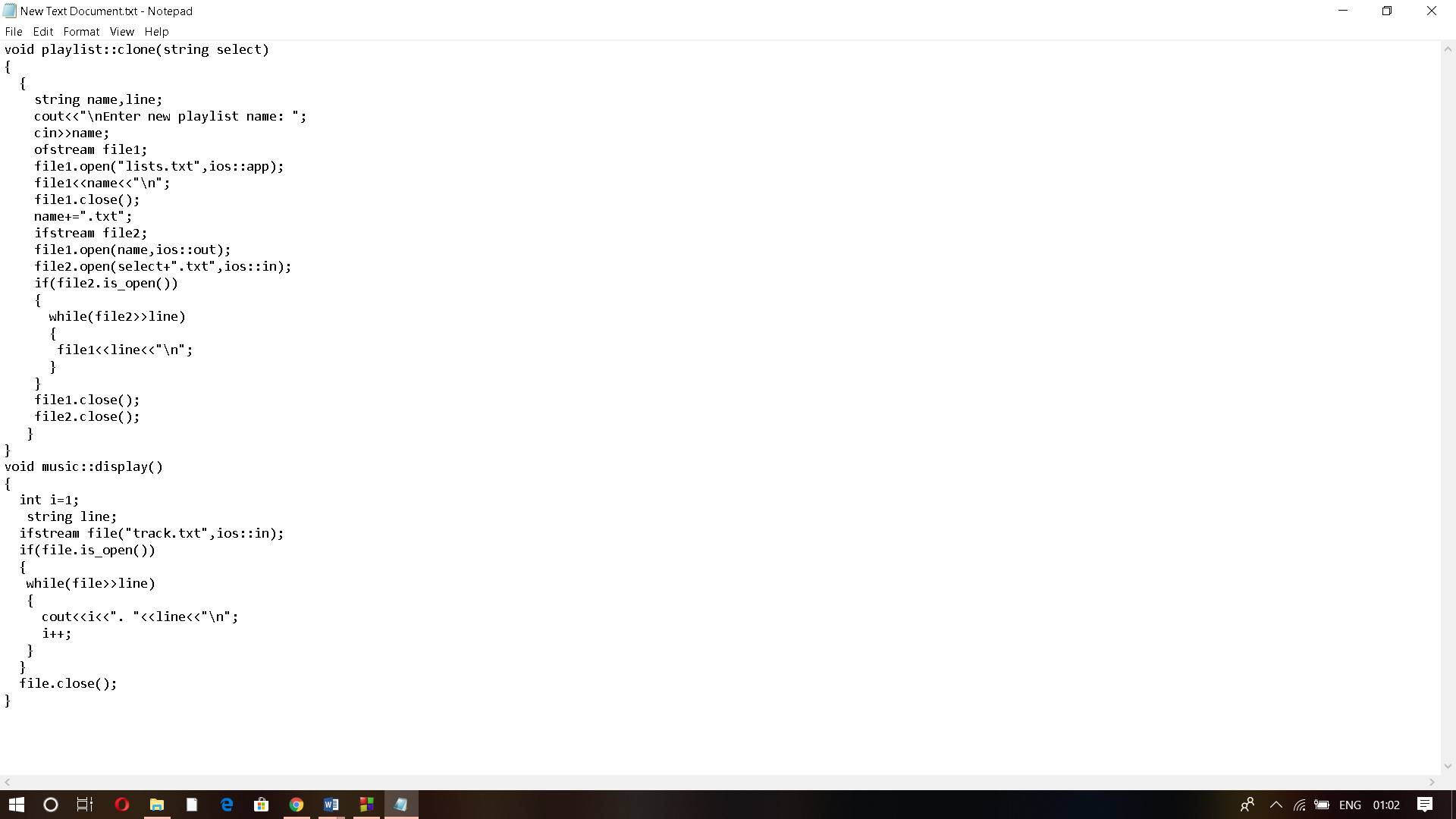
This part of the program will add a song and create a playlist in the Music Library.

To check to open a playlist and delete a particular playlist:



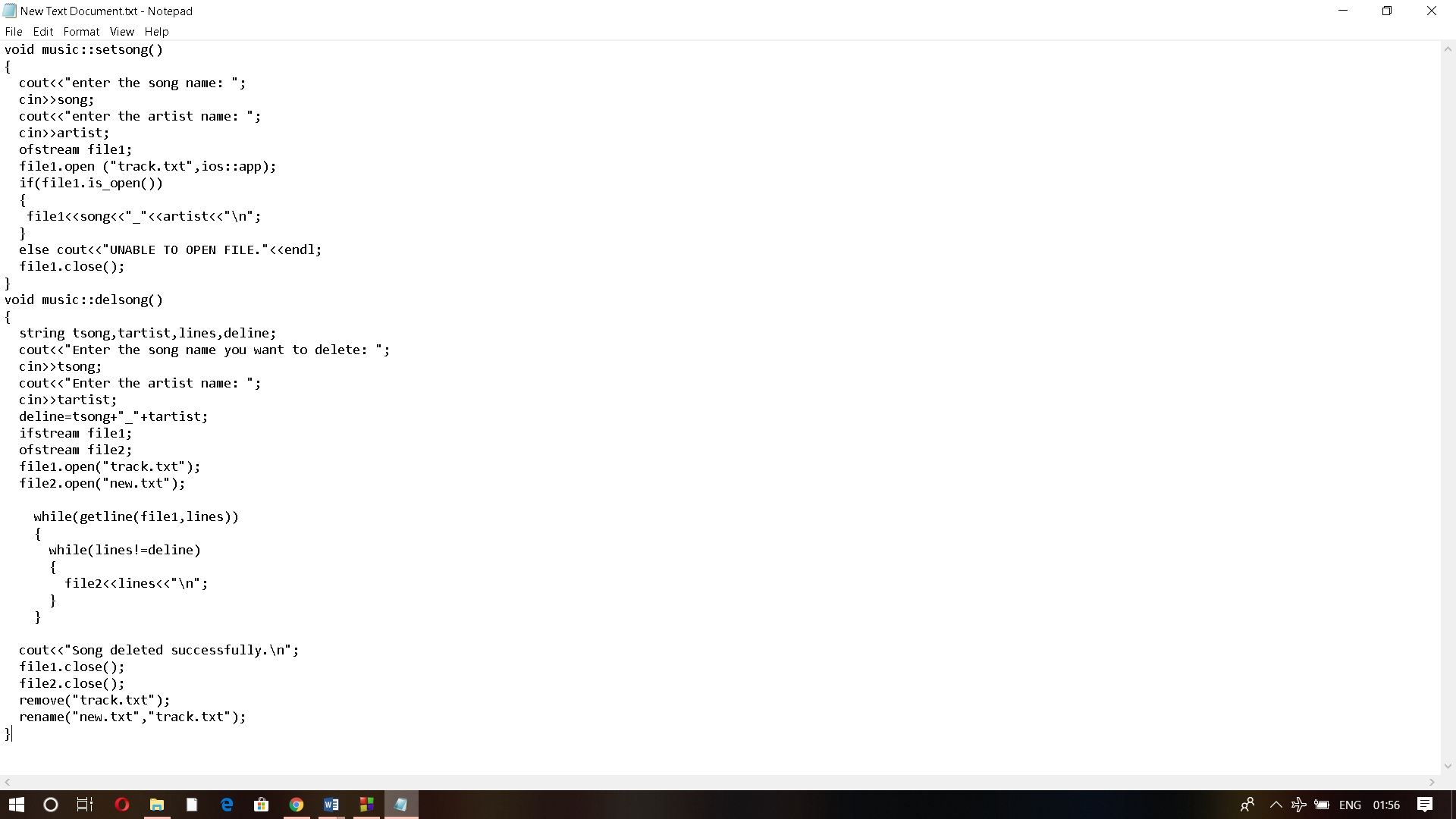
This part of the program will check to open a playlist and delete a particular playlist.

Checking to copy the contents of a playlist and to read and display:



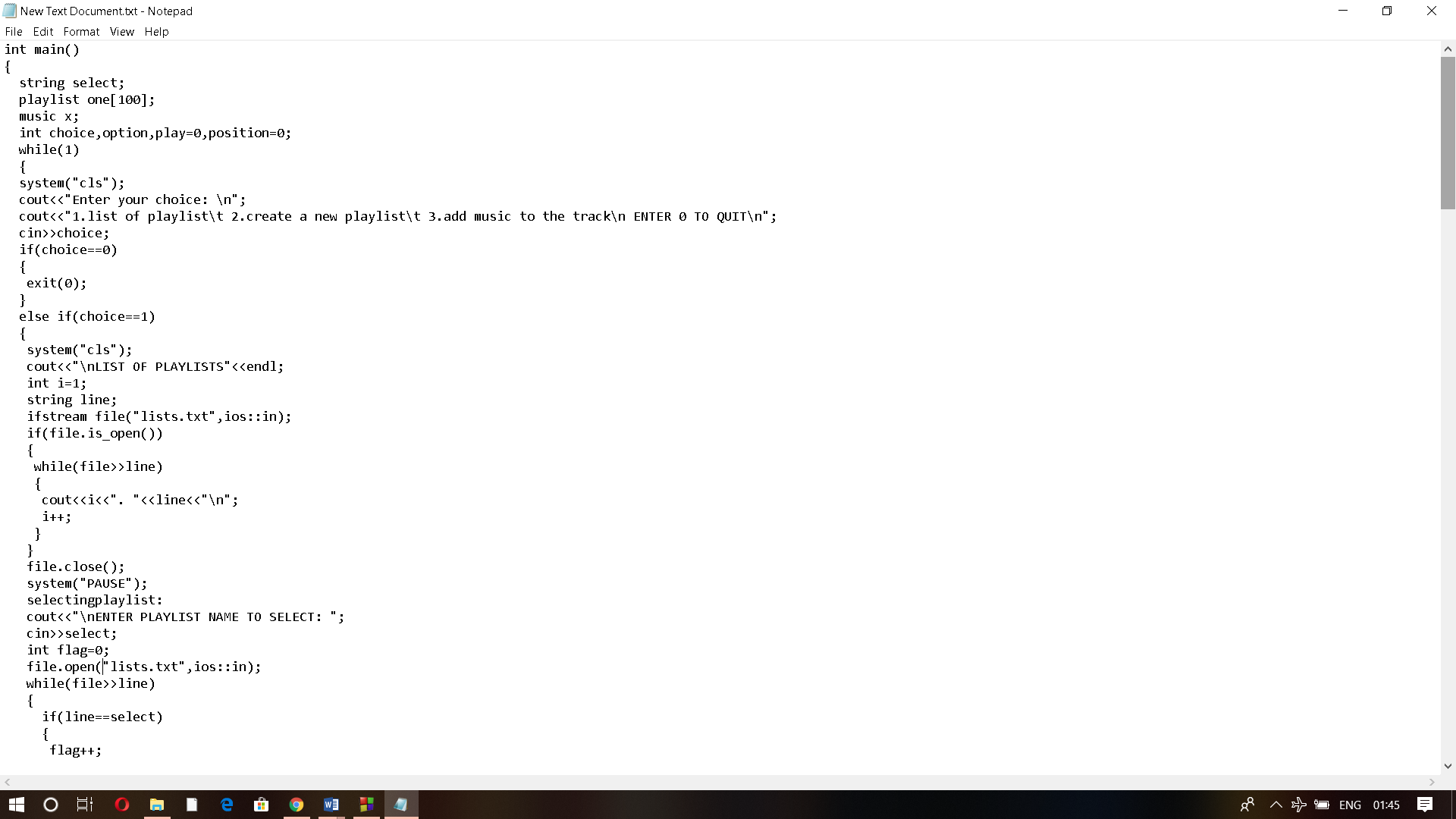
This part of the program will check the contents of a particular playlist and read and display it.

Setting the Song and Deleting the Song in Music liberary:

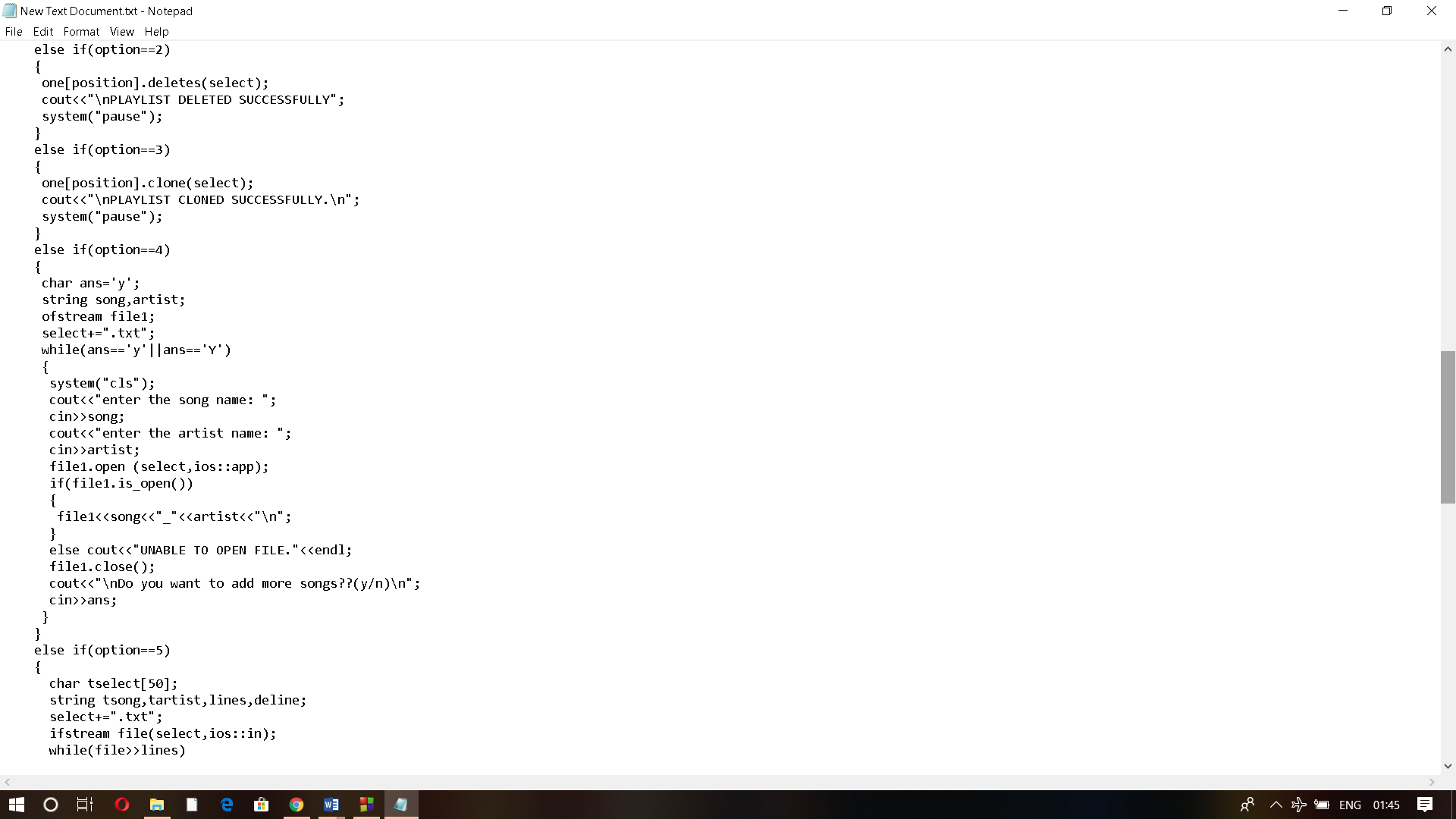


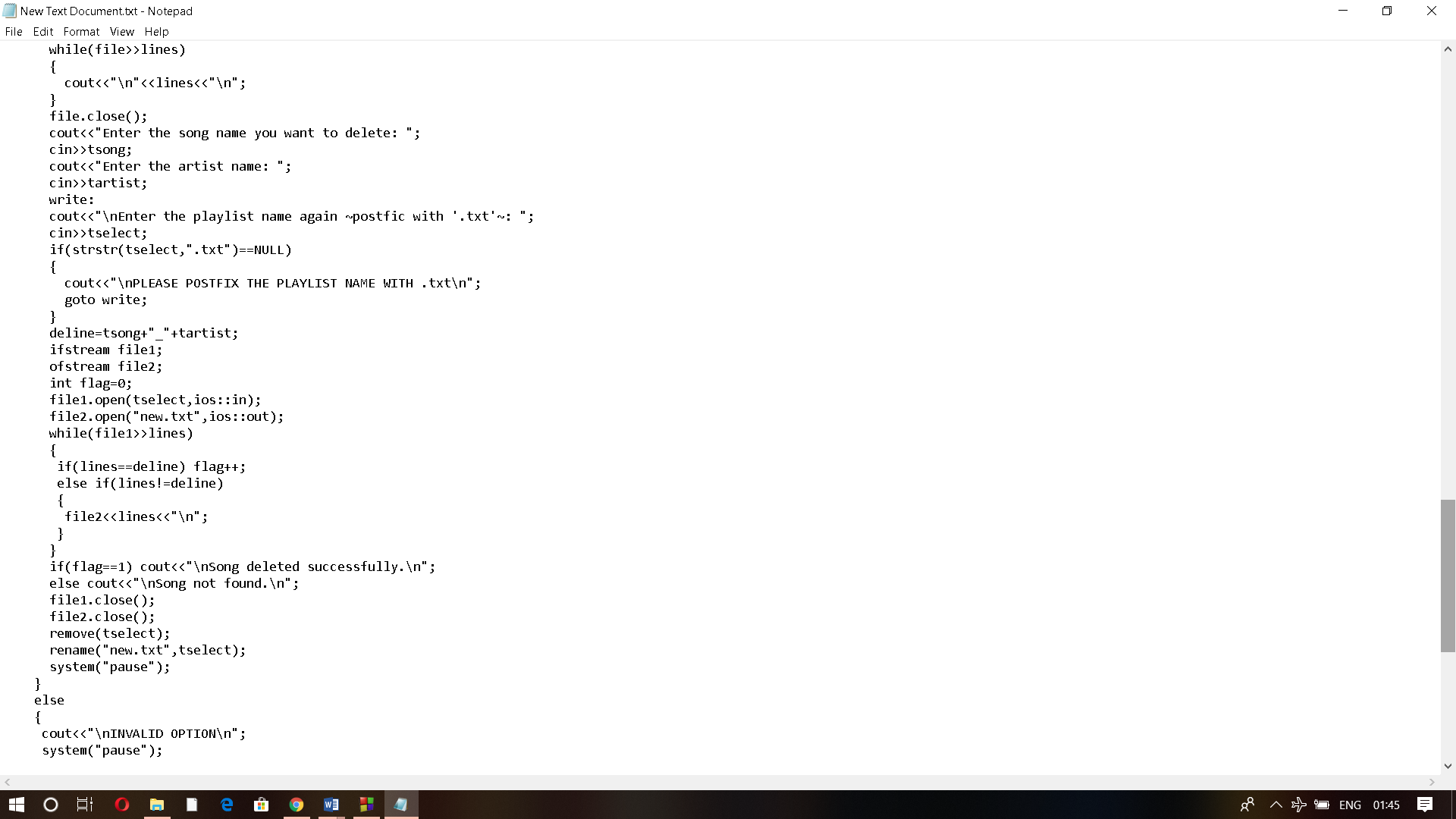
This part of the program will set the song and delete the song in Music Library.

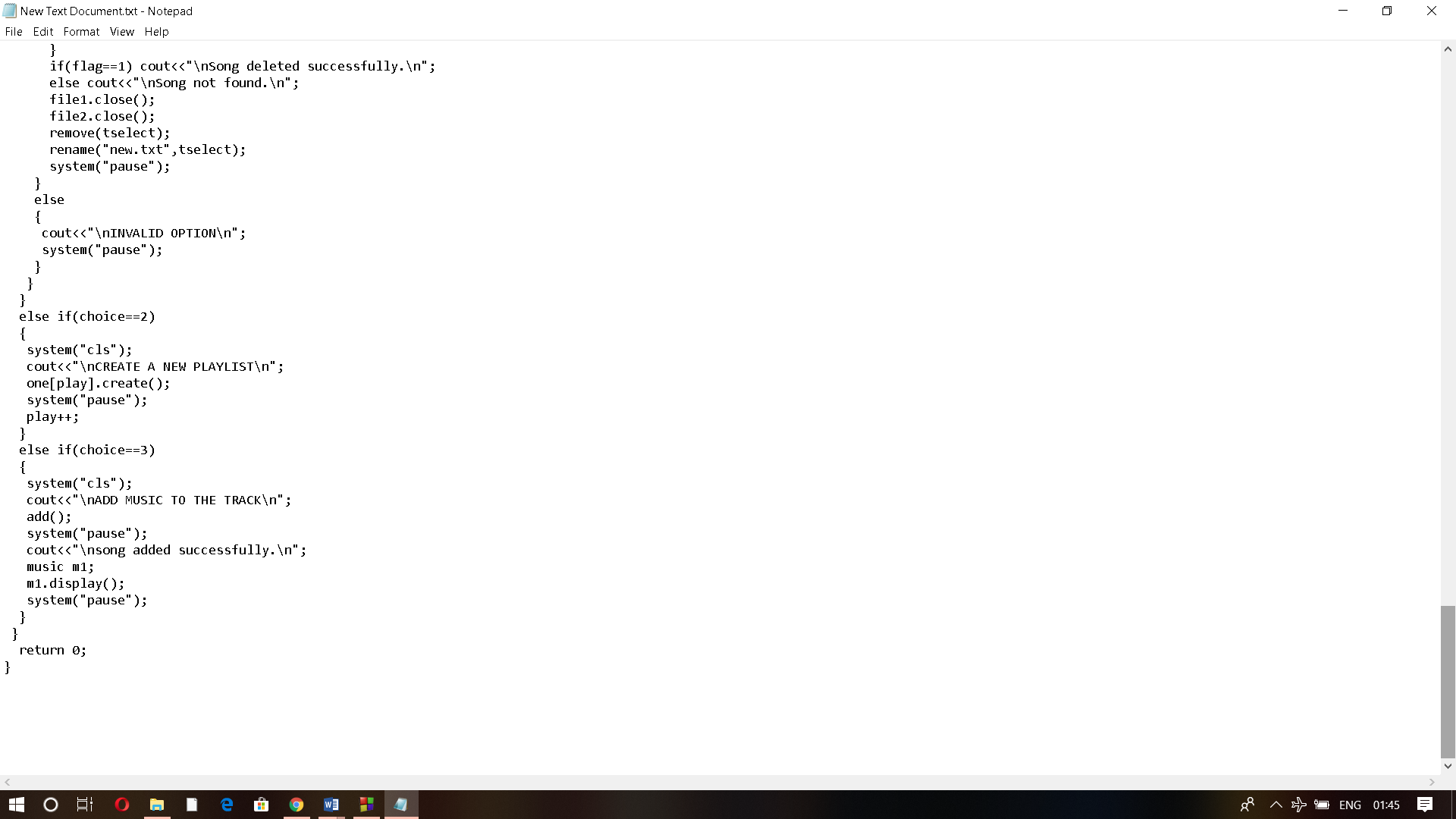
Main function:









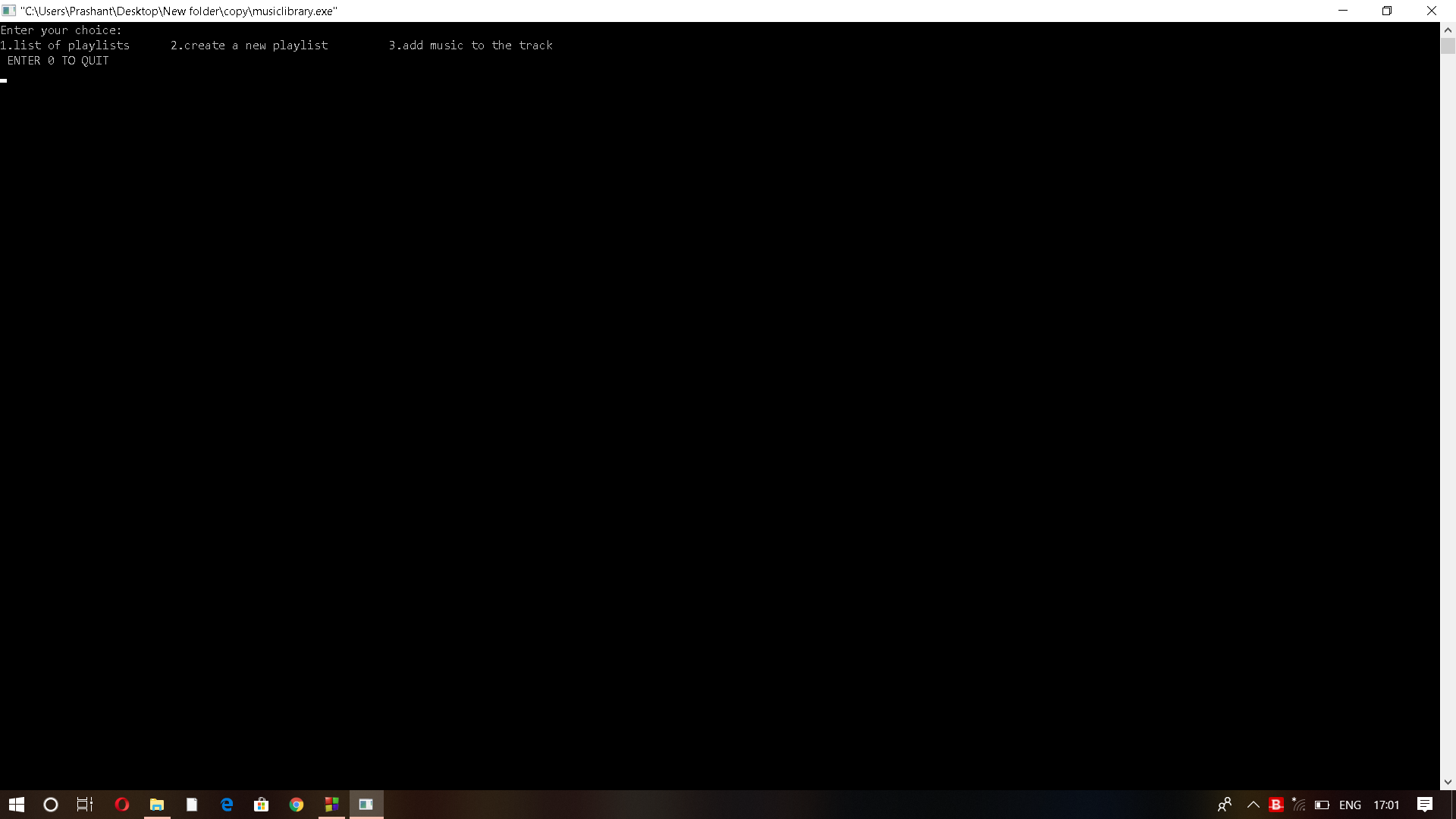


This is the main function of the program. Here, the main menu of the music library is given and the options are available for different operations. If a user chooses to open the list of playlists, the program displays all the playlists names. The user can select any playlist and he/she is provided with the choice to open, delete, clone, or add music to or delete music from the playlist. While default playlists like RECENTLY ADDED, ROCK, POP, etc. are provided with the only option to open and clone the playlist. In the option, create a new playlist, a member function to create a new playlist is called. While for add music to the track, user can add music to the track and the music of certain genre will be automatically added to the default playlists.

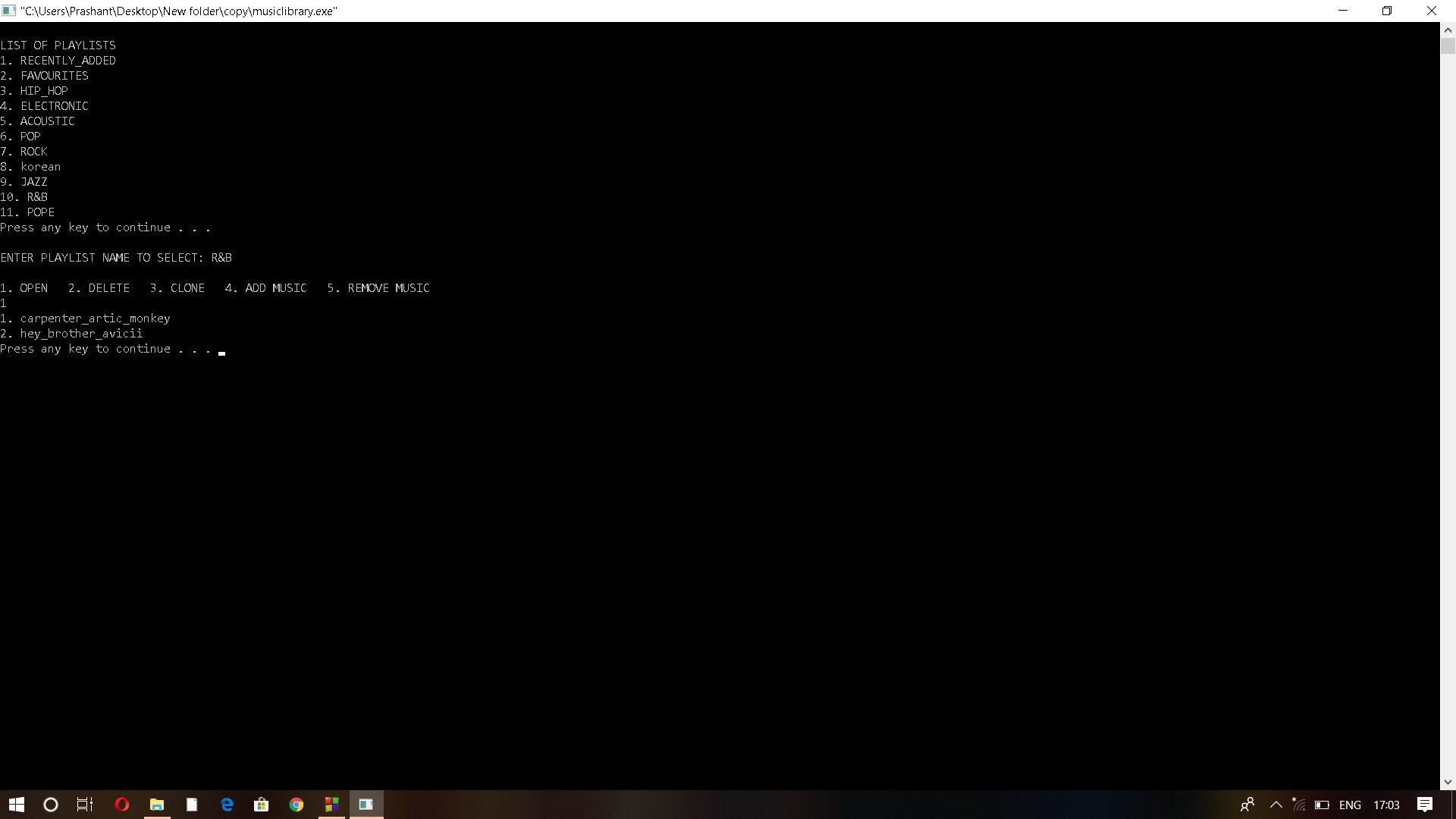
**CHAPTER 5**

**OUTPUT SNAPSHOTS**

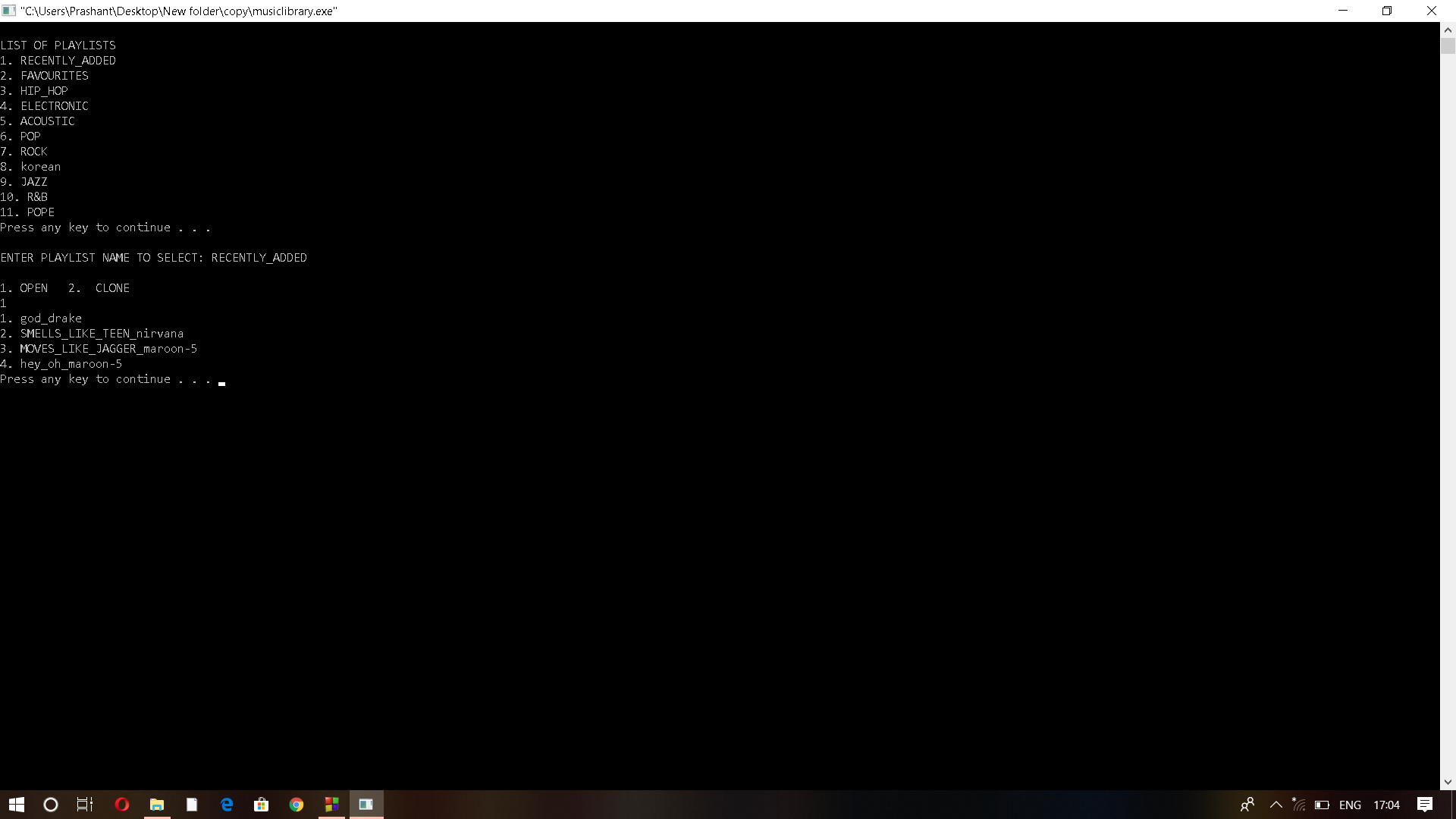
Main menu:

****

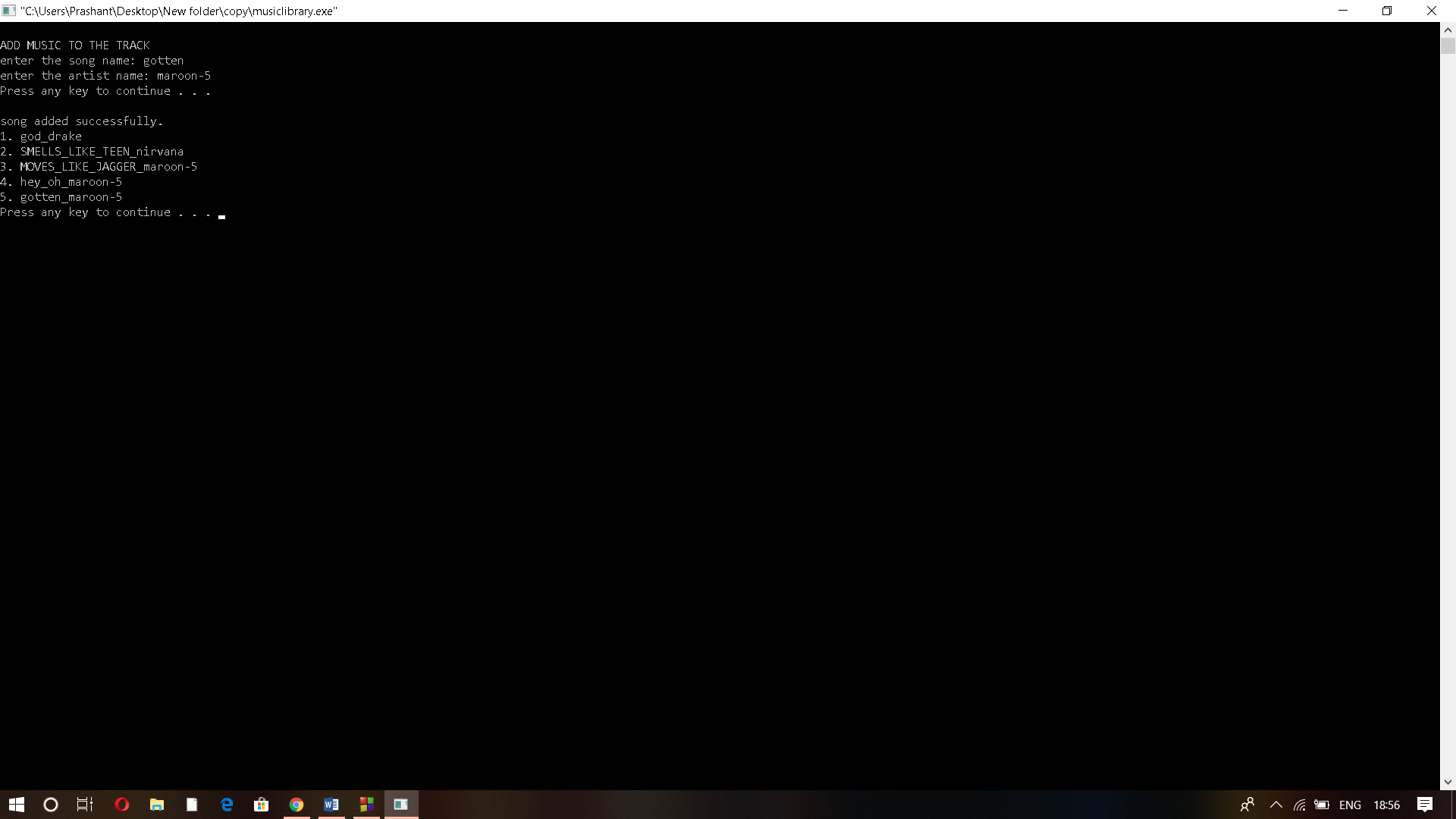
Opening a user created playlist:

****

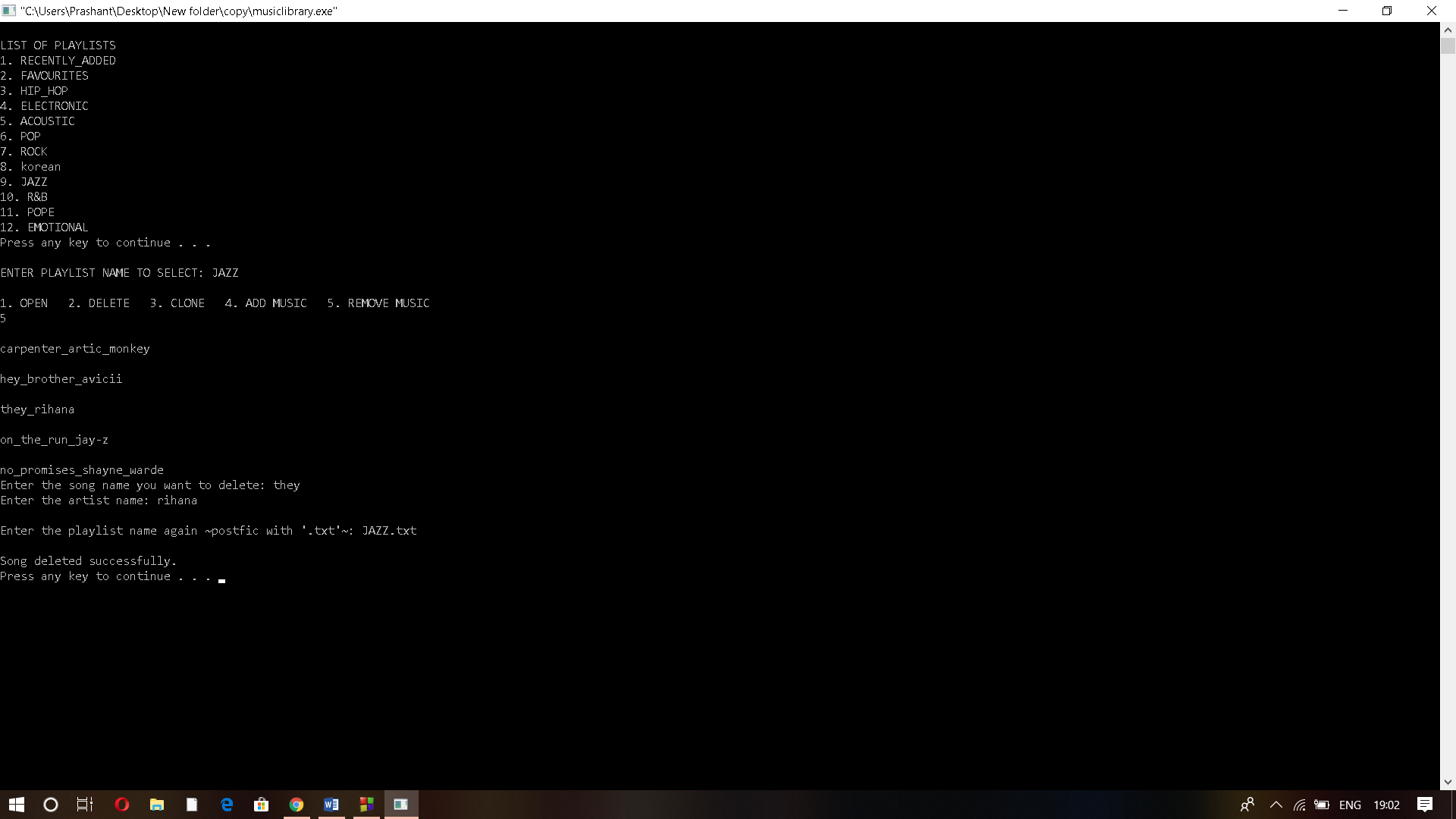
Opening a default playlist:

****

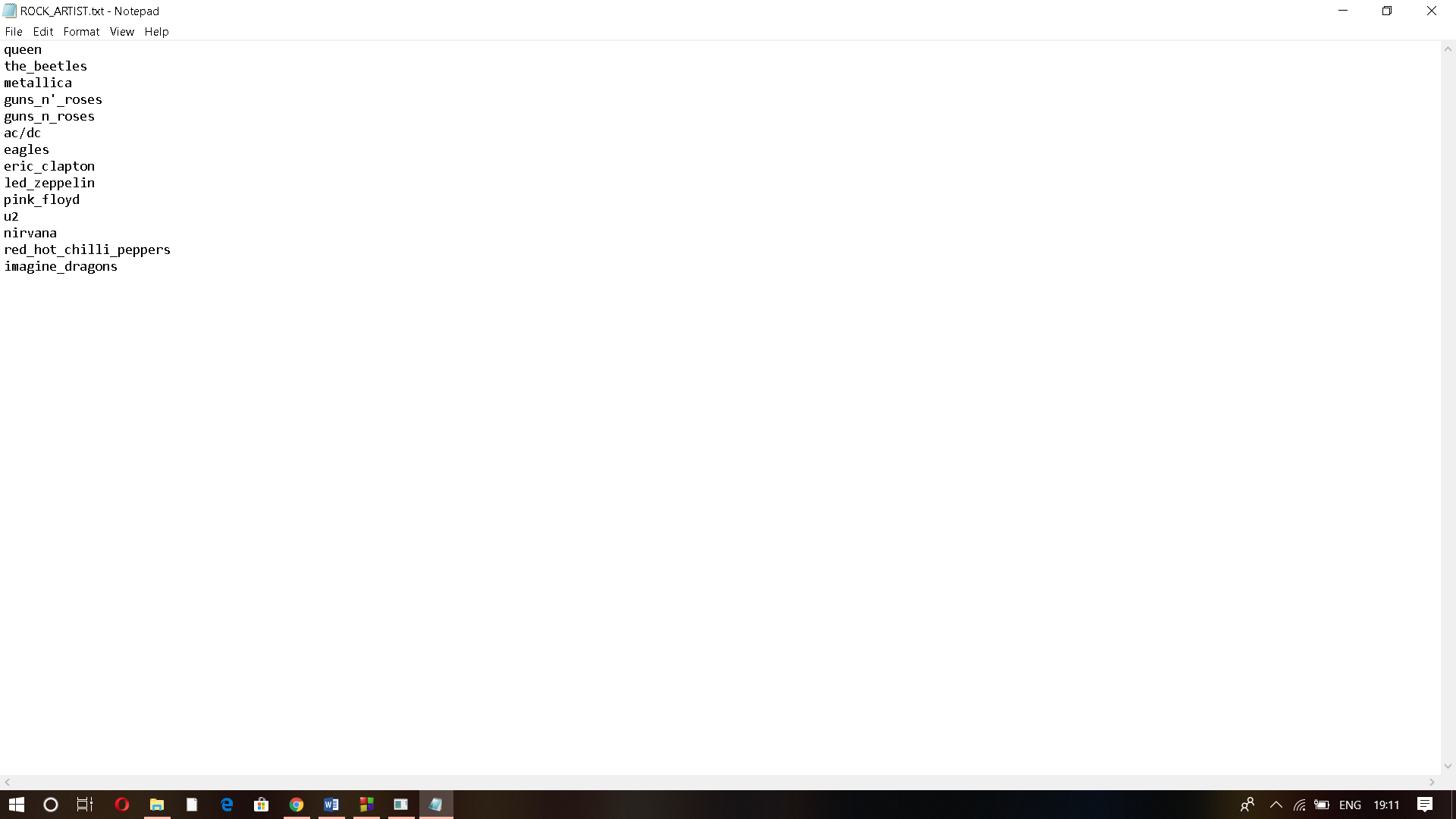
Adding songs in the track:

****

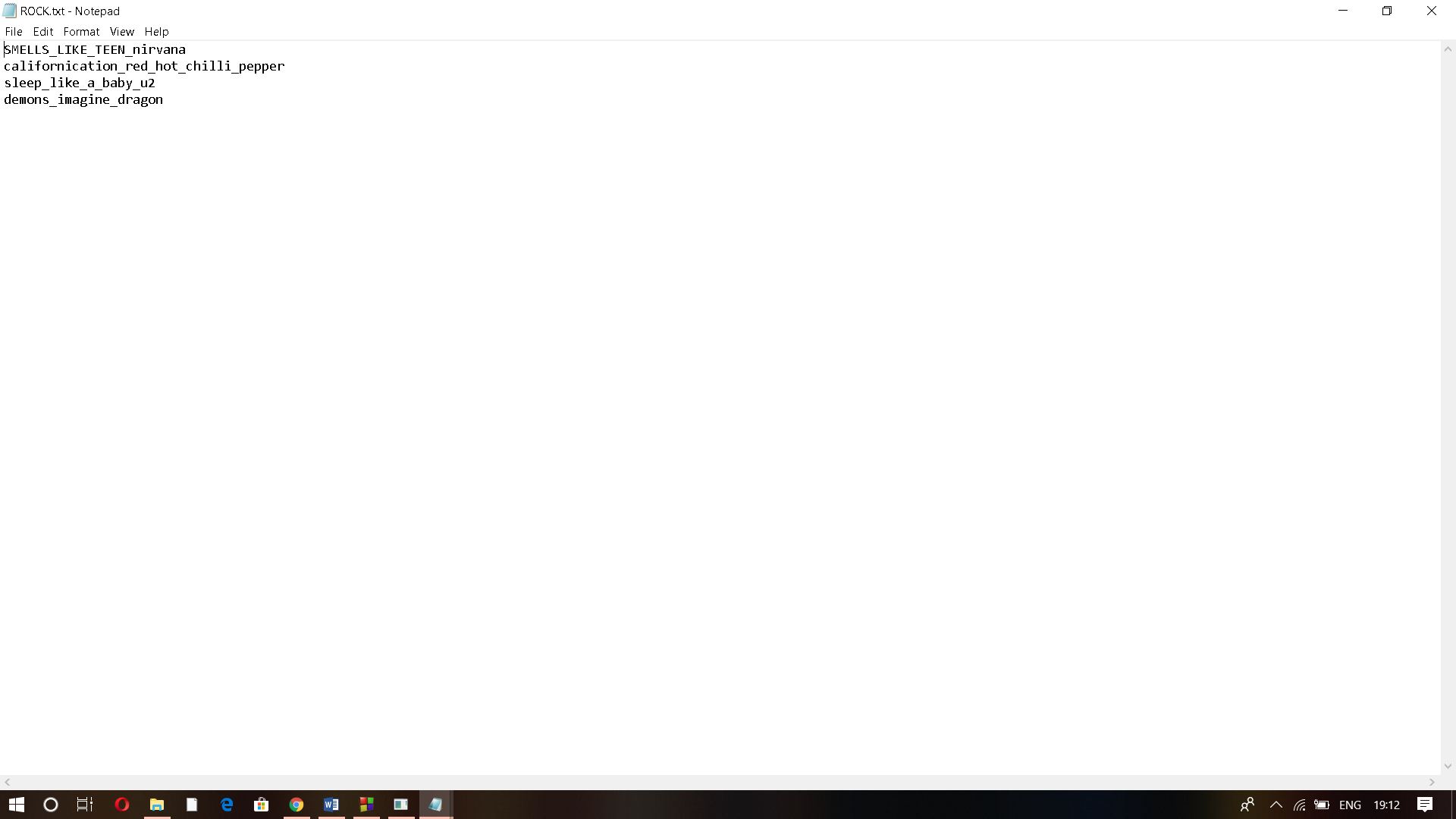
Delete song from a playlist:

****

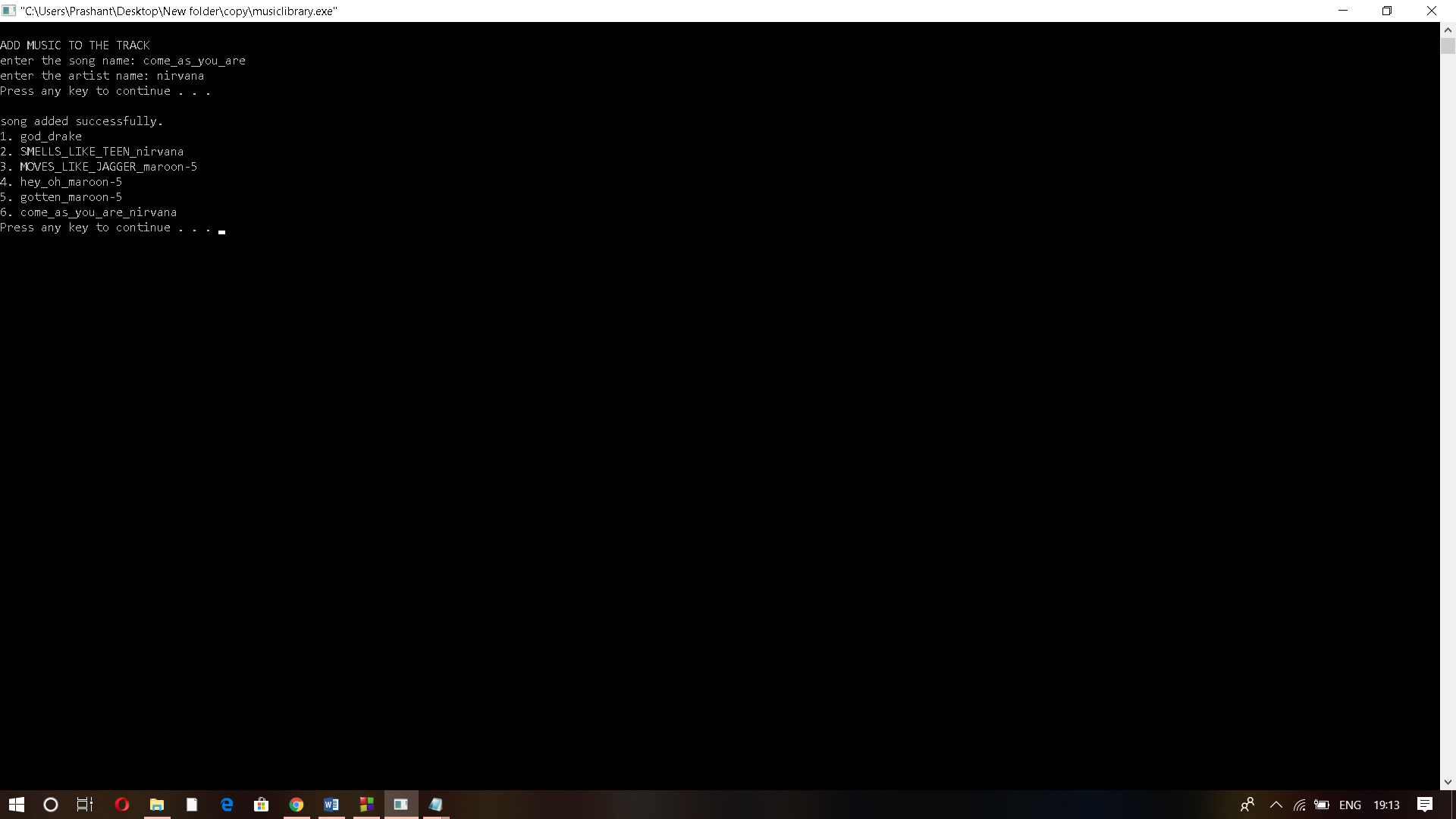
Artists in rock playlist:

****

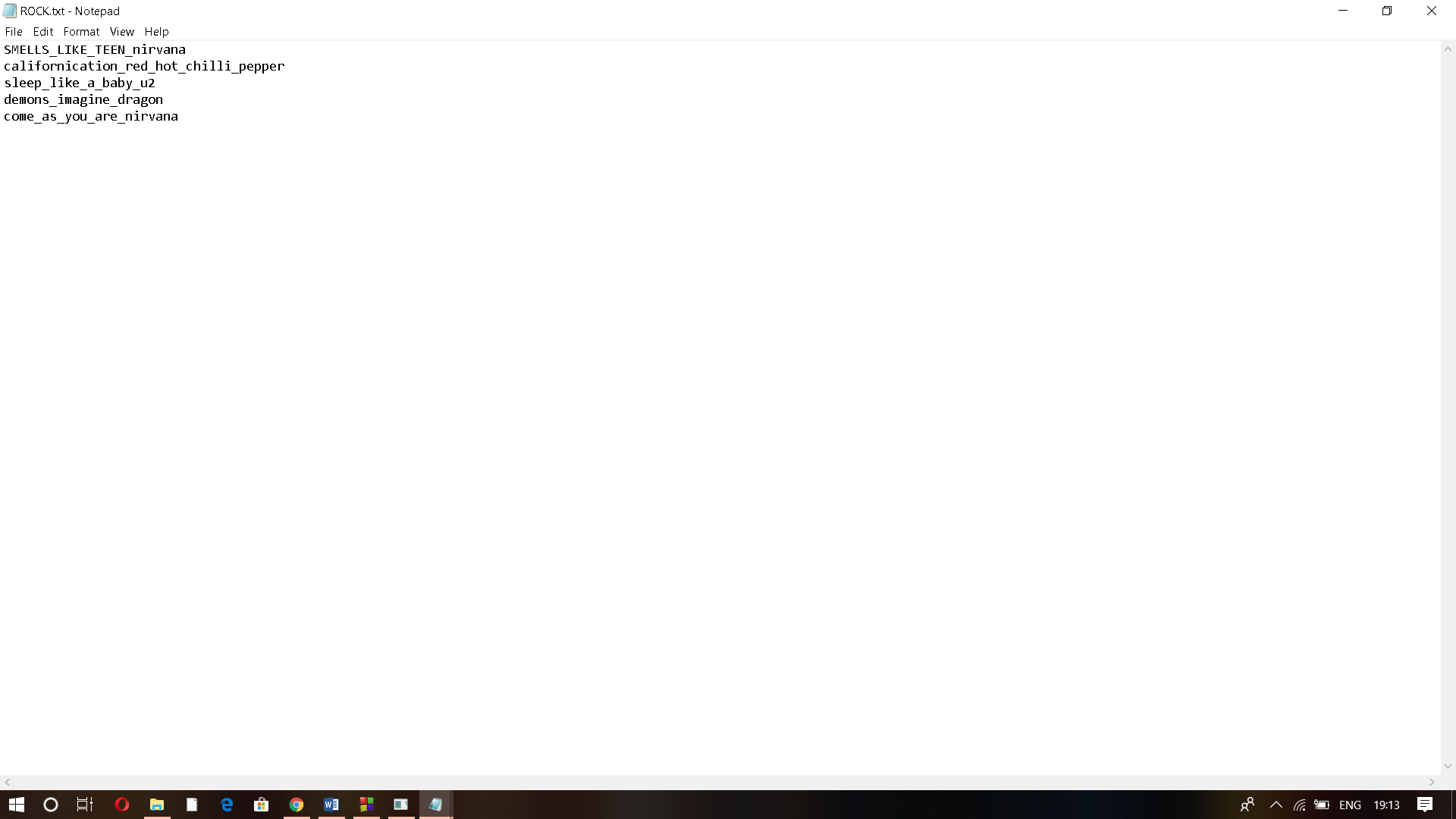
Songs on rock playlist before adding to the track:

****

Adding song to rock playlist through track:

****

Song added successfully in rock playlist:

****

**CHAPTER 6**

**CONCLUSION**

The Projects helps to organize songs in the playlist in an orderly manner. The project implies the concepts of oop. One of the main concepts of oops like inheritance has been implemented in this project. The project is working properly and output is coming as expected.

**CHAPTER 7**

**BIBLIOGRAPHY**

* Object Oriented Programming with C++ - Balagurusamy.
* Geeksforgeeks.
* Tutorialspoint.
* Google.
* btechsmartclass.com