PIMPRI CHINCHWAD EDUCATION TRUST's.

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

(An Autonomous Institute)

S.Y. B. TECH **Year:** 2024 – 25 **Semester:** I

Name: Abhishek Joshi **PRN: 123B1B150**

Department: Computer Engineering **Division:** C (C1)

Course: Data Structures Laboratory **Course Code:** BCE23PC02

Date:

Assignment –3

- **Aim:**

Consider the playlist in a music player. Implement a playlist feature in music player application using singly linked list. Each song in the playlist is represented as a node in the linked list. Each node contains information about the song (such as title or artist or duration, etc.). The playlist should allow users to:
a. Add songs
b. Remove songs
c. Display the entire playlist
d. Play specific songs

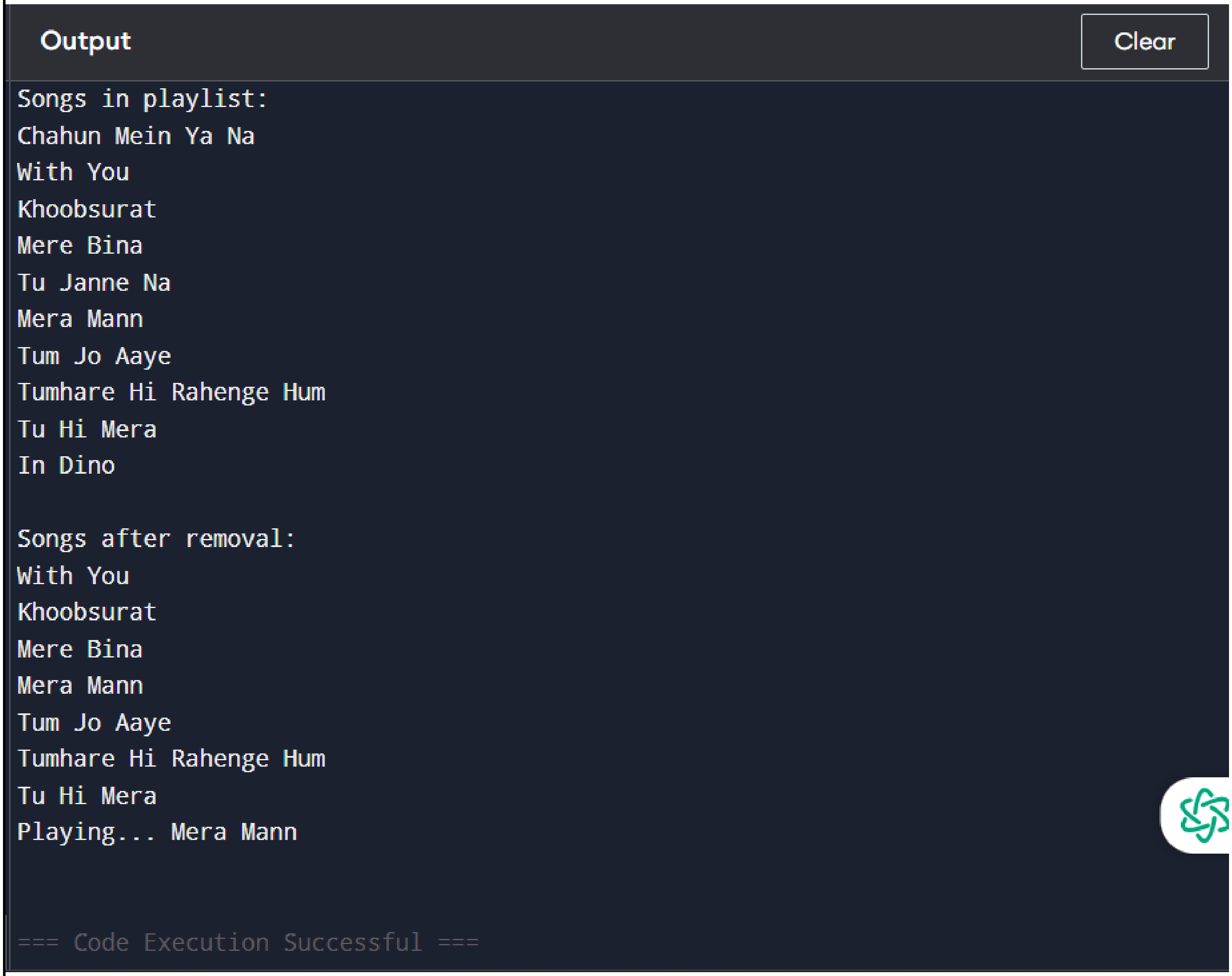
- **Source Code :**

```
#include<iostream>
using namespace std;

class Song{
   public:
   string Title;
   Song* nextSong;

   Song(){
      Title = "";
      nextSong = NULL;
   }
   Song(string newTitle){
      Title = newTitle;
      nextSong = NULL;
   }
};
class Playlist{
   public:
   Song *firstSong = NULL;
   void  addSongAtFirst(string newTitle );
```

```
    void  addSongAtLast(string newTitle );
    void  addSongBet(string newTitle, int key);
    void  removeSongAtFirst( );
    void  removeSongAtLast();
    void removeSongBet(int key);
    void  DisplaySongs();
    void  PlaySpecificSong(string name);
};

void Playlist::addSongAtFirst(string newTitle ){
    Song *newSong = new Song(newTitle);
    if(firstSong == NULL){
        firstSong = newSong;
        return;
    }
    newSong->nextSong = firstSong;
    firstSong = newSong;
}
void Playlist::addSongAtLast(string newTitle ){
    Song* newSong = new Song(newTitle);
    Song* temp = firstSong;
    while(temp->nextSong != NULL){
        temp = temp->nextSong;
    }
    temp->nextSong = newSong;
}
void Playlist::addSongBet(string newTitle, int key){
    int pos=0;
    Song *newSong = new Song(newTitle);
    Song* temp = firstSong;
    while((pos+1) != key){
        temp = temp->nextSong;
        pos++;
    }
    newSong->nextSong = temp->nextSong;
    temp->nextSong = newSong;

}

void Playlist::removeSongAtFirst(){
    Song* temp = firstSong;
    firstSong = temp->nextSong;
    free(temp);
}
```

```
void Playlist::removeSongAtLast(){
    Song* last2nd = firstSong;
    while(last2nd->nextSong->nextSong != NULL){
        last2nd = last2nd->nextSong;
    }
    Song* temp = last2nd->nextSong;
    last2nd->nextSong = NULL;
    free(temp);
}
void Playlist::removeSongBet(int key){
    int pos = 0;
    Song* songAtpos = firstSong;

    while((pos+1) != key){
        if(key == 0){
            removeSongAtFirst();
            return;
        }
        songAtpos = songAtpos->nextSong;
        pos++;
    }
    Song* temp = songAtpos->nextSong;
    songAtpos->nextSong = songAtpos->nextSong->nextSong;
    free(temp);
}

void Playlist::DisplaySongs(){
    Song *temp = firstSong;
    while(temp->nextSong != NULL){
        cout << temp->Title << endl;
        temp = temp->nextSong;
    }
    cout<<temp->Title<<endl;
}

void Playlist::PlaySpecificSong(string name){
    Song *temp = firstSong;
    while(temp->nextSong->Title != name && temp->nextSong != NULL){
        temp = temp->nextSong;
    }
    if(temp->nextSong != NULL)
        cout<<"Playing..."<<temp->nextSong->Title<<endl;
    else
        cout<<"No song found."<<endl;
}

int main() {
    Playlist list;
    list.addSongAtFirst("Tu Janne Na");
    list.addSongAtFirst("Mere Bina");
    list.addSongAtFirst("With You");
    list.addSongAtFirst("Chahun mein ya na");
    list.addSongAtLast("Mera Mann");
    list.addSongAtLast("Tumhare hi rahenge hum");
    list.addSongAtLast("Tu hi mera");
    list.addSongAtLast("In Dino");
    list.addSongBet("Khoobsurat", 2);
    list.addSongBet("Tum jo aaye", 6);
        cout<<"Songs are------------------------- "<<endl;
        list.DisplaySongs();
    list.removeSongAtFirst();
    list.removeSongAtLast();
    list.removeSongBet(3);
        cout<<"Songs are------------------------- "<<endl;
        list.DisplaySongs();
    list.PlaySpecificSong("Mera Mann");
    return 0;
}
```

- **Screen Shot of Output :**



```
Songs in playlist:
Chahun Mein Ya Na
With You
Khoobsurat
Mere Bina
Tu Janne Na
Mera Mann
Tum Jo Aaye
Tumhare Hi Rahenge Hum
Tu Hi Mera
In Dino

Songs after removal:
With You
Khoobsurat
Mere Bina
Mera Mann
Tum Jo Aaye
Tumhare Hi Rahenge Hum
Tu Hi Mera
Playing... Mera Mann


=== Code Execution Successful ===
```

- **Conclusion:**

  Hence, we studied about application of Singly Linked List such as Adding Node, Deleting Node and Displaying List with their algorithm and programs.