```c
//                  ========== DSA CASE STUDIES ==========
//                     By: Akshay S and B Vikram Seervi
//
//               ======== 1. MUSIC PLAYLIST SYSTEM ========

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct Song {
  char title[50];
  char artist[50];
  char genre[30];
  struct Song *next;
  struct Song *prev;
};

struct Song *head = NULL;
struct Song *tail = NULL;
struct Song *current = NULL;

void addSong(char title[], char artist[], char genre[]);
void displayPlaylist();
void displayCurrentSong();
void playNext();
void playPrevious();
void switchToSong(char title[]);
void displayByGenre();

int main() {
  int choice;
  char title[50], artist[50], genre[30];

  do {
    printf("\n=== Music Playlist System ===\n");
    printf("1. Add a Song\n");
    printf("2. Display Playlist\n");
    printf("3. Display Current Song\n");
    printf("4. Play Next Song\n");
    printf("5. Play Previous Song\n");
    printf("6. Switch to a Song\n");
    printf("7. Display Songs by Genre\n");
    printf("8. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);
    getchar(); // Clear input buffer

    switch (choice) {
    case 1:
      printf("Enter song title: ");
      scanf(" %[^\n]", title);
      printf("Enter artist: ");
      scanf(" %[^\n]", artist);
      printf("Enter genre: ");
      scanf(" %[^\n]", genre);
      addSong(title, artist, genre);
      break;
```

```c
    case 2:
      displayPlaylist();
      break;
    case 3:
      displayCurrentSong();
      break;
    case 4:
      playNext();
      break;
    case 5:
      playPrevious();
      break;
    case 6:
      printf("Enter song title to switch to: ");
      scanf(" %[^\n]", title);
      switchToSong(title);
      break;
    case 7:
      displayByGenre();
      break;
    case 8:
      printf("Exiting... Thank you!\n");
      break;
    default:
      printf("Invalid choice. Please try again.\n");
      break;
    }
  } while (choice != 8);

  return 0;
}

// Add a song to the playlist
void addSong(char title[], char artist[], char genre[]) {
  struct Song *newSong = (struct Song *)malloc(sizeof(struct Song));
  strcpy(newSong->title, title);
  strcpy(newSong->artist, artist);
  strcpy(newSong->genre, genre);
  newSong->next = NULL;
  newSong->prev = NULL;

  if (head == NULL) {
    head = tail = newSong;
  } else {
    tail->next = newSong;
    newSong->prev = tail;
    tail = newSong;
  }

  if (current == NULL) {
    current = head;
  }

  printf("Song '%s' by %s added to the playlist.\n", title, artist);
}
```

```c
// Display all songs in the playlist
void displayPlaylist() {
  struct Song *temp = head;
  if (temp == NULL) {
    printf("Playlist is empty.\n");
    return;
  }
  printf("\n=== Playlist ===\n");
  while (temp != NULL) {
    printf("Title: %s, Artist: %s, Genre: %s\n", temp->title, temp->artist,
           temp->genre);
    temp = temp->next;
  }
}

// Play the next song
void playNext() {
  if (current == NULL) {
    printf("No song is currently playing.\n");
    return;
  }
  if (current->next != NULL) {
    current = current->next;
    printf("Now playing: '%s' by %s\n", current->title, current->artist);
  } else {
    printf("You are at the last song in the playlist.\n");
  }
}

// Play the previous song
void playPrevious() {
  if (current == NULL) {
    printf("No song is currently playing.\n");
    return;
  }
  if (current->prev != NULL) {
    current = current->prev;
    printf("Now playing: '%s' by %s\n", current->title, current->artist);
  } else {
    printf("You are at the first song in the playlist.\n");
  }
}

// Switch to a specific song
void switchToSong(char title[]) {
  struct Song *temp = head;
  while (temp != NULL) {
    if (strcmp(temp->title, title) == 0) {
      current = temp;
      printf("Switched to: '%s' by %s\n", current->title, current->artist);
      return;
    }
    temp = temp->next;
  }
  printf("Song '%s' not found in the playlist.\n", title);
}
```

```c
// Display all songs grouped by genres
void displayByGenre() {
  struct Song *temp = head;
  char genres[100][30];
  int genreCount = 0;

  if (temp == NULL) {
    printf("Playlist is empty.\n");
    return;
  }

  // Collect unique genres
  while (temp != NULL) {
    int printed = 0;
    for (int i = 0; i < genreCount; i++) {
      if (strcmp(genres[i], temp->genre) == 0) {
        printed = 1;
        break;
      }
    }
    if (!printed) {
      strcpy(genres[genreCount], temp->genre);
      genreCount++;
    }
    temp = temp->next;
  }

  // Display songs grouped by genres
  printf("\n=== Songs Grouped by Genres ===\n");
  for (int i = 0; i < genreCount; i++) {
    printf("\n--- Genre: %s ---\n", genres[i]);
    temp = head;
    while (temp != NULL) {
      if (strcmp(temp->genre, genres[i]) == 0) {
        printf("Title: %s, Artist: %s\n", temp->title, temp->artist);
      }
      temp = temp->next;
    }
  }
}

// Display the currently playing song
void displayCurrentSong() {
  if (current == NULL) {
    printf("\nNo song is currently playing.\n");
  } else {
    printf("\n=== Currently Playing ===\n");
    printf("Title: %s\n", current->title);
    printf("Artist: %s\n", current->artist);
    printf("Genre: %s\n", current->genre);
  }
}

//              ============= THANK YOU =================
```

```
vikram@RealmeBook:~/dsa$ gcc musicPlaylistSystem.c
vikram@RealmeBook:~/dsa$ ./a.out

=== Music Playlist System ===
1. Add a Song
2. Display Playlist
3. Display Current Song
4. Play Next Song
5. Play Previous Song
6. Switch to a Song
7. Display Songs by Genre
8. Exit
Enter your choice: 1
Enter song title: Faded
Enter artist: Alan Walker
Enter genre: Electronic
Song 'Faded' by Alan Walker added to the playlist.

=== Music Playlist System ===
1. Add a Song
2. Display Playlist
3. Display Current Song
4. Play Next Song
5. Play Previous Song
6. Switch to a Song
7. Display Songs by Genre
8. Exit
Enter your choice: 1
Enter song title: Mockingbird
Enter artist: Eminem
Enter genre: Rap
Song 'Mockingbird' by Eminem added to the playlist.

=== Music Playlist System ===
1. Add a Song
2. Display Playlist
3. Display Current Song
4. Play Next Song
5. Play Previous Song
6. Switch to a Song
7. Display Songs by Genre
8. Exit
Enter your choice: 1
Enter song title: Lovely
Enter artist: Billie Eilish
Enter genre: Electronic
Song 'Lovely' by Billie Eilish added to the playlist.

=== Music Playlist System ===
1. Add a Song
2. Display Playlist
3. Display Current Song
4. Play Next Song
5. Play Previous Song
6. Switch to a Song
7. Display Songs by Genre
8. Exit
Enter your choice: 2

=== Playlist ===
Title: Faded, Artist: Alan Walker, Genre: Electronic
Title: Mockingbird, Artist: Eminem, Genre: Rap
Title: Lovely, Artist: Billie Eilish, Genre: Electronic

=== Music Playlist System ===
1. Add a Song
2. Display Playlist
3. Display Current Song
4. Play Next Song
5. Play Previous Song
6. Switch to a Song
7. Display Songs by Genre
8. Exit
Enter your choice: 3

=== Currently Playing ===
Title: Faded
Artist: Alan Walker
Genre: Electronic

=== Music Playlist System ===
1. Add a Song
2. Display Playlist
3. Display Current Song
4. Play Next Song
5. Play Previous Song
6. Switch to a Song
7. Display Songs by Genre
8. Exit
Enter your choice:
```

```
=== Music Playlist System ===
1. Add a Song
2. Display Playlist
3. Display Current Song
4. Play Next Song
5. Play Previous Song
6. Switch to a Song
7. Display Songs by Genre
8. Exit
Enter your choice: 3

=== Currently Playing ===
Title: Faded
Artist: Alan Walker
Genre: Electronic

=== Music Playlist System ===
1. Add a Song
2. Display Playlist
3. Display Current Song
4. Play Next Song
5. Play Previous Song
6. Switch to a Song
7. Display Songs by Genre
8. Exit
Enter your choice: 4
Now playing: 'Mockingbird' by Eminem

=== Music Playlist System ===
1. Add a Song
2. Display Playlist
3. Display Current Song
4. Play Next Song
5. Play Previous Song
6. Switch to a Song
7. Display Songs by Genre
8. Exit
Enter your choice: 5
Now playing: 'Faded' by Alan Walker

=== Music Playlist System ===
1. Add a Song
2. Display Playlist
3. Display Current Song
4. Play Next Song
5. Play Previous Song
6. Switch to a Song
7. Display Songs by Genre
8. Exit
Enter your choice: 6
Enter song title to switch to: Lovely
Switched to: 'Lovely' by Billie Eilish

=== Music Playlist System ===
1. Add a Song
2. Display Playlist
3. Display Current Song
4. Play Next Song
5. Play Previous Song
6. Switch to a Song
7. Display Songs by Genre
8. Exit
Enter your choice: 7

=== Songs Grouped by Genres ===

--- Genre: Electronic ---
Title: Faded, Artist: Alan Walker
Title: Lovely, Artist: Billie Eilish

--- Genre: Rap ---
Title: Mockingbird, Artist: Eminem

=== Music Playlist System ===
1. Add a Song
2. Display Playlist
3. Display Current Song
4. Play Next Song
5. Play Previous Song
6. Switch to a Song
7. Display Songs by Genre
8. Exit
Enter your choice: 8
Exiting... Thank you!
vikram@RealmeBook:~/dsa$
```

```c
//       ======== 2. MANAGING TABLE RESERVATION FOR A RESTAURANT ========

#include <stdbool.h>
#include <stdio.h>
#include <string.h>

#define MAX_TABLES 10

typedef struct {
  int tableNumber;
  char name[50];
  bool isAvailable;
} Table;

void initializeTables(Table tables[], int size) {
  for (int i = 0; i < size; i++) {
    tables[i].tableNumber = i + 1;
    tables[i].isAvailable = true;
    strcpy(tables[i].name, "\0");
  }
}

void displayTables(Table tables[], int size) {
  printf("Table Status:\n");
  printf("-------------------------\n");
  for (int i = 0; i < size; i++) {
    printf("Table %d:\t", tables[i].tableNumber);
    if (tables[i].isAvailable == true) {
      printf("Available\n");
    } else {
      printf("Reserved by %s\n", tables[i].name);
    }
  }
}

void reserveTable(Table tables[], int size, int tableNumber, char name[]) {
  if (tableNumber > 0 && tableNumber <= size) {
    if (tables[tableNumber - 1].isAvailable) {
      tables[tableNumber - 1].isAvailable = false;
      strcpy(tables[tableNumber - 1].name, name);
      printf("Table %d has been reserved.\n", tableNumber);
    } else {
      printf("Table %d is already reserved.\n", tableNumber);
    }
  } else {
    printf("Invalid table number.\n");
  }
}

void cancelReservation(Table tables[], int size, int tableNumber) {
  if (tableNumber > 0 && tableNumber <= size) {
    if (!tables[tableNumber - 1].isAvailable) {
      tables[tableNumber - 1].isAvailable = true;
      printf("Reservation for Table %d has been canceled.\n", tableNumber);
    } else {
      printf("Table %d is not reserved.\n", tableNumber);
    }
  } else {
    printf("Invalid table number.\n");
  }
}
```

```c
void changeReservation(Table tables[], int size, int oldTable, int newTable) {
  if (oldTable > 0 && oldTable <= size && newTable > 0 && newTable <= size) {
    if (!tables[oldTable - 1].isAvailable && tables[newTable - 1].isAvailable) {
      tables[oldTable - 1].isAvailable = true;
      tables[newTable - 1].isAvailable = false;
      strcpy(tables[newTable - 1].name, tables[oldTable - 1].name);
      strcpy(tables[oldTable - 1].name, "\0");
      printf("Reservation moved from Table %d to Table %d.\n", oldTable,
             newTable);
    } else if (tables[oldTable - 1].isAvailable) {
      printf("Table %d is not currently reserved.\n", oldTable);
    } else {
      printf("Table %d is not available.\n", newTable);
    }
  } else {
    printf("Invalid table numbers.\n");
  }
}

int main() {
  int choice, tableNumber, oldTable, newTable;
  char name[50];
  Table tables[MAX_TABLES];
  initializeTables(tables, MAX_TABLES);
  do {
    printf("\n=== Restaurant Reservation System ===\n1. Display Tables\n2. "
           "Reserve Table\n3. Cancel Reservation\n4. Change Reservation\n");
    printf("5. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);
    switch (choice) {
    case 1:
      displayTables(tables, MAX_TABLES);
      break;
    case 2:
      printf("Enter table number to reserve: ");
      scanf("%d", &tableNumber);
      printf("Enter your name: ");
      scanf(" %[^\n]", name);
      reserveTable(tables, MAX_TABLES, tableNumber, name);
      break;
    case 3:
      printf("Enter table number to cancel reservation: ");
      scanf("%d", &tableNumber);
      cancelReservation(tables, MAX_TABLES, tableNumber);
      break;
    case 4:
      printf("Enter current reserved table number: ");
      scanf("%d", &oldTable);
      printf("Enter new table number to reserve: ");
      scanf("%d", &newTable);
      changeReservation(tables, MAX_TABLES, oldTable, newTable);
      break;
    case 5:
      printf("Exiting...\n");
      break;
    default:
      printf("Invalid choice. Try again.\n");
      break;
    }
  } while (choice != 5);
  return 0;
}
```

```
vikram@RealmeBook:~/dsa$ gcc restaurantTableReservation.c
vikram@RealmeBook:~/dsa$ ./a.out

=== Restaurant Reservation System ===
1. Display Tables
2. Reserve Table
3. Cancel Reservation
4. Change Reservation
5. Exit
Enter your choice: 1
Table Status:
--------------------------
Table 1:        Available
Table 2:        Available
Table 3:        Available
Table 4:        Available
Table 5:        Available
Table 6:        Available
Table 7:        Available
Table 8:        Available
Table 9:        Available
Table 10:       Available

=== Restaurant Reservation System ===
1. Display Tables
2. Reserve Table
3. Cancel Reservation
4. Change Reservation
5. Exit
Enter your choice: 2
Enter table number to reserve: 4
Enter your name: Vikram Seervi
Table 4 has been reserved.

=== Restaurant Reservation System ===
1. Display Tables
2. Reserve Table
3. Cancel Reservation
4. Change Reservation
5. Exit
Enter your choice: 2
Enter table number to reserve: 8
Enter your name: Akshay
Table 8 has been reserved.

=== Restaurant Reservation System ===
1. Display Tables
2. Reserve Table
3. Cancel Reservation
4. Change Reservation
5. Exit
Enter your choice: 1
Table Status:
--------------------------
Table 1:        Available
Table 2:        Available
Table 4:        Reserved by Vikram Seervi
Table 5:        Available
Table 6:        Available
Table 7:        Available
Table 8:        Reserved by Akshay
Table 9:        Available
Table 10:       Available

=== Restaurant Reservation System ===
1. Display Tables
2. Reserve Table
3. Cancel Reservation
4. Change Reservation
5. Exit
Enter your choice:


=== Restaurant Reservation System ===
1. Display Tables
2. Reserve Table
3. Cancel Reservation
4. Change Reservation
5. Exit
Enter your choice: 4
Enter current reserved table number: 4
Enter new table number to reserve: 6
Reservation moved from Table 4 to Table 6.

=== Restaurant Reservation System ===
1. Display Tables
2. Reserve Table
3. Cancel Reservation
4. Change Reservation
5. Exit
Enter your choice: 1
Table Status:
--------------------------
Table 1:        Available
Table 2:        Available
Table 3:        Available
Table 4:        Available
Table 5:        Available
Table 6:        Reserved by Vikram Seervi
Table 7:        Available
Table 8:        Reserved by Akshay
Table 9:        Available
Table 10:       Available

=== Restaurant Reservation System ===
1. Display Tables
2. Reserve Table
3. Cancel Reservation
4. Change Reservation
5. Exit
Enter your choice: 3
Enter table number to cancel reservation: 8
Reservation for Table 8 has been canceled.

=== Restaurant Reservation System ===
1. Display Tables
2. Reserve Table
3. Cancel Reservation
4. Change Reservation
5. Exit
Enter your choice: 1
Table Status:
--------------------------
Table 1:        Available
Table 2:        Available
Table 3:        Available
Table 4:        Available
Table 5:        Available
Table 6:        Reserved by Vikram Seervi
Table 7:        Available
Table 8:        Available
Table 9:        Available
Table 10:       Available

=== Restaurant Reservation System ===
1. Display Tables
2. Reserve Table
3. Cancel Reservation
4. Change Reservation
5. Exit
Enter your choice: 5
Exiting...
vikram@RealmeBook:~/dsa$
```