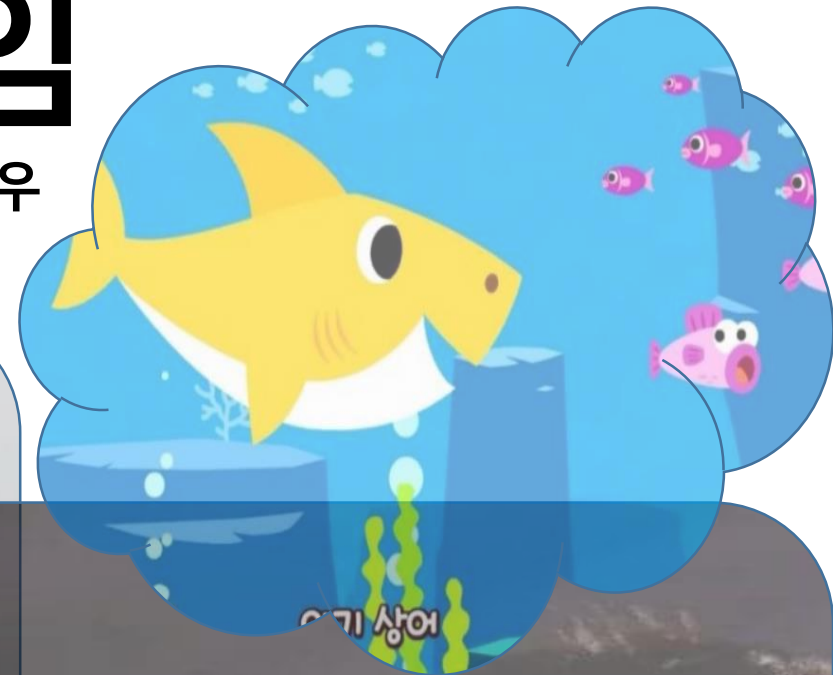


노래 맞추기 게임

201601720 이강우



1. 계획

- 제공되어진 코드 중 옥타브 표와 Beep함수를 이용하고 건반은 이용하지 x

음계	도	C#	레	D#	미	파	F#	솔	G#	라	A#	시
octave	(C)		(D)		(E)	(F)		(G)		(A)		(B)
1	33	35	37	39	41	44	46	49	52	55	58	62
2	65	69	73	78	82	87	92	98	104	110	117	123
3	131	139	147	156	165	175	185	196	208	220	233	247
4	262	277	294	311	330	349	370	392	415	440	466	494
5	523	554	587	622	659	698	740	784	831	880	932	988
6	1047	1109	1175	1245	1319	1397	1480	1568	1661	1760	1865	1976

Beep	함수원형	BOOL WINAPI Beep(__in DWORD dwFreq, __in DWORD dwDuration);	
	함수인자	DWORD dwFreq	소리의 주파수로서 37부터 32,767 사이의 값만을 사용한다.
		DWORD dwDuration	소리의 지속시간을 의미하며 밀리초(milliseconds)로 나타낸다.
	반환 값	정상적으로 실행되면 0이 아닌 값을 반환한다.	

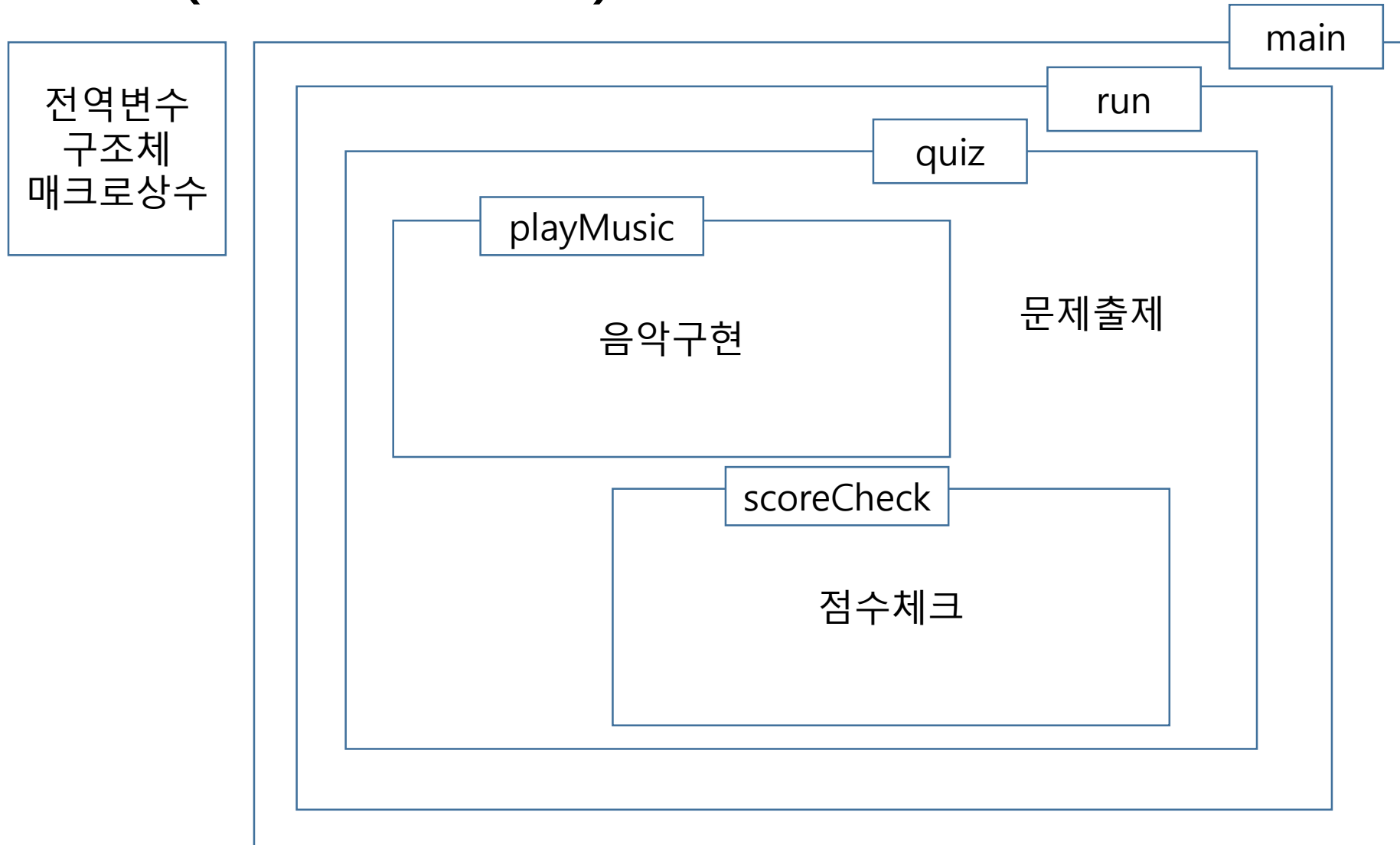
2.요구분석

- 기존 코드의 문제점
- 문제점 1 : 옥타브 음역대 1옥타브 제한 -> 3옥타브
- 문제점 2 : 키보드를 통해 음을 구현 -> 악보로 구현
- 문제점 3 : 재미적 요소 x -> 게임구현

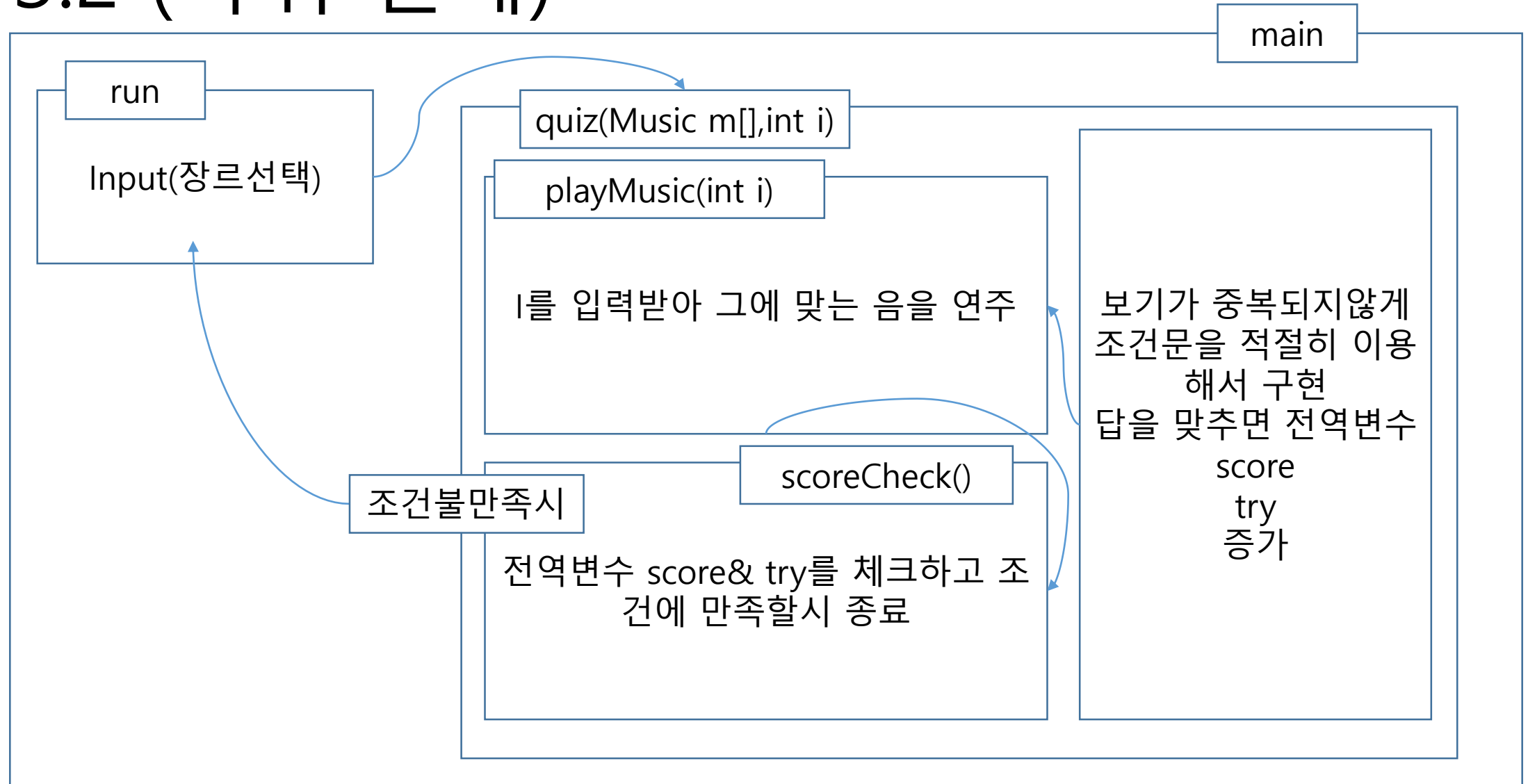
3.설계

- 상위 설계(High Level Design)
- 하위 설계(Low Level Design)

3.1 (상위 설계)



3.2 (하위 설계)



4.구현 (헤더파일)

- <time.h>: 시간을 받아와 랜덤변수생성하기위해
- <stdlib.h>: 랜덤변수를 사용하기위해
- <windows.h>: Beep함수를 사용하기위해
- <conio.h>: getch()함수를 사용하기위해

```
#include <stdio.h>
#include<time.h>
#include<stdlib.h>
#include <windows.h>
#include <conio.h>
```

4.구현 (구조체)

- typedef struct

{

char name[20];

char title[20];

} Music;

음악 구조체 가수이름 & 노래제목

```
typedef struct
{
    char name[20];
    char title[20];
} Music;

Music ballad[3] = { {"민경훈", "가시"}, {"로킴", "봄봄봄"}, {"김범수", "보고싶다"} };
Music child[3] = { {"핑크퐁", "아기상어", }, {"작곡가미상", "나비야"}, {"작곡가미상", "곰세마리"} };
Music troat[3] = { {"홍진영", "사랑의배터리"}, {"조용필", "여행을떠나요"}, {"태진아", "동반자"} };
```


4.구현 (매크로상수+전역변수)

- #define starkato 200 : 반음
- #define note 300 : 한음 1/4박자
- #define note2 600 : 한음*2
- #define note3 900 : 한음*3
- #define one_sol 392 : 1옥타브
- #define two_do 523 : 2옥타브
- #define three_do 1047 : 3옥타브
- Static int score : (전역변수) 게임의 점수
- Static int try : (전역변수) 게임시도횟수

```
static int score = 0;  
static int try = 0;
```

```
#define starkato 200  
#define note 300  
#define note2 600  
#define note3 900  
#define one_sol 392  
#define one_la 440  
#define one_si 494  
#define two_do 523  
#define two_dos 554  
#define two_rae 587  
#define two_raes 622  
#define two_mi 659  
#define two_fa 698  
#define two_fas 740  
#define two_sol 784  
#define two_sols 831  
#define two_la 880  
#define two_las 932  
#define two_si 988  
#define three_do 1047  
#define three_dos 1109  
#define three_rae 1175
```

4.구현 (PlayMusic(index))

- Void PlayMusic(int index){
If(index==0){1번노래재생 }
else if (index==1){2번노래재생}...
}

노래 재생함수

0~2 발라드/3~5 동요/6~8 트로트

매크로 상수를 이용해 구현

```
void playMusic(int index) {  
    if (index == 0) { //버즈 가시  
        Beep(two_si, note); //그  
        Sleep(starkato);  
        Beep(two_si, starkato); //대  
        Beep(three_mi, starkato); //기  
        Beep(three_fas, starkato); //억  
        Beep(three_sols, note3); //이  
        Sleep(note);  
        Beep(two_si, starkato); //지  
        Sleep(note);  
        Beep(two_si, starkato); //난  
        Beep(three_mi, starkato); //사  
        Beep(three_fas, starkato); //랑  
        Beep(three_sols, note3); //이  
        Sleep(note);  
        Beep(three_fa, starkato); //내  
        Beep(three_fa, starkato); //안  
        Beep(three_mi, starkato); //을  
        Sleep(note);  
        Beep(three_rae, starkato); //파
```

4.구현 (run)

- 실질적인 프로그램이 실행되는 함수

Quiz함수에 구조체배열과 선택장르를 매개변수로 넘겨줌

```
void run() {  
    int input;  
    printf("----- 「노래맞추기 게임ver1.0」 -----\\n");  
    printf("    Press any Button to Start    ");  
    _getch();  
    printf("\\n");  
    while (TRUE) {  
        printf("장르를 선택하세요. 1:발라드(30점) 2:동요(10점) 3:트로트(20점)\\n>>");  
        scanf_s("%d", &input);  
  
        if (input == 1) {  
            Quiz(ballad, input);  
        }  
        else if (input == 2) {  
            Quiz(child, input);  
        }  
        else if (input == 3) {  
            Quiz(troat, input);  
        }  
  
        else {  
            printf("잘못 입력하셨습니다\\n");  
        }  
    }  
}
```

4.구현 (quiz)

- 문제를 내는 함수

랜덤으로 index 0~2 생성

input값에 따라 index에 일정값 +
3초후 playmusic() 실행

Ansplace = 보기 3곳중 정답의 위치

One or two = 정답을 제외한 오답
의 위치를 결정해주는 변수

```
void Quiz(Music m[], int input) {
    srand(time(NULL));
    int index = rand() % 3;
    int ansplace;
    int oneortwo;
    int answer;
    if (input == 1) {
        printf("발라드 장르를 선택하셨습니다.3초후에 노래가 시작됩니다.\n");
    }
    else if (input == 2) {
        printf("동요 장르를 선택하셨습니다.3초후에 노래가 시작됩니다.\n");
    }
    else if (input == 3) {
        printf("트로트 장르를 선택하셨습니다.3초후에 노래가 시작됩니다.\n");
    }
    printf("3\n");
    Sleep(1000);
    printf("2\n");
    Sleep(1000);
    printf("1\n");
    Sleep(1000);
    printf("♪ Music Playing ♪\n");
    if (input == 1) {
        playMusic(index);
    }
    else if (input == 2) {
        playMusic(index + 3);
    }
    else if (input == 3) {
        playMusic(index + 6);
    }
    ansplace = rand() % 3 + 1;
    oneortwo = rand() % 2 + 1;
    printf("보기: ");
    if ((ansplace == 1) && (index == 0)) { ... }
    else if ((ansplace == 1) && (index == 1)) { ... }
    else if ((ansplace == 1) && (index == 2)) { ... }
    else if ((ansplace == 2) && (index == 0)) { ... }
    else if ((ansplace == 2) && (index == 1)) { ... }
    else if ((ansplace == 2) && (index == 2)) { ... }
    else if ((ansplace == 3) && (index == 0)) { ... }
    else if ((ansplace == 3) && (index == 1)) { ... }
    else if ((ansplace == 3) && (index == 2)) { ... }
```

4.구현 (quiz~)

- Ansplace와 index (케이스별 분류)
- Oneortwo (오답 보기 위치)

매개변수로 받은 구조체멤버에
접근->가수+노래제목

```
if ((ansplace == 1) && (index == 0)) {
    printf("[1] %s %s ", m[index].name, m[index].title);
    if (oneortwo == 1) {
        printf("[2] %s %s ", m[index + 1].name, m[index + 1].title);
        printf("[3] %s %s \n", m[index + 2].name, m[index + 2].title);
    }
    else if (oneortwo == 2) {
        printf("[2] %s %s ", m[index + 2].name, m[index + 2].title);
        printf("[3] %s %s \n", m[index + 1].name, m[index + 1].title);
    }
}

else if ((ansplace == 3) && (index == 2)) {
    if (oneortwo == 1) {
        printf("[1] %s %s ", m[index - 2].name, m[index - 2].title);
        printf("[2] %s %s ", m[index - 1].name, m[index - 1].title);
        printf("[3] %s %s \n", m[index].name, m[index].title);
    }
    else if (oneortwo == 2) {
        printf("[1] %s %s ", m[index - 1].name, m[index - 1].title);
        printf("[2] %s %s ", m[index - 2].name, m[index - 2].title);
        printf("[3] %s %s \n", m[index].name, m[index].title);
    }
}
```

4.구현 (quiz~)

- Answer 와 ansplace가 일치
(장르에 해당하는 점수획득)
(시도횟수 증가)

- 다를시
(시도횟수 증가)

ScoreCheck함수로 이동

```
printf("Answer>>");
scanf_s("%d", &answer);

if ((answer == ansplace)&&(input==1)) {
    printf("Correct\n");
    try++;
    score += 30;
}
else if ((answer == ansplace) && (input == 2)) {
    printf("Correct\n");
    try++;
    score += 10;
}
else if ((answer == ansplace) && (input == 3)) {
    printf("Correct\n");
    try++;
    score += 20;
}
else {
    printf("Wrong\n");
    try++;
}
scoreCheck();
}
```

4.구현 (scoreCheck)

- 점수 체크 함수

전역변수 score가 100점이넘으면
성공메시지 출력후 종료
시도횟수가 10번이되면
실패메시지 출력후 종료

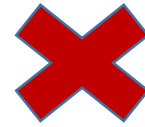
```
void scoreCheck() {  
    printf("점수 : %d\n", score);  
    printf("시도횟수 : %d\n", try);  
    if (score >= 100) {  
        printf("You Win\n");  
        printf("최종시도횟수 %d번\n", try);  
        system("pause");  
        exit(1);  
    }  
    else if (try >= 10) {  
        printf("You Lose\n");  
        printf("최대시도횟수 %d번이 되었습니다.\n", try);  
        system("pause");  
        exit(1);  
    }  
}
```

5.테스트

```
----- 「노래맞추기 게임ver1.0」 -----  
Press any Button to Start  
장르를 선택하세요. 1:발라드(30점) 2:동요(10점) 3:트로트(20점)  
>>2  
동요 장르를 선택하셨습니다.3초후에 노래가 시작됩니다.  
3  
2  
1  
♪Music Playing♪  
보기: [1] 핑크퐁 아기상어 [2] 작곡가미상 나비야 [3] 작곡가미상 곶세마리  
Answer>>
```



```
♪Music Playing♪  
보기: [1] 핑크퐁 아기상어 [2] 작곡가미상 나비야 [3] 작곡가미상 곶세마리  
Answer>>3  
Correct  
점수 : 100  
시도횟수 : 4  
You Win  
최종시도횟수 4번  
계속하려면 아무 키나 누르십시오 . . .
```



```
♪Music Playing♪  
보기: [1] 민경훈 가시 [2] 로x김 봄봄봄 [3] 김범수 보고싶다  
Answer>>1  
Wrong  
점수 : 60  
시도횟수 : 10  
You Lose  
최대시도횟수 10번이 되었습니다.  
계속하려면 아무 키나 누르십시오 . . .
```


6.유지보수

- 프로그램을 구현하면서 보기를 구현하는데 있어 한계점..
- 악보 구현시 노가다x ->파일스트림
- 비프음이 생각보다 구별하기 쉽지않음