

# CSED101. Programming & Problem solving Spring, 2020

## Programming Assignment #4 (75 points)

오재영 (jaeyoungoh@postech.ac.kr)

■ 제출 마감일: 2021.06.02 23:59

■ 개발 환경: Windows Visual Studio 2019

### ■ 제출물

- C 소스 코드 (assn4.c)
  - 프로그램의 소스 코드를 이해하기 쉽도록 반드시 주석을 붙일 것.
- 보고서 파일 (assn4.docx, assn4.hwp 또는 assn4.pdf)
  - AssnReadMe.pdf 를 참조하여 작성할 것.
  - 명예서약(Honor code): 표지에 다음의 내용을 포함한다. “ 나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다. ” 보고서 표지에 명예 서약이 없는 경우는 과제를 제출하지 않은 것으로 처리한다.
  - 소스코드와 보고서 파일을 PLMS를 이용하여 제출한다.

### ■ 주의사항

- 각 문제에 해당하는 요구사항을 반드시 지킬 것.
- 모든 문제의 출력 형식은 채점을 위해 아래에 제시된 예시들과 최대한 비슷하게 작성해 주세요.
- 각 문제에 제시되어 있는 파일이름으로 제출 할 것. 그 외의 다른 이름으로 제출하면 감점 또는 0점 처리된다.
- 컴파일 & 실행이 안되면 무조건 0점 처리된다.
- 하루 late시 20%가 감점되며, 3일 이상 지나면 받지 않는다. (0점 처리)
- 부정행위에 관한 규정은 POSTECH 전자컴퓨터공학부 학부위원회의 ‘ POSTECH 전자컴퓨터공학부 부정행위 정의 ’ 를 따른다. (PLMS의 본 과목 공지사항에 등록된 글 중, 제목이 [document about cheating]인 글에 첨부되어 있는 disciplinary.pdf를 참조하세요.)
- 이번 과제에서는 추가 기능 구현과 관련된 추가 점수가 따로 없습니다.

## ■ Problem: 음악 정보 관리 프로그램

### [요약]

플레이리스트란 원하는 음악들을 모아 듣고 싶은 순서대로 저장한 것을 말한다. 이번 과제에서는 음악을 재생하지는 않지만, 플레이리스트와 같이 음악의 정보를 저장하고 관리하는 것을 목적으로 한다.

### [개요]

주요 기능은 아래와 같다.

1. 텍스트 파일로 저장된 음악 정보를 읽고 플레이리스트에 저장
2. 플레이리스트 내의 음악 정보를 출력
3. 플레이리스트 내의 음악 정보를 검색
4. 플레이리스트에 음악 정보를 추가
5. 플레이리스트 내의 음악 정보를 삭제
6. 플레이리스트 내의 음악 정보의 순서를 이동
7. 프로그램 종료

### [목적]

과제의 목적은 아래와 같다.

1. 구조체(struct)의 사용법을 익힌다.
2. 연결리스트(linked list)의 사용법을 익힌다.

### [주의사항]

1. 이번 과제는 총 6 개의 명령어("show", "search", "add", "delete", "move", "quit")를 입력 받아 각 기능을 수행한다. 적어도 각 명령어 별로 함수를 정의하여 사용해야 한다. 명령어 외에 필요한 함수는 정의하여 사용할 수 있다. 기능적으로 독립됐거나 반복적으로 사용되는 기능은 사용자 함수를 정의해서 구현하도록 한다.
2. 이번 과제는 구조체와 연결리스트를 활용하는 것이 목표이므로, 문제에 구조체와 연결리스트를 언급한 부분(플레이리스트의 구현)을 배열을 이용하여 해결 시 감점 처리 된다.
3. 붉은 글씨로 명시한 예러 처리 외에는 고려할 필요가 없다.  
붉은 글씨로 명시된 예외처리는, [필수]가 표시된 것은 점수가 포함된 것으로 반드시 구현하고, [가이드라인] 표시된 것은 점수에는 포함되지 않으나 구현에 참고하도록 한다.
4. 프로그램의 입력 및 출력 형식은 채점을 위해 아래의 실행 예시와 일맥상통하게 한다.
  - 비슷하거나 형식이 이해되는 수준은 허용된다.
5. string.h 에서 제공하는 라이브러리 함수를 사용할 수 있다.
6. 전역 변수와 goto 문은 사용할 수 없다.

### [설명 및 요구사항]

- 프로그램 구현은 구조체와 연결리스트를 사용하여야 하며, 각 항목에 대한 제약은 다음과 같다.
  - 플레이리스트: 음악의 제목, 아티스트, 재생시간, 재생된 횟수 등의 정보를 가진 구조체를 정의하고 연결리스트로 구현한다.
    1. 음악의 제목: 제목은 띄어쓰기를 포함하는 45 자 이내의 문자열.
    2. 아티스트: 아티스트는 띄어쓰기를 포함하는 45 자 이내의 문자열.
    3. 재생시간: 음악의 재생시간(초). 1000 보다 작은 자연수.
    4. 재생횟수: 음악이 재생된 횟수. 2,000,000,000 보다 작은 자연수.

### [프로그램 시작 시]

1. 프로그램을 실행하면 아래와 같이 읽어올 파일 이름을 입력 받는다. 파일이름에는 공백이 없으며 확장자를 포함하여 30 자를 넘지 않는다고 가정한다.

기존의 플레이리스트를 저장한 파일 이름을 입력해주세요. >>

- 예외처리[필수]:

만약 존재하지 않는 파일의 이름을 받은 경우, 아래와 같은 에러 메시지를 출력 후 다시 파일 이름을 입력 받는다. (실행 예시의 밑줄은 사용자 입력에 해당)

기존의 플레이리스트를 저장한 파일 이름을 입력해주세요. >> wrong.txt

유효하지 않은 입력입니다. 기존의 플레이리스트를 저장한 파일 이름을 입력해주세요. >> music.txt

2. 파일을 열어 위에서부터 읽으며 연결리스트(이하 플레이리스트)에 삽입하되 최종적으로 플레이리스트 내의 음악들의 순서가 재생된 횟수의 내림차순이 되도록 한다. 재생된 횟수가 동일한 경우 순서는 상관없다. 주어진 파일 music.txt의 내용과 구성은 아래와 같다.

Mood	24kGoldn ft. iann dior	150	220472847
WAP	Cardi B	252	395732843
Dynamite	BTS	223	1067721437
Savage Love	BTS	185	147251721

- 한 줄에 하나의 음악 정보가 들어가며, 각 음악 정보는 앞에서부터 차례로 음악의 제목, 아티스트, 재생시간, 재생 횟수로 구성된다. 제목, 아티스트, 재생시간, 재생된 횟수 사이는 tab 키로 구분되어 있다.
- 음악의 제목과 아티스트 정보는 띄어쓰기(space)를 포함하여 45 자 이내의 문자열이며, 재생시간은 1000 보다 작은 자연수이다. 재생 횟수는 2,000,000,000 보다 작은 자연수이다. 이외의 입력에 대해서는 고려하지 않는다.

3. 플레이리스트에서 상위 3 개의 음악들의 정보를 출력한다. 이후 **[플레이리스트 관리]**를 시작한다.

기존의 플레이리스트를 저장한 파일 이름을 입력해주세요. >> wrong.txt  
유효하지 않은 입력입니다. 기존의 플레이리스트를 저장한 파일 이름을 입력해주세요. >> music.txt

재생 횟수 Top 3 음악의 정보입니다.

	title	artist	time	# of plays
1	Dynamite	BTS	223	1067721437
2	WAP	Cardi B	252	395732843
3	Mood	24kGoldn ft. iann dior	150	220472847

[total: 3 music, 625 (s)]

명령어를 입력해주세요. >>

### [플레이리스트 관리]

- 플레이리스트 관리를 위해서는 "show", "search", "add", "delete", "move", "quit" 로 총 6 개의 명령어가 필요하다. "quit" 명령어를 입력 받기 전까지는 계속해서 명령어를 입력 받고 그에 따른 작업을 수행한다.
1. 플레이리스트를 파일로부터 성공적으로 읽어서 작성을 하게 되면, 그 다음으로 아래와 같이 실행할 명령어를 입력 받는다.

명령어를 입력해주세요. >>

- **예외처리[필수]:**

만약 위의 6 개 외의 명령어가 아닌 입력을 받은 경우, 아래와 같은 에러 메시지를 출력 후 다시 명령어를 입력 받는다.

명령어를 입력해주세요. >> sohwww  
유효하지 않은 입력입니다. 명령어를 입력해주세요. >>

2. 각 명령어에 따른 작업을 수행한다.

#### (1) "show" 명령어: 플레이리스트의 내용 출력

- 플레이리스트 내의 모든 음악의 정보를 순서대로, 순서와 함께 출력한다.
- 맨 아래에는 **음악의 총 개수와 총 재생시간**을 출력한다.

명령어를 입력해주세요. >> show

	title	artist	time	# of plays
1	Dynamite	BTS	223	1067721437
2	WAP	Cardi B	252	395732843
3	Mood	24kGoldn ft. iann dior	150	220472847
4	Savage Love	BTS	185	147251721

[total: 4 music, 810 (s)]

명령어를 입력해주세요. >>

- 예외처리[가이드라인]: 플레이리스트에 음악이 존재하지 않을 경우에, 아래와 같이 틀만 출력하도록 한다.

명령어를 입력해주세요. >> show

```
=====
title                artist                time                # of plays
=====
[total: 0 music, 0 (s)]
```

명령어를 입력해주세요. >>

(2) "search" 명령어: 플레이리스트의 특정 아티스트의 음악 정보를 출력

- 어떤 아티스트의 음악 정보를 출력할지 입력 받는다.
- 플레이리스트 내의 해당 아티스트의 모든 음악의 정보를 **플레이리스트 내에서의 순서대로, 플레이리스트에서의 원 순서**와 함께 출력한다.
- 맨 아래에는 **출력된 음악의 총 개수와 총 재생시간**을 출력한다.

명령어를 입력해주세요. >> search

어떤 아티스트의 음악을 검색할까요? >> BTS

```
=====
title                artist                time                # of plays
=====
1      Dynamite      BTS                  223                1067721437
4      Savage Love   BTS                  185                147251721
=====
[total: 2 music, 408 (s)]
```

명령어를 입력해주세요. >>

- 예외처리[가이드라인]: 플레이리스트에 입력한 아티스트의 음악이 존재하지 않을 경우에, 아래와 같이 출력하고 이번 작업을 마친다.

명령어를 입력해주세요. >> search

어떤 아티스트의 음악을 검색할까요? >> BTTTSS

플레이리스트에 해당 아티스트의 음악이 없습니다.

명령어를 입력해주세요. >>

(3) "add" 명령어: 플레이리스트의 특정 순서에 음악을 추가

- 주의! 음악을 플레이리스트의 1) 맨 앞, 2) 맨 뒤, 또는 3) 중간에 추가하는 세가지 경우가 모두 동작하는지 확인할 것.
- 몇 번째 순서로 음악을 추가할지 입력 받는다.
  - 예외처리[가이드라인]: 1 보다 작거나 (현재 플레이리스트의 총 음악 개수 + 1) 보다 큰 수가 입력된 경우에, 아래와 같은 에러메시지를 출력 후 다시 입력 받는다.
  - add 명령어뿐 아니라, [플레이리스트 관리] 전체에서 음악의 순서로 입력 받는 모든 내용은 0 이상 10000 이하의 정수뿐이다. 이외의 경우는 고려할 필요가 없다.

```
명령어를 입력해주세요. >> add
몇 번째 순서로 음악을 추가할까요? >> 0
유효하지 않은 입력입니다. 몇 번째 순서로 음악을 추가할까요? >>
```

- 추가할 음악의 제목, 아티스트, 재생시간, 재생 횟수를 하나씩 아래의 예시처럼 입력 받는다.
- 음악의 제목, 아티스트 정보는 띄어쓰기를 포함하여 45 자 이내인 문자열이며, 재생시간은 1000 보다 작은 자연수, 재생 횟수는 2,000,000,000 보다 작은 자연수만이 입력된다. 이외의 입력 값에 대해서는 고려하지 않는다.
- 이미 플레이리스트에 존재하는 음악과 똑같은 음악을 추가하는 경우는 고려하지 않는다.
- 입력받은 정보대로 음악을 플레이리스트에 추가한다.

```
명령어를 입력해주세요. >> add
몇 번째 순서로 음악을 추가할까요? >> 5
추가할 음악의 제목을 입력해주세요. >> Laugh Now Cry Later
추가할 음악의 아티스트를 입력해주세요. >> Drake Ft. Lil Drunk
추가할 음악의 재생 시간을 입력해주세요. >> 301
추가할 음악의 재생 횟수를 입력해주세요. >> 286098956
```

```
명령어를 입력해주세요. >> show
```

	title	artist	time	# of plays
1	Dynamite	BTS	223	1067721437
2	WAP	Cardi B	252	395732843
3	Mood	24kGoldn ft. iann dior	150	220472847
4	Savage Love	BTS	185	147251721
5	Laugh Now Cry Later	Drake Ft. Lil Druk	301	286098956

```
[total: 5 music, 1111 (s)]
```

```
명령어를 입력해주세요. >>
```

(4) "delete" 명령어: 플레이리스트 내의 특정 순서의 음악을 삭제

- 주의! 플레이리스트의 1) 맨 앞, 2) 맨 뒤, 3) 중간에 있는 음악을 삭제하는 세가지 경우가 모두 동작하는지 확인할 것.
- 몇 번째 순서의 음악을 삭제할지 입력 받는다.
  - 예외처리[가이드라인]: 1 보다 작거나 (현재 플레이리스트의 총 음악 개수) 보다 큰 수가 입력된 경우에, 아래와 같은 에러메시지를 출력 후 다시 입력 받는다.

```
명령어를 입력해주세요. >> delete
몇 번째 순서의 음악을 삭제할까요? >> 1
유효하지 않은 입력입니다. 몇 번째 순서의 음악을 삭제할까요? >>
```

- 입력 받은 순서의 음악을 플레이리스트에서 삭제한다. 삭제 시에는 해당 음악 정보의 저장을 위해 동적 할당 받은 메모리를 할당 해제(free) 한다.

```
명령어를 입력해주세요. >> delete
몇 번째 순서의 음악을 삭제할까요? >> 3
```

	title	artist	time	# of plays
3	Mood	24kGoldn ft. iann dior	150	220472847

```
[total: 1 music, 150 (s)]
위 음악이 삭제되었습니다.
```

```
명령어를 입력해주세요. >> show
```

	title	artist	time	# of plays
1	Dynamite	BTS	223	1067721437
2	WAP	Cardi B	252	395732843
3	Savage Love	BTS	185	147251721
4	Laugh Now Cry Later	Drake Ft. Lil Druk	301	286098956

```
[total: 4 music, 961 (s)]
```

```
명령어를 입력해주세요. >>
```

- 예외처리[가이드라인]: 만약 플레이리스트에 음악이 존재하지 않는 경우, 아래와 같이 출력하고 이번 작업을 마친다.

```
명령어를 입력해주세요. >> show
```

	title	artist	time	# of plays
--	-------	--------	------	------------

```
[total: 0 music, 0 (s)]
```

```
명령어를 입력해주세요. >> delete
플레이리스트에 삭제할 음악이 없습니다.
```

```
명령어를 입력해주세요. >>
```

(5) "move" 명령어: 플레이리스트 내의 특정 음악의 순서를 변경

- 주의! Swap 이 아님. 특정 음악의 순서가 변경됨에 따라 다른 음악의 순서가 밀리며 이동.
- 주의! add, delete 와 마찬가지로 여러가지 경우에 대해서 모두 잘 동작하는지 확인할 것.
- 몇 번째 순서의 음악을 옮길지 입력 받는다.
  - 예외처리[가이드라인]: 1 보다 작거나 (현재 플레이리스트의 총 음악 개수) 보다 큰 수가 입력된 경우에, 아래와 같은 에러메시지를 출력 후 다시 입력 받는다.

```
명령어를 입력해주세요. >> move
몇 번째 순서의 음악의 위치를 바꿀까요? >> 105
유효하지 않은 입력입니다. 몇 번째 순서의 음악의 위치를 바꿀까요? >>
```

- 선택한 음악을 몇 번째 순서로 옮길지 입력 받는다.
  - 예외처리[가이드라인]: 1 보다 작거나 (현재 플레이리스트의 총 음악 개수) 보다 큰 수이거나, 원래 순서와 같은 수가 입력된 경우에, 아래와 같은 에러메시지를 출력 후 다시 입력 받는다.

```
명령어를 입력해주세요. >> move
몇 번째 순서의 음악의 위치를 바꿀까요? >> 4
몇 번째 순서로 이동시킬까요? >> 0
유효하지 않은 입력입니다. 몇 번째 순서로 이동시킬까요? >>
```

- 입력받은 순서대로 플레이리스트 내용을 수정한다.

```
명령어를 입력해주세요. >> move
몇 번째 순서의 음악의 위치를 바꿀까요? >> 4
몇 번째 순서로 이동시킬까요? >> 2
```

```
명령어를 입력해주세요. >> show
```

	title	artist	time	# of plays
1	Dynamite	BTS	223	1067721437
2	Laugh Now Cry Later	Drake Ft. Lil Druk	301	286098956
3	WAP	Cardi B	252	395732843
4	Savage Love	BTS	185	147251721

```
[total: 4 music, 961 (s)]
```

```
명령어를 입력해주세요. >>
```

- 예외처리[가이드라인]:  
만약 플레이리스트에 음악이 1 곡 이하로 존재하는 경우, 아래와 같이 출력하고 이번 작업을 마친다.

```
명령어를 입력해주세요. >> show
```

	title	artist	time	# of plays
1	Dynamite	BTS	223	1067721437



```
[total: 1 music, 223 (s)]
```

```
명령어를 입력해주세요. >> move  
플레이리스트에 순서를 이동할 음악이 없습니다.
```

```
명령어를 입력해주세요. >>
```

#### (6) "quit" 명령어: 프로그램 종료

- 현재까지 편집한 내용을 파일명을 입력 받아서 저장하도록 한다. 앞서 본 "music.txt"와 같은 형식으로 내용을 구성한다.
- 파일 이름은 확장자를 포함하여 최대길이가 30 자이며, 파일 이름에 공백이 없다고 가정한다.
- 종료 시, 동적 할당 받은 memory 를 모두 free 시킨 후 종료한다.

```
명령어를 입력해주세요. >> show
```

	title	artist	time	# of plays
1	Dynamite	BTS	223	1067721437
2	Laugh Now Cry Later	Drake Ft. Lil Druk	301	286098956
3	WAP	Cardi B	252	395732843
4	Savage Love	BTS	185	147251721

```
[total: 4 music, 961 (s)]
```

```
명령어를 입력해주세요. >> quit  
저장할 파일이름을 입력해주세요. >> new_music.txt
```

```
계속하려면 아무 키나 누르십시오. . .
```

- 아래는 생성된 new\_music.txt 파일의 내용이다.

Dynamite	BTS	223	1067721437
Laugh Now Cry Later	Drake Ft. Lil Druk	301	286098956
WAP	Cardi B	252	395732843
Savage Love	BTS	185	147251721

- 예외처리[가이드라인]:

만약 플레이리스트가 비었을 경우, 아래와 같이 출력한다. 이 경우에는 내용을 파일로 따로 저장할 필요는 없다.

```
명령어를 입력해주세요. >> show
```

	title	artist	time	# of plays
--	-------	--------	------	------------

```
[total: 0 music, 0 (s)]
```

```
명령어를 입력해주세요. >> quit  
플레이리스트에 저장할 내용이 없습니다.
```

```
계속하려면 아무 키나 누르십시오. . .
```