20IT092 **Shreeram S**

Question 1 Ex.No: 5

AIM :

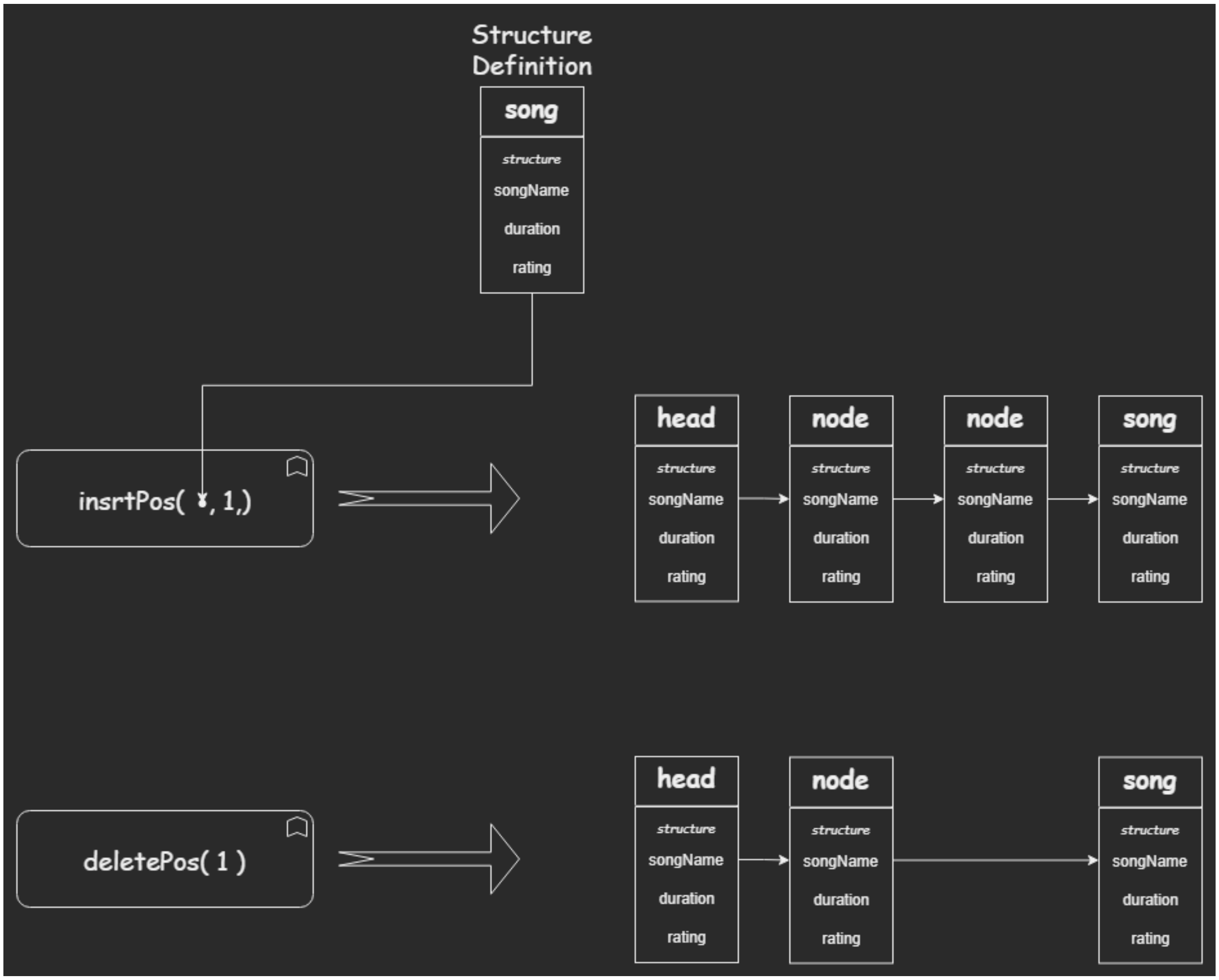
Implementation of Music Player Playlist using Linked List ADT.

Procedure :

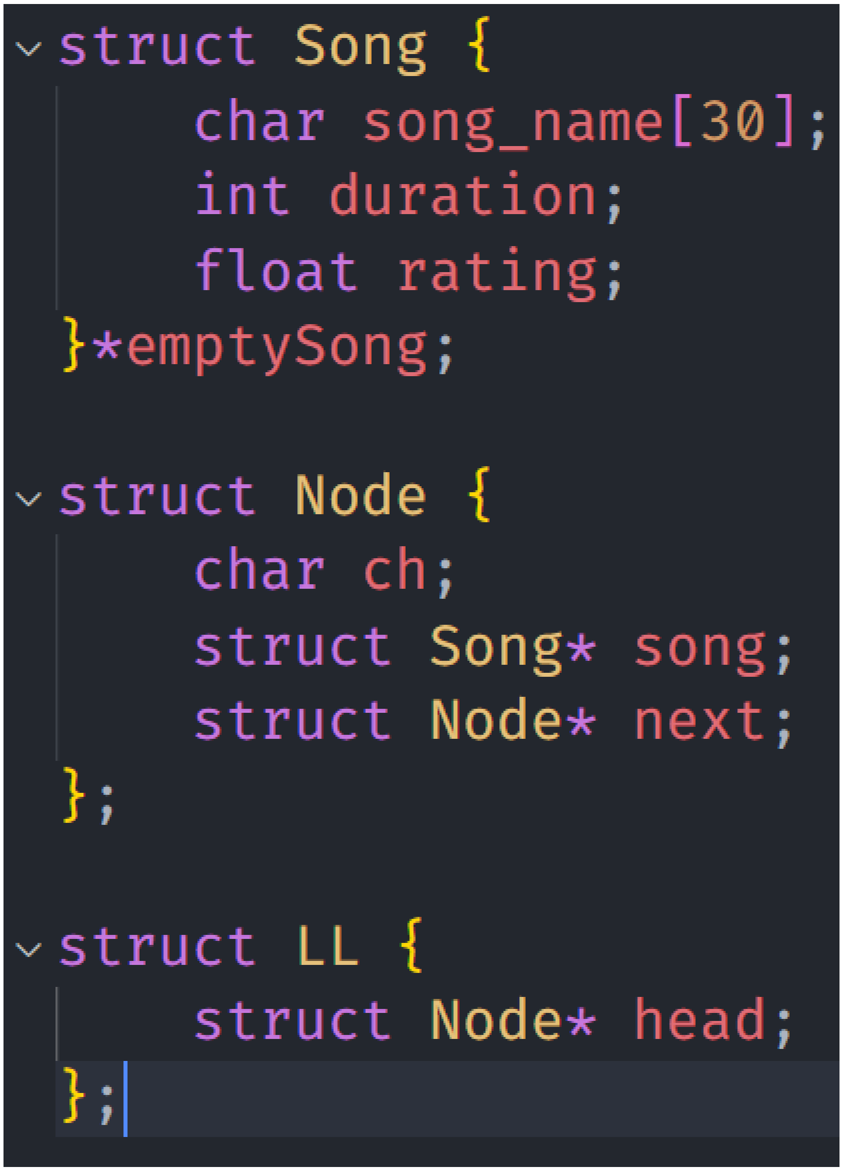
Structure is used for implementation, song is the name of the song, ll is the name of the playlist structure.

```
insert(song s, int pos):
START
newNd <- new empty Node
nwNd->song = song;
if insert at first or head not NULL:
    if head is !NULL:
        head = head->next;
        head = nwNd;
    else:
        head = nwNd;
        head->song = song;
else:
    ptr = ll->head;
    int i = 0;

    loop ptr till it reaches pos:
    if ptr->next is not NULL:
        nwNd->next = ptr->next;
        ptr->next = nwNd;
    else:
        nwNd->next = NULL;
        ptr->next = nwNd;

dltNext(ll, node dlt):
    if dlt and dlt->next is not NULL:
        ptr = dlt->next;
        x = ptr->song
        dlt->next = dlt->next->next
        delete ptr
        return x
```

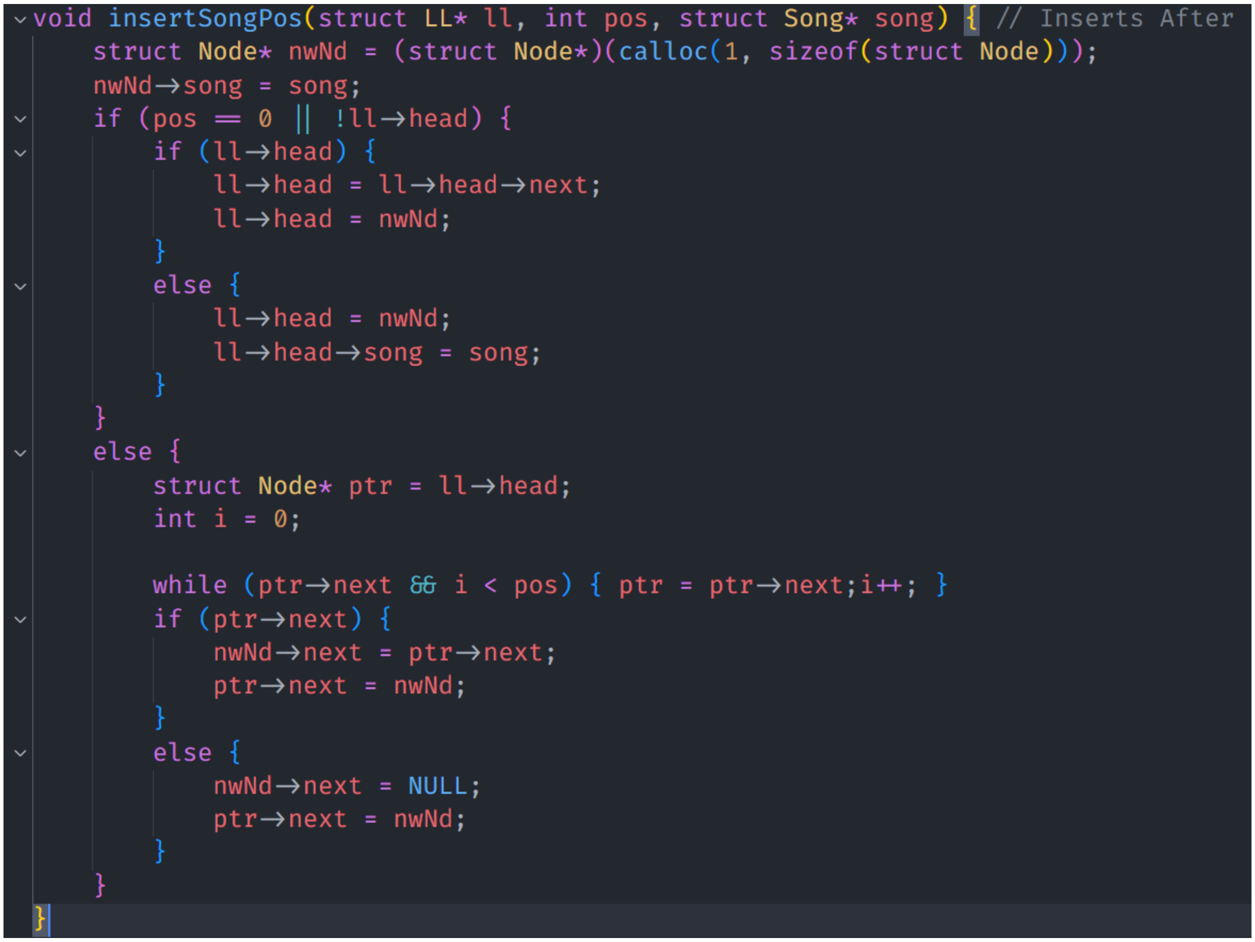
DIAGRAMS :


Structure Definition
song
structure
songName
duration
rating
insrtPos(, 1,)
head
structure
songName
duration
rating
node
structure
songName
duration
rating
node
structure
songName
duration
rating
song
structure
songName
duration
rating
deletePos(1)
head
structure
songName
duration
rating
node
structure
songName
duration
rating
song
structure
songName
duration
rating

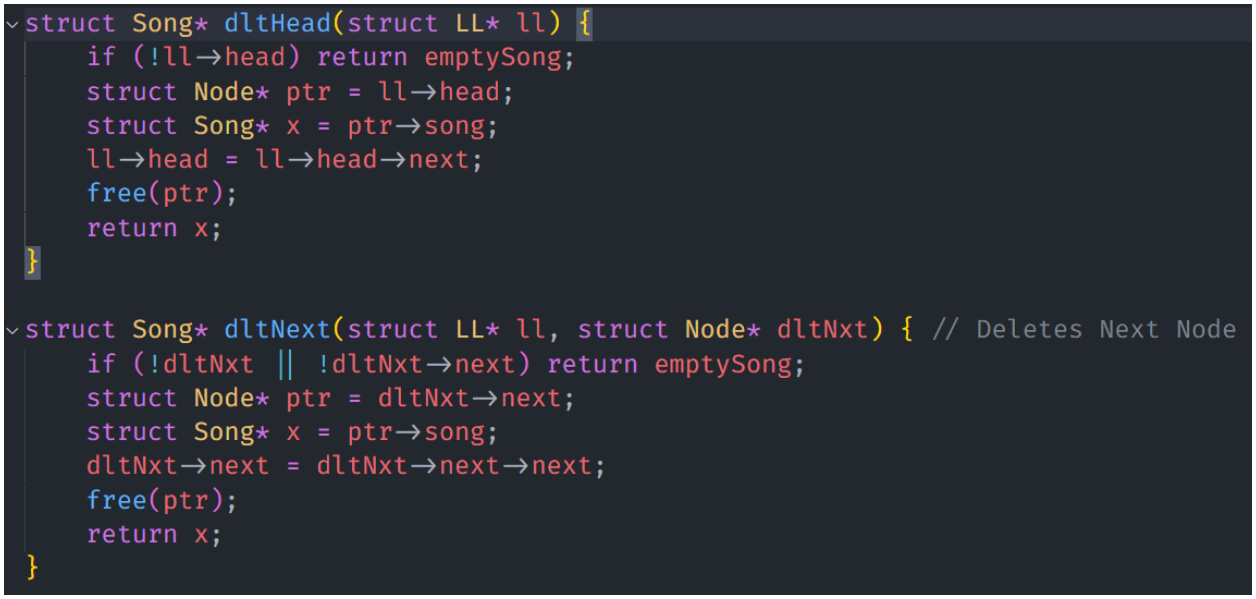
CODE :



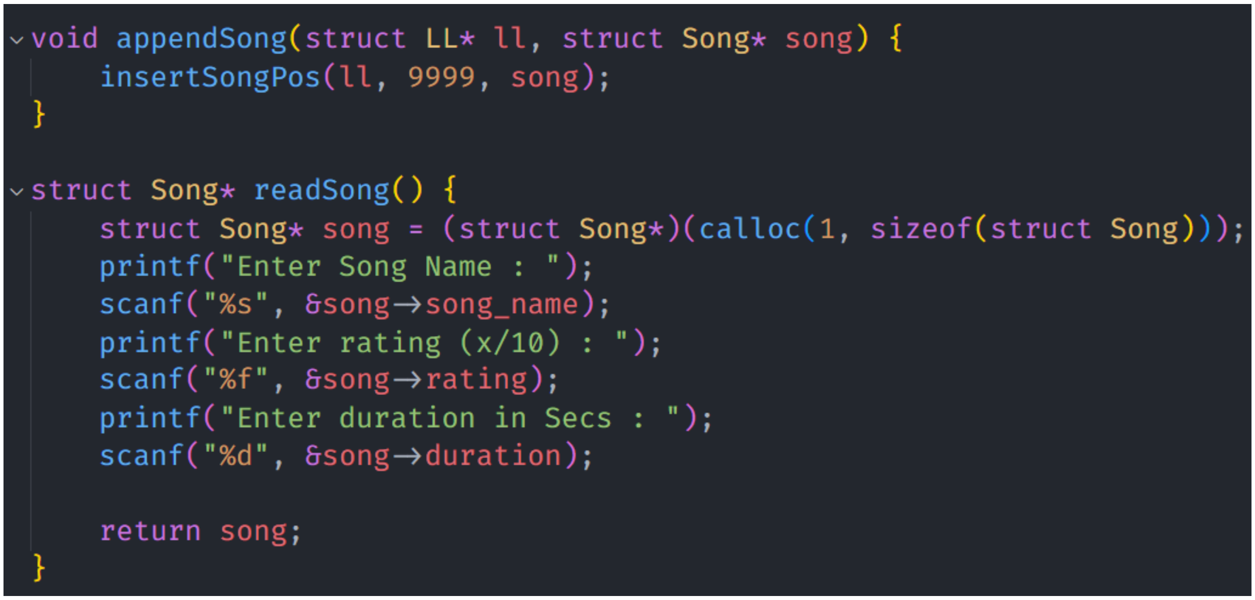
```c
struct Song {
    char song_name[30];
    int duration;
    float rating;
}*emptySong;

struct Node {
    char ch;
    struct Song* song;
    struct Node* next;
};

struct LL {
    struct Node* head;
};
```



```c
void insertSongPos(struct LL* ll, int pos, struct Song* song) { // Inserts After
    struct Node* nwNd = (struct Node*)(calloc(1, sizeof(struct Node)));
    nwNd->song = song;
    if (pos == 0 || !ll->head) {
        if (ll->head) {
            ll->head = ll->head->next;
            ll->head = nwNd;
        }
        else {
            ll->head = nwNd;
            ll->head->song = song;
        }
    }
    else {
        struct Node* ptr = ll->head;
        int i = 0;

        while (ptr->next && i < pos) { ptr = ptr->next;i++; }
        if (ptr->next) {
            nwNd->next = ptr->next;
            ptr->next = nwNd;
        }
        else {
            nwNd->next = NULL;
            ptr->next = nwNd;
        }
    }
}
```

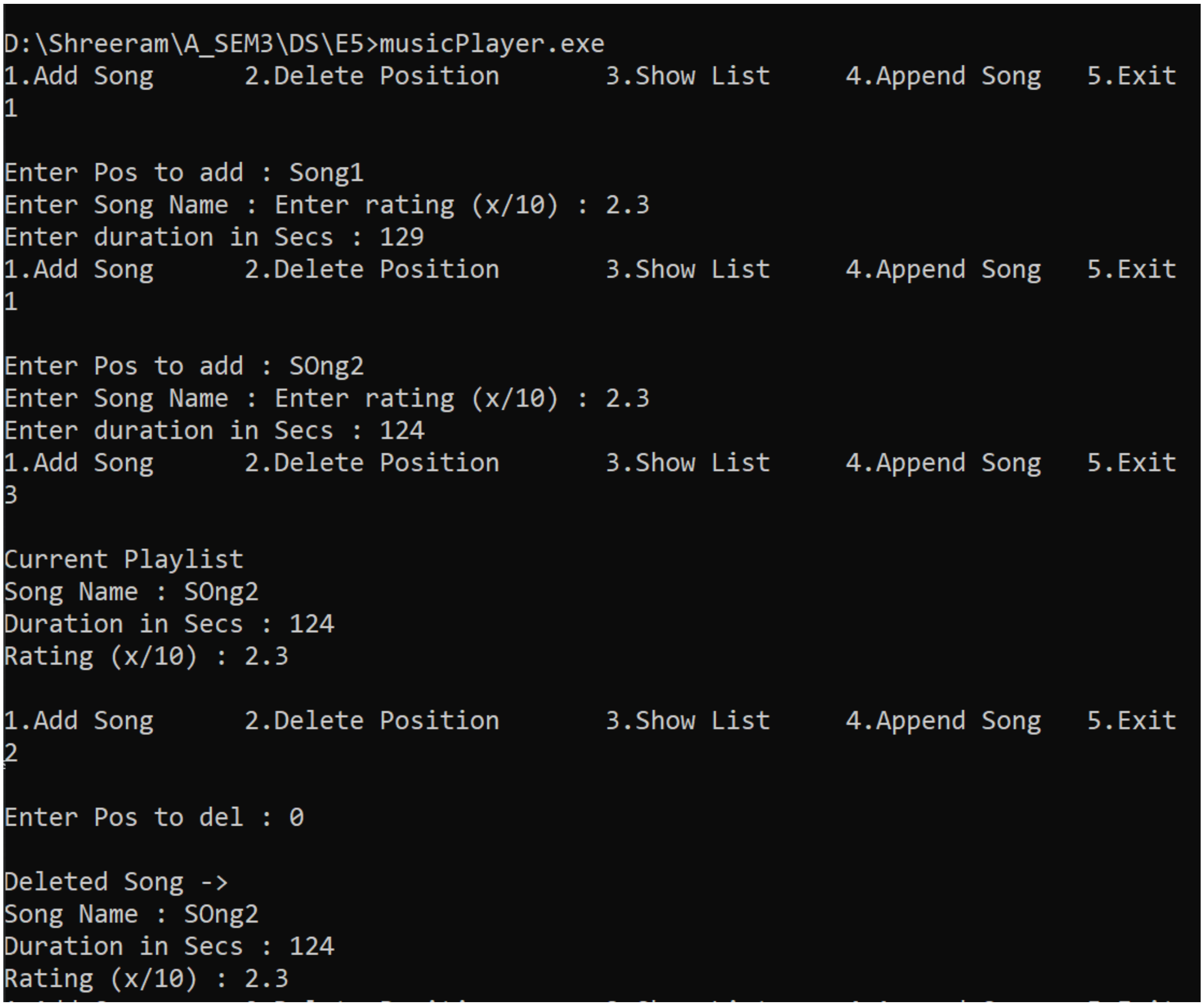


```c
struct Song* dltHead(struct LL* ll) {
    if (!ll->head) return emptySong;
    struct Node* ptr = ll->head;
    struct Song* x = ptr->song;
    ll->head = ll->head->next;
    free(ptr);
    return x;
}

struct Song* dltNext(struct LL* ll, struct Node* dltNxt) { // Deletes Next Node
    if (!dltNxt || !dltNxt->next) return emptySong;
    struct Node* ptr = dltNxt->next;
    struct Song* x = ptr->song;
    dltNxt->next = dltNxt->next->next;
    free(ptr);
    return x;
}
```



```c
void appendSong(struct LL* ll, struct Song* song) {
    insertSongPos(ll, 9999, song);
}

struct Song* readSong() {
    struct Song* song = (struct Song*)(calloc(1, sizeof(struct Song)));
    printf("Enter Song Name : ");
    scanf("%s", &song->song_name);
    printf("Enter rating (x/10) : ");
    scanf("%f", &song->rating);
    printf("Enter duration in Secs : ");
    scanf("%d", &song->duration);

    return song;
}
```

OUTPUT :



```
D:\Shreeram\A_SEM3\DS\E5>musicPlayer.exe
1.Add Song      2.Delete Position       3.Show List     4.Append Song   5.Exit
1

Enter Pos to add : Song1
Enter Song Name : Enter rating (x/10) : 2.3
Enter duration in Secs : 129
1.Add Song      2.Delete Position       3.Show List     4.Append Song   5.Exit
1

Enter Pos to add : SOng2
Enter Song Name : Enter rating (x/10) : 2.3
Enter duration in Secs : 124
1.Add Song      2.Delete Position       3.Show List     4.Append Song   5.Exit
3

Current Playlist
Song Name : SOng2
Duration in Secs : 124
Rating (x/10) : 2.3

1.Add Song      2.Delete Position       3.Show List     4.Append Song   5.Exit
2

Enter Pos to del : 0

Deleted Song ->
Song Name : SOng2
Duration in Secs : 124
Rating (x/10) : 2.3
```