

C 프로그래밍 및 실습

플레이리스트 추천

진척 보고서 #2

제출일자: 12/10

제출자명: 김도형

제출자학번: 231630

1. 프로젝트 목표

1) 배경 및 필요성

음악 플랫폼 수가 급격히 증가하면서, 음악 플랫폼 간의 경쟁이 치열해짐. 최근 이용자들은 자신의 취향에 맞는 노래를 자동 재생하여 주는 것을 선호함. 여러 음악재생 플랫폼들 사이에서 살아남기 위해서는 사용자들을 구체적으로 분석하고 파악하여 이용자들에게 맞는 노래는 추천해주어야 한다고 생각

2) 프로젝트 목표

이용자들의 나이, 이용 시간대, 선호하는 장르 등을 파악하여 이용자에게 최적의 노래를 추천해 주는 것을 목표로 함

3) 차별점

기존에 존재하는 음악 플랫폼들은 인기차트나 새로운 곡을 소개하는 형식으로 되어있지만, 이 프로그램을 통하여 취향이 비슷한 노래를 추천받고, 선호하는 장르가 비슷한 사람들의 플레이리스트를 공유 받을 수 있다.

2. 기능 계획

1) 기능 1: 재생목록 삭제 및 추가

가장 기본적인 음악 플랫폼의 기능. 사용자들에게 추가할 노래, 삭제할 노래, 수정할 노래, 노래의 장르를 입력 받고 저장 및 출력. (반복문, 함수, 2 차원배열, 조건문, 2 차원 배열 사용 +포인터 사용예정)

2) 기능 2: 사용자 5 명에게 정보를 입력 받고 각자의 재생목록 생성

이용자 5 명에게 이름, 선호 장르, 나이, 이용시간 등을 입력 받고 각자의 플레이리스트를 생성하고 관리하게 함 (2, 3 차원 배열, 반복문, 조건문, 함수 사용)

3) 기능 3: 가장 인기있는 장르 알려주기

이용자들에게 장르를 입력 받을 때 겹치거나, 가장 많이 나온 장르를 인기있는 노래로 인식하고, 가장 인기있는 장르로 출력 (반복문, 조건문, 함수, 배열, 포인터)

4) 기능 4: 고객센터 + 노래 중복 제거 + 나이 입력

이용자들에게 메뉴얼을 입력받고, 고객센터에서 쿠폰을 등록하거나 민원을 요청하고, 이용권을 해지할 수 있는 기능(조건문 사용), 겹치는 노래를 플레이리스트에서 제거하는 기능 추가(delindex 와 유사하게 작성 >strcpy, 함수, 반복문 사용)

모든 기능의 배열에 포인터, 구조체 사용, 함수(가능한 모두)를 헤더파일로 분리해서 작성할 예정

3. 진척사항

1) 기능 구현

(1) 재생목록에 노래/장르 추가, 삭제, 수정, 종료

- 입출력: 선호하는 장르, 노래 제목, 매뉴얼 번호 입력, 사용자 이름
- 설명: 사용자 5 명의 선호 노래와 장르를 입력 받고 플레이리스트 관리 목록을 만듦.

#define 을 이용하여 노래 개수와 이름 길이 등을 정의하고, 장르와 노래, 이름 등을 입력

받을 변수들을 2 차원 배열을 이용하여 선언하였다.

Addsong, delsong, printsong, recommendsong 함수를 통해 재생목록을 관리하는 코드를 작성하였다.

Add song에서는 노래 이름과 노래 장르 사용자 번호를 넘겨주며 노래를 입력 받고, 그 노래를 해당 장르에 추가하는 코드로 구성하였다. Printf 와 scanf_s 를 이용하여 입력받고 Strcpy 를 사용하여 저장하며, songCount 를 1 늘린다

Delsong에서는 삭제할 매뉴얼의 번호와 사용자 번호를 넘겨주며, 삭제할 노래의 인덱스를 입력 받고 삭제하면 songCount 를 1 감소시킨다. for 반복문 사용

Printsonglist에서는 원하는 사용자의 재생목록을 출력해주도록 작성하였다.

For 반복문을 사용하여 사용자 번호에 따른 노래와 노래의 장르를 songCount 만큼 모두 출력한다

- 적용된 배운 내용 (예: scanf, printf, while 반복문, (if, switch)조건문, 함수, 2 차원 배열

(2) 사용자 5 명에게 정보를 입력 받고 각자의 플레이리스트 생성, 관리, 추천

- 입출력: 사용자 이름, 사용자 별 플레이리스트(노래와 장르), 선호 장르와 그에 따른 노래들 목록

- 설명: 2 차원 배열을 통해 장르의 이름과 개수, 그리고 선호하는 장르와 노래를 입력 받을 변수를 선언하였다. 기능 1 에서 작성한 코드에 이름을 입력 받는 코드와 genre, personindex 를 추가하여 추가, 삭제, 수정할 때 사용자의 플레이리스트에서 작업이 실행되게 변경.

Recommendsong에서는 입력 받은 장르가 같은 경우에 그 장르의 노래들을

추천해주도록 작성하였다. For 를 이용한 이중반복문을 통해 입력한 inputgenre 에 해당하는 노래들을 출력한다.

Addsong 함수에서는 입력받은 노래를 peoplefavoritesongs 의 배열에 복사하여 저장하고

songCount 를 셀 때 사용자마다 플레이리스트를 만들었기 때문에 songCount[personindex]++로 수정한다.

Delsong 함수에서는 노래를 삭제할 때 입력한 사용자의 플레이리스트에서 노래를 지우도록 변경하였고, 따라서 songCount[personindex]- - 로 수정한다. 또한 위에서 복사하고 저장해놓은 personfavoritesongs 의 배열에서 [i-1]을 통해 지워준다.

Printsonglist 함수에서는 재생목록을 출력할 때 사용자의 번호를 입력 받고 그 사용자의 플레이리스트만을 출력하도록 for 반복문을 사용하여 작성하였다. 전 과는 다르게 사용자 인덱스, 노래, 장르를 모두 출력하도록 수정하였다.

main 함수에서는 for 반복문을 통하여 사용자 수만큼 이름과 장르, 그리고 가장 좋아하는 노래를 입력 받는다. 전 과는 다르게, 사용자가 여러 명이기 때문에 choice 를 입력 받기 전에 어떤 사용자에게 어떤 작업을 수행할 건지 입력 받는 코드를 추가하였다.

- 적용된 배운 내용 (scanf, printf, while 반복문, (if, switch)조건문, 함수, 2 차원 배열)

(3)가장 인기있는 장르 알려주기

메인함수의 switch 문에서 메뉴얼 7 을 입력했을 때 출력되게 작성. For 문을 통해 사람 수 만큼 반복을 돌리면서 문자열을 비교하는 strcmp 함수를 사용해 장르가 입력받은 수 만큼 genreCount[i]에서 해당되는 것을 ++한다. (genreCount[j]++ 😊)

가장 인기있는 장르의 genre 에서의 인덱스는 popularindex 라는 변수로 지정하고, 가장 인기있는 장르로 genre[popularindex]를 출력한다.

-적용된 배운 내용 (반복문, 조건문, 배열, 인기있는 도시 랭킹 매기기와 유사)

(4)고객센터

Switch 문의 8 을 입력했을 때 실행. 1 또는 2 를 입력 받고, 2 를 입력했을 때는 종료를, 1 을 입력했을 때는 고객센터를 접속. 접속한 후에도 메뉴얼 입력을 통해 구독해지 또는 쿠폰 등록을 할 수 있음. 정해진 쿠폰번호가 입력한 쿠폰번호와 일치하는 경우에 보상을 지급. (적용된 배운 내용 : 조건문)

(5) 현재 노래 수 출력방식 변경

전 기능에는 현재 노래 수를 셀 때 전체 사람의 노래수가 합쳐져서 출력되는 오류가 있었음. > 그래서 메뉴가 출력될 때 현재 노래수를 출력하지 않게 삭제하였고, 개개인의 재생목록을 출력하기위해 3 번 인덱스를 늘렸을 때, 개개인의 현재 노래 개수를 출력하도록 수정. > printSongList 함수에서 (노래 수 : %d), songCount[personindex

로 변경하였다. (-적용된 배운 내용 : 함수, 배열, 반복문)

```
play_list.c (전역 범위) main0
1 //231630 김도형 플레이리스트 프로젝트
2 #include <stdio.h>
3 #include <string.h>
4
5 #define MAX_SONGS 10 //플레이리스트 노래 개수 10개 제한
6 #define SONG_LEN 100 //노래 제목 길이
7 #define GENRE_NAME 100 //장르 이름
8 #define GENRE_NUM 3 //장르 개수
9 #define PERSON_COUNT 5 //이용자 수
10
11
12 char genre[GENRE_NUM][GENRE_NAME] = { "POP", "발라드", "힙합" }; //일단 장르 3개로 제한
13 char songs[MAX_SONGS][SONG_LEN] = { "" }; //노래제목을 입력받는 2차원 배열
14 int songCount[PERSON_COUNT] = { 0 }; //이용자 5명의 노래 개수 세는 변수
15
16 char peopleNames[PERSON_COUNT][50]; //이용자의 이름을 입력받는 변수
17 char peopleFavoriteGenres[PERSON_COUNT][GENRE_NAME]; //이용자의 선호하는 장르를 입력받는 2차원 배열
18 char peopleFavoriteSongs[PERSON_COUNT][MAX_SONGS][SONG_LEN]; //사용자의 선호노래를 입력받는 3차원 배열
19
20
21 void addSong(char song[], char songGenre[], int personIndex, int delIndex) {
22     printf("노래를 입력하세요: ");
23     scanf_s("%s", song, (int)sizeof(song));
24
25     for (int i = 0; i < songCount[personIndex]; i++) {
26
27         if (strcmp(peopleFavoriteSongs[personIndex][i], song) == 0) {
28             printf("이미 재생목록에 있는 노래입니다. 다른 노래를 입력하세요\n");
29             for (int i = delIndex; i < songCount[personIndex]; i++) {
30                 strcpy_s(peopleFavoriteSongs[personIndex][i - 1], sizeof(peopleFavoriteSongs[personIndex][i]), peopleFavoriteSongs[personIndex][i]);
31             }
32             songCount[personIndex]--;
33         }
34     }
35     strcpy_s(peopleFavoriteSongs[personIndex][songCount[personIndex]], sizeof(peopleFavoriteSongs[personIndex][songCount[personIndex]]), song);
36 }
```

```
play_list.c (전역 범위)
37 // 입력한 노래의 장르를 사용자가 선호하는 장르로 저장
38 strcpy_s(peopleFavoriteGenres[personIndex], sizeof(peopleFavoriteGenres[personIndex]), songGenre);
39
40 printf("%s 노래가 '%s' 장르에 추가되었습니다.\n", song, songGenre);
41
42 songCount[personIndex]++;
43 }
44
45 void delSong(int delIndex, int personIndex) {
46     printf("%d. %s : 노래를 삭제합니다.\n", delIndex, peopleFavoriteSongs[personIndex][delIndex - 1]);
47     for (int i = delIndex; i < songCount[personIndex]; i++) {
48         strcpy_s(peopleFavoriteSongs[personIndex][i - 1], sizeof(peopleFavoriteSongs[personIndex][i]), peopleFavoriteSongs[personIndex][i]);
49     }
50     songCount[personIndex]--;
51 }
52
53 void printSongList(int personIndex) {
54     printf("%s의 재생목록 (노래 수: %d) :\n", peopleNames[personIndex], songCount[personIndex]);
55     for (int i = 0; i < songCount[personIndex]; i++) {
56         printf("%d. %s (장르: %s)\n", i + 1, peopleFavoriteSongs[personIndex][i], peopleFavoriteGenres[personIndex]);
57     }
58     printf("\n");
59 }
60
61 void recommendSong(char inputGenre[]) {
62     printf("\n'%s'와 비슷한 장르의 노래를 추천합니다.\n", inputGenre);
63
64
65
66     for (int i = 0; i < PERSON_COUNT; i++) {
67
68         if (strcmp(inputGenre, peopleFavoriteGenres[i]) == 0) {
69
70             for (int j = 0; j < songCount[i]; j++) {
71                 printf("%s 추천 노래 for %s: %s (장르: %s)\n", inputGenre, peopleNames[i], peopleFavoriteSongs[i][j], inputGenre);
72             }
73 }
```



```

79
80
81 int main() {
82     int choice = 0; //메뉴 입력받을 변수
83     int end = 0; //종료 변수
84     int delIndex = -1; //삭제를 위한 변수
85     char ch; //버퍼 제거...?
86     int modifyIndex = 0; //수정을 위한 변수
87     char inputGenre[GENRE_NAME] = ""; //장르이름을 입력받을 변수
88     int genreCount[GENRE_NUM] = { 0 }; // 각 장르의 등장 횟수를 저장하는 배열
89     int select = 0;
90     int center_select = 0;
91     int coupon = 0;
92     int coupon_answer = 231630;
93
94     // 각 이용자의 정보 입력
95     for (int i = 0; i < PERSON_COUNT; i++) {
96         printf("사용자 %d, 이름을 입력하세요: ", i + 1);
97         scanf_s("%s", peopleNames[i], (int)sizeof(peopleNames[i]));
98     }
99     printf("음악 플레이리스트 시작!\n");
100
101     while (1) {
102         printf("-----\n");
103         printf("메뉴를 입력해주세요.\n");
104         printf("1. 재생목록에 노래 추가\n2. 노래 삭제\n3. 재생목록\n4. 종료\n5. 노래 수정\n6. 비슷한 노래 둘러보기\n7. 가장 인기있는 장르\n8. 고객센터\n");
105
106         printf("-----\n");
107         scanf_s("%d", &choice); //실행할 메뉴 입력받음
108
109         switch (choice) { //입력받은 CHOICE에 따라 하는 일을 조건문을 통해 분류
110             case 1:
111                 printf("사용자를 선택하세요 (1-5): ");
112                 int personIndex;
113                 scanf_s("%d", &personIndex);
114
115

```

```

play_list.c (전역 범위)
115
116     if (personIndex <= 0 || personIndex > PERSON_COUNT) {
117         printf("잘못된 선택입니다.\n");
118         break;
119     }
120
121     printf("노래의 장르를 입력하세요 (POP, 발라드, 힙합 등): ");
122     scanf_s("%s", inputGenre, (int)sizeof(inputGenre));
123     addSong(peopleFavoriteSongs[personIndex - 1][songCount[personIndex - 1] % MAX_SONGS], inputGenre, personIndex - 1, delIndex);
124
125     break;
126
127     case 2:
128         printf("사용자를 선택하세요 (1-5): ");
129         scanf_s("%d", &personIndex);
130
131         if (personIndex <= 0 || personIndex > PERSON_COUNT) {
132             printf("잘못된 선택입니다.\n");
133             break;
134         }
135
136         printf("삭제할 노래의 인덱스를 입력하세요: ");
137         scanf_s("%d", &delIndex);
138         if (delIndex > songCount[personIndex - 1] || delIndex <= 0) {
139             printf("삭제할 노래가 없습니다.\n");
140         }
141         else {
142             delSong(delIndex, personIndex - 1);
143         }
144         break;
145
146     case 3:
147         printf("사용자를 선택하세요 (1-5): ");
148         scanf_s("%d", &personIndex);
149
150         if (personIndex <= 0 || personIndex > PERSON_COUNT) {
151             printf("잘못된 선택입니다.\n");

```

```

151         printf("잘못된 선택입니다.\n");
152         break;
153     }
154
155     printSongList(personIndex - 1);
156     break;
157
158 case 4:
159     end = 1;
160     break;
161
162 case 5:
163     printf("사용자를 선택하세요 (1-5): ");
164     scanf_s("%d", &personIndex);
165
166     if (personIndex <= 0 || personIndex > PERSON_COUNT) {
167         printf("잘못된 선택입니다.\n");
168         break;
169     }
170
171     printf("수정할 노래의 번호를 입력해주세요. (1부터 시작): ");
172     scanf_s("%d", &modi fyIndex);
173     ch = getchar();
174     printf("새롭게 추가할 노래를 입력해주세요: ");
175     scanf_s("%s", peopleFavoriteSongs[personIndex - 1][modi fyIndex - 1], (int)sizeof(peopleFavoriteSongs[personIndex - 1][modi fyIndex - 1]));
176     printf("새로운 노래가 추가되었습니다: %d, %s (장르: %s)\n", modi fyIndex, peopleFavoriteSongs[personIndex - 1][modi fyIndex - 1], genre[modi fyIndex - 1]);
177     break;
178
179 case 6:
180     printf("선택하는 장르를 입력하세요 (POP, 발라드, 힙합): ");
181     scanf_s("%s", inputGenre, (int)sizeof(inputGenre));
182     recommendSong(inputGenre);
183     break;
184
185 case 7:
186     for (int i = 0; i < PERSON_COUNT; i++) {
187         for (int j = 0; j < GENRE_NUM; j++) {

```

play_list.c (전역 범위)

```

187         for (int j = 0; j < GENRE_NUM; j++) {
188             if (strcmp(peopleFavoriteGenres[i], genre[j]) == 0) {
189                 genreCount[j]++;
190             }
191         }
192     }
193
194     int popularIndex = 0;
195     for (int i = 1; i < GENRE_NUM; i++) {
196         if (genreCount[i] > genreCount[popularIndex]) {
197             popularIndex = i;
198         }
199     }
200
201     // 가장 인기 있는 장르를 출력
202     printf("가장 인기있는 장르: %s\n", genre[popularIndex]);
203     break;
204
205 case 8:
206
207     printf("고객센터에 접속하셨습니다. 도움이 필요하시면 1, 필요 없다면 2를 입력해주세요 : ");
208     scanf_s("%d", &select);
209     if (select == 2) {
210         printf("고객센터를 종료합니다.");
211     }
212     else if (select == 1) {
213         printf("고객센터 메뉴\n1.해지 신청\n2.쿠폰 등록\n");
214         scanf_s("%d", &center_select);
215         if (center_select == 1) {
216             printf("이용권을 해지합니다.");
217         }
218         else if (center_select == 2) {
219             printf("쿠폰을 등록합니다. 쿠폰 번호를 입력해주세요 : \n");
220             scanf_s("%d", &coupon);
221             if (coupon == coupon_answer) {

```

```

223     if (coupon == coupon_answer) {
224         printf("쿠폰이 확인되었습니다. 한 달 무료 쿠폰이 증명되었습니다.\n");
225     }
226     else if (coupon != coupon_answer) {
227         printf("쿠폰 번호가 틀렸습니다.\n");
228     }
229 }
230
231 }
232
233 }
234
235 }
236
237 break;
238
239
240
241
242
243
244 default:
245     printf("잘못된 선택입니다. 다시 선택하세요.\n");
246 }
247
248 if (end == 1) {
249     printf("재생 목록을 종료합니다.\n");
250     break;
251 }
252 if (songCount[0] == MAX_SONGS) {
253     printf("노래가 %d개로 다 찼습니다.\n", songCount[0]);
254     break;
255 }
256 }
257 return 0;
258
259

```

2) 테스트 결과

(1) 현재 노래수 삭제 > 개개인의 노래수 따로 출력

1 번 사용자에게 노래 두 개를 입력한 뒤 3 > 1 을 입력하여 노래수가 출력 확인.

```

3
사용자를 선택하세요 (1~5): 1
가의 재생목록 (노래 수 : 2) :
1. 종니 (장르: 발라드)
2. 가시 (장르: 발라드)

```

```

-----
메뉴를 입력해주세요.
1. 재생목록에 노래 추가
2. 노래 삭제
3. 재생목록
4. 종료
5. 노래 수정
6. 비슷한 노래 둘러보기
7. 가장 인기있는 장르
8. 고객센터
-----

```

(2)가장 인기있는 장르 출력

사용자 1 에게 발라드 두 곡을 입력하고 3 에게 힙합 한 곡을 입력한 후, 가장 인기있는 장르를 출력하고자 했을 때 발라드가 출력되는지 확인.

```
1
사용자를 선택하세요 (1~5): 1
노래의 장르를 입력하세요 (POP, 발라드, 힙합 등): 발라드
노래를 입력하세요: 가시
'가시' 노래가 '발라드' 장르에 추가되었습니다.

-----

메뉴를 입력해주세요.
1. 재생목록에 노래 추가
2. 노래 삭제
3. 재생목록
4. 종료
5. 노래 수정
6. 비슷한 노래 둘러보기
7. 가장 인기있는 장르
8. 고객센터

1
사용자를 선택하세요 (1~5): 3
노래의 장르를 입력하세요 (POP, 발라드, 힙합 등): 힙합
노래를 입력하세요: 시차
'시차' 노래가 '힙합' 장르에 추가되었습니다.

-----

메뉴를 입력해주세요.
1. 재생목록에 노래 추가
2. 노래 삭제
3. 재생목록
4. 종료
5. 노래 수정
```

```
메뉴를 입력해주세요.
1. 재생목록에 노래 추가
2. 노래 삭제
3. 재생목록
4. 종료
5. 노래 수정
6. 비슷한 노래 둘러보기
7. 가장 인기있는 장르
8. 고객센터

-----

1
사용자를 선택하세요 (1~5): 1
노래의 장르를 입력하세요 (POP, 발라드, 힙합 등): 발라드
노래를 입력하세요: 종니
'종니' 노래가 '발라드' 장르에 추가되었습니다.

-----

메뉴를 입력해주세요.
1. 재생목록에 노래 추가
2. 노래 삭제
3. 재생목록
4. 종료
5. 노래 수정
6. 비슷한 노래 둘러보기
7. 가장 인기있는 장르
8. 고객센터

-----
```

```

메뉴를 입력해주세요.
1. 재생목록에 노래 추가
2. 노래 삭제
3. 재생목록
4. 종료
5. 노래 수정
6. 비슷한 노래 둘러보기
7.가장 인기있는 장르
8.고객센터
-----
7
가장 인기있는 장르: 발라드

```

(3)고객센터

8을 입력했을 때 고객센터가 실행. 2를 입력하면 종료, 1을 입력하면 안내창 띄움

```

-----
메뉴를 입력해주세요.
1. 재생목록에 노래 추가
2. 노래 삭제
3. 재생목록
4. 종료
5. 노래 수정
6. 비슷한 노래 둘러보기
7.가장 인기있는 장르
8.고객센터
-----
8
고객센터에 접속하셨습니다. 도움이 필요하시면 1, 필요 없다면 2를 입력해주세요 : 2
고객센터를 종료합니다

```

8>1을 입력했을 때 이용권 해지한다는 문구 출력 확인

```

메뉴를 입력해주세요.
1. 재생목록에 노래 추가
2. 노래 삭제
3. 재생목록
4. 종료
5. 노래 수정
6. 비슷한 노래 둘러보기
7.가장 인기있는 장르
8.고객센터
-----
8
고객센터에 접속하셨습니다. 도움이 필요하시면 1, 필요 없다면 2를 입력해주세요 : 1
<고객센터 메뉴>
1.해지 신청
2.쿠폰 등록
1
이용권을 해지합니다.
-----

```

8>2를 입력했을 때 쿠폰을 입력하라는 문구 출력. 맞으면 쿠폰 증정, 틀리면 틀렸다는 문구 출력.

```

8
고객센터에 접속하셨습니다. 도움이 필요하시면 1, 필요 없다면 2를 입력해주세요 : 1
<고객센터 메뉴>
1.해지 신청
2.쿠폰 등록
3
쿠폰을 등록합니다. 쿠폰 번호를 입력해주세요 :
1111
쿠폰 번호가 틀렸습니다.
-----

```

```

91      int coupon = 0;
92      int coupon_answer = 231630;
93

```

```

8
고객센터에 접속하셨습니다. 도움이 필요하시면 1, 필요 없다면 2를 입력해주세요
<고객센터 메뉴>
1.해지 신청
2.쿠폰 등록
3
쿠폰을 등록합니다. 쿠폰 번호를 입력해주세요 :
231630
쿠폰이 확인 되었습니다. 한 달 무료 쿠폰이 증정되었습니다.

메뉴를 입력해주세요.
1. 재생목록에 노래 추가
2. 노래 삭제
3. 재생목록
4. 종료
5. 노래 수정
6. 비슷한 노래 둘러보기
7. 가장 인기있는 장르
8. 고객센터
-----

```

4. 계획 대비 변경 사항

1) 변경 내역 제목

- 이전 : 노래별 재생횟수 카운트, 이용시간 나이 입력
- 이후 : 고객센터 추가, 노래별 재생횟수 카운트 삭제, 이용시간 나이 중 하나 삭제
- 사유 : 재생 횟수출력은 의미 없다고 생각. 구조체화, 포인터화에 초점을 맞추기 위해서

(이용시간 입력과 나이를 추가할 경우 코드가 너무 복잡해짐. 이미 배열과 변수가 많아서 너무 복잡한 느낌이어서 적어도 한 개만 진행 할 예정)

5. 프로젝트 일정

업무		11/3	11/4~11/18		11/19~11/26	11/26~12/10	12/11~12/26
제안서 작성		완료					
기능 1	재생목록 입출력, 삭제, 수정		완료				
기능 2	사용자 이름, 장르 입력 및 같은 장르 노래 출력			완료			
기능 3	장르 순위 출력 고객센터				완료		
기능 4	포인터화, 구조체화, 함수 헤더파일 화					진행 예정 -----	