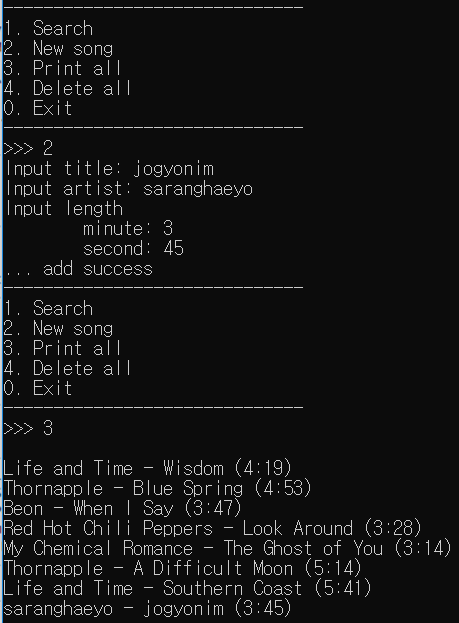
**소프트웨어 입문 설계 일곱 번째 과제**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 3번: 전체 음악 출력 -> 2번: 새 음악 추가 -> 3번: 전체 음악 출력 -> 1번: 음악 찾기 -> 1-2번: 음악 삭제 -> 3번: 전체 음악 출력 -> 1번: 음악 찾기 -> 1-1번: 내용 수정 -> 4번: 전체음악 삭제 -> 3번: 전체 음악 출력 -> 0번: 저장 후 종료 -> 1-0번: 이전 -> 새 음악 추가할 때 오류들 | | |

2019042497 송정명

음악 리스트를 담기 위해 음악의 길이(분, 초)를 담는 구조체를 선언하고, 음악의 정보(제목, 가수, 음악의 길이)를 담는 구조체를 선언하고, 최종적으로 음악 리스트(음악들의 정보, 음악의 수, 배열의 크기)를 담는 구조체를 선언했습니다.

SongList dataLoad(): 파일 안의 데이터를 읽어오는 함수입니다.

int dataSave(SongList): 구조체 안에 있는 음악 리스트를 파일 안에 덮어쓰며 저장하기 위한 함수입니다.

void printMenu(): 메뉴를 출력하는 함수입니다.

void printSong(Song\*): 선택한 음악의 정보를 출력하는 함수입니다.

void printSongMenu(): 메뉴의 1번 안의 음악 수정, 삭제, 이전 메뉴를 출력하는 함수입니다.

Song\* searchSong(SongList, char\*, char\*): 음악 리스트 안에 음악이 있는지 찾는 함수입니다.

int editSong(Song\*): 선택한 음악의 정보를 수정하는 함수입니다.

int deleteSong(SongList\*, Song\*): 음악 리스트에서 선택한 음악을 지우는 함수입니다.

int addSong(SongList\*): 새로운 음악을 음악 리스트에 추가하는 함수입니다.

void printAllSongs(SongList): 음악 리스트에 있는 음악들을 출력하는 함수입니다.

int deleteAll(SongList\*): 음악 리스트에 있는 음악들을 모두 삭제하는 함수입니다.

메뉴에서 scanf 오입력을 예외처리하기 while (scanf(" %d", &num) != 1)를 씌워서 변수에 몇 번을 고를 것인지 받으면서 문자 입력 받는 것을 걸러주었으며, while ((ch = getchar()) != '\n');를 사용해서 버퍼에 남아있을 문자를 다 빼주었습니다.

또한, 혹시나 버퍼에 개행(‘\n’)이 남아있을 수 있는 문제를 해결하기 위해서 scanf(“ %s”,title); 이런 느낌으로 입력 받는 함수마다 형식지정자 앞에 띄어쓰기를 모두 해주었습니다.

파일을 처음 열 때 읽기 모드(‘r’)으로 열었고, 만약에 열린 파일이 없을 경우를 대비해 추가 모드(‘a’)로 파일을 생성해주었습니다.

음악 리스트에 음악들을 추가하기 위해 음악 정보를 담는 구조체 부분을 동적할당해 음악들을 추가했습니다.

프로그램 종료 하기 전에 동적할당 했던 songlist.songs의 메모리를 정리해줬습니다.