

프로젝트 개요

기본적으로 C언어에서 배운 자료형, 함수, 포인터, 파일입출력 등에 대한 자세한 내용을 이해하고 구현하는 능력을 평가하고자 한다. 아래의 문제를 프로그램으로 구현하고 그 결과를 보고서로 제출한다.

1. 프로젝트 주제

DG레코드는 시내에서 음반을 판매하고 있는 작은 매장으로 음반 제작사로부터 새로운 음반을 입고 하여 소비자에게 판매하고 있다. DG레코드는 음반 판매 및 재고를 관리할 수 있는 소프트웨어를 원하고 있다. 소프트웨어는 다음의 기능이 구현되어야 한다.

음반 판매시에는 거래 내역을 기록하고 판매수량에 해당하는 판매액을 계산하고 출력한다.

판매 또는 반품처리 할 경우에 재고 목록에서 현재의 재고량을 수정한다.

음반을 입고, 판매, 검색에 따라 재고 목록에 음반을 추가, 삭제, 수정하거나, 검색할 수 있어야 한다. 다양한 보고서들을 출력해야 한다.

[처리 방법과 기준]

새로운 음반 음반을 판매점에서 등록할 경우 고유한 분류체계를 이용하여 구분하지만 문제를 간단히 하기 위해 분류번호와 장르별 음반의 구분은 제외하고, 다음과 같은 항목으로만 처리한다고 가정한다.

음반 분류 항목: 음반 제목, 가수 또는 그룹, 발매년도, 가격, 재고수량 그리고 현재의 재고상태는 아래의 재고현황 표와 같다고 가정하고 이를 다음과 같이 텍스트파일(albumdata.txt)로 만들어 재고관리 파일로 사용한다. 또한 거래가 발생할 때마다 해당 거래내역을 아래의 표와 같이 기록하기 위한 텍스트파일(transaction.txt)도 작성하여 사용한다.

<<각 메뉴의 기능>>

1. 입고

- 현재의 재고현황을 재고현황 표와 같은 형식으로 출력한다.
- 재고가 있는 음반에 대해서는 입고된 음반의 번호와 수량을 입력하고 재고수량을 가산한다.
- 음반 재고 파일을 검색하여 신규 입고에 대해서는 현 데이터의 마지막에 추가하는 것으로 한다.
- **화면출력:** 입고 처리를 완료하면 음반번호, 음반명, 가수, 년도, 단가, 수량의 정보가 추가되었음을 출력한다.
- 입고할 항목을 입력하는 화면을 정확히 설계 후 기능을 구현한다.
- 입력화면 출력 예)

2. 판매

- 먼저 재고현황을 앞의 재고현황 표와 같은 형식으로 출력한다.
- 판매할 음반의 번호와 수량을 입력한다.
- 해당 제품의 재고수량을 감산한다. (해당 음반이 모두 팔린 경우도 파일에서 삭제하지 않는다)
- **화면출력:** 판매 내역으로 음반번호, 음반명, 가수, 년도, 단가, 수량, 판매액을 출력한다.
- 거래 내역 파일에 새로운 거래내역을 추가한다.
- 거래내역의 거래총액은 기존의 거래액에 새로운 거래액을 누적하고, 누적 부가세는 10%로 계산한다.

3. 반품
 - 먼저 재고현황을 앞의 재고현황 표와 같은 형식으로 출력한다.
 - 반품된 음반의 번호와 수량을 입력한다.
 - 해당 제품의 재고수량에 감산한다.
 - **화면출력:** 반품 내역으로 음반번호, 음반명, 가수, 년도, 단가, 반환액을 화면에 출력한다.
 - 거래 내역 파일에 새로운 거래내역을 추가한다.
 - 거래내역의 거래총액은 기존의 거래액에 새로운 거래액을 누적하고, 누적 부가세는 10%로 계산한다.
4. 출력
 - 출력을 위한 옵션을 선택할 수 있게 구현한다.
 - **화면출력:** 현재의 재고현황을 앞의 재고현황 표와 같은 형식으로 출력한다.
 - **화면출력:** 현재의 거래내역을 앞의 거래내역 표와 같은 형식으로 출력한다.
5. 종료
 - 종료 직전에 변경된 재고수량의 정보를 파일에 저장한다.

2. 처리 및 제한조건

1. 입력 - 재고관리 파일(albumdata.txt)과 기존 거래내역 파일(transaction.txt)을 읽어 들임
 - ① 읽어들인 파일 내용을 위한 자료구조에는 구조체, 배열, 포인터 등을 활용한다.
2. 처리 - 선택된 메뉴에 의해 재고관리와 거래내역을 처리 함
 - ① 각 메뉴의 처리를 위해서는 switch문과 열거형 상수를 이용한다.
 - ② 각 메뉴의 처리는 서로 다른 함수로 구현한다.
 - ③ 각 메뉴의 처리 현황을 화면에 출력한다. (요구사항에 맞게 화면 설계가 필요함)
3. 출력 - 현재의 재고내용인 재고관리 파일(albumdata.txt)과 거래내역(transaction.txt)을 화면에 출력

[[거래내역서]]

거래일자	앨범명	가수	유형	단가	수량	거래액	거래총액	누적부가세
14.10.01	METEORA	Linkin Park	판매	19000	2	38400	38400	3840
14.10.02	I'M DIFFERENT	HI SUHYUN	판매	18700	1	18700	57100	5710
14.10.05	Coast to Coast	WESTLIFE	판매	23000	1	23000	80100	8010
14.10.14	SHINE/RIDE ON	동방신기	판매	14400	2	28800	108900	10890
14.10.15	FOREVER	QUEEN	판매	19500	3	58500	167400	16740
14.10.25	SHINE/RIDE ON	동방신기	반품	14400	2	28800	138600	13860

1.문제 정의

-음반관리 프로그램이므로 사용 시 화면에 메뉴가 출력되고 메뉴에 해당하는 번호를 누르면 입력 및 처리가 가능해야 한다. 입고를 누르면 재고가 있는 음반인지 구분하고 신규 입고는 현 데이터의 마지막에 신규 추가 한다. 판매,반품에서는 재고현황표를 먼저 출력하고 판매할 음반의 번호와 수량을 입력해서 재고수량을 감가산한다. 감가산된 내용을 화면에 출력하고 거래내역 데이터를 신규추가하고 거래내역파일에 거래내역을 추가한다. 출력을 누르면 재고현황과 거래내역 중에 어느 항목을 출력할 것인지 선택할 수 있어야 하고 선택된 항목이 화면 출력된다. 종료를 누르면 변경된 재고현황과 신규 추가된 재고현황이 재고현황 파일에 저장되어야 한다.

2.요구사항 분석

2.1기능 정의

음반관리 프로그램

- 파일 저장, 불러오기 기능
- 재고현황 입력 기능
- 음반 판매, 반품기능
- 거래내역서 거래내역파일에 자동 추가기능
- 재고현황, 거래내역서 화면출력 기능

2.2 모듈 별 상세 기능 설명

- 파일저장, 불러오기 기능

재고현황 입력 및 판매, 반품으로 변경된 수량 변경 시 구조체의 데이터를 가지고 있다가 종료할 때 구조체 배열에 있는 데이터를 파일에 저장할 수 있어야 한다. 또한 프로그램 실행 시에 사용자가 원하면 저장된 데이터를 불러올 수 있어야 한다.

- 재고현황 입력 기능

앨범 명에 공백이 삽입되어 있어도 입력을 받을 수 있어야 하고 새로운 재고현황과 기존 재고현황을 비교해서 기존 재고현황 앨범이 있으면 수량을 합산한다. 기존 재고수량에 앨범이 없으면 신규입고를 통해서 현 재고현황의 마지막에 새로운 재고현황을 추가한

다.

●음반 판매, 반품기능

판매기능을 실행하면 우선 현재 재고현황이 출력되어야 한다. 그 다음에 사용자로부터 판매할 음반의 번호를 입력 받아 기존 재고현황 앨범 번호와 비교를 해서 판매할 재고현황 앨범을 찾아낸다. 찾아낸 재고현황 앨범의 수량을 감가산한다. 감가산된 수량을 재고현황 표가 출력되어서 확인할 수 있어야 하고 거래내역 구조체 배열에 새로운 거래내역 데이터가 저장되어야 한다. 거래내역의 거래총액은 기존 거래금액이 있는지 검색해서 기존 거래금액이 있으면 새로운 거래금액을 합산해서 거래총액에 저장한다. 누적부가세는 거래총액의 10%로 계산해서 저장한다.

●거래내역서 거래내역파일에 자동 추가기능

재고현황의 앨범이 판매 및 반품이 발생했을 때 새로운 거래내역이 거래내역 구조체 배열에 저장된다. 저장된 거래내역을 프로젝트가 종료되기 직전 거래내역파일에 추가모드로 해서 파일에 저장이 가능해야 한다.

●재고현황, 거래내역서 화면출력 기능

사용자가 메인 메뉴에서 화면출력을 선택하면 재고현황 출력과 거래내역 출력 중에서 하나를 선택할 수 있어야 하고 선택된 출력기능이 화면에 현재 재고현황이나 거래내역을 출력해줄 수 있어야 한다.

3. 설계

3.1 입력과 자료의 형식 정의

이름	타입	범위	관련 모듈
stock_count	int	프로젝트	프로젝트
transaction_count	int	프로젝트	프로젝트
menu	Menu	main함수	main함수
stock_f	FILE*	프로젝트	main함수
transaction_f	FILE*	프로젝트	main함수
stock_container[MAX]	album_stock	프로젝트	main함수
transaction_container[MAX]	album_transaction	프로젝트	main함수
func_return_number	int	main함수	main함수
import_selection_number	int	main함수	main함수
append_stock_num	int	main함수	main함수
print_selection_num	int	main함수	main함수
exit_returnNum	int	main함수	main함수
i	int	main함수	main함수
j	int	main함수	main함수
container[]	album_stock	album_stock_print	album_stock_print
i	int	album_stock_print	album_stock_print
container[]	album_transaction	album_transaction_print	album_transaction_print
i	int	album_transaction_print	album_transaction_print
function_exit_condition	char	album_transaction_print	album_transaction_print
stock_container[]	album_stock	album_stock_exit	album_stock_exit
stock_f	FILE*	album_stock_exit	album_stock_exit
stock_container[]	album_stock	album_stock_start	album_stock_start
i	static int	album_stock_start	album_stock_start
function_exit_condition	char	album_stock_start	album_stock_start
stock_container[]	album_stock	album_stock_input	album_stock_input
i	int	album_stock_input	album_stock_input
input_amount	int	album_stock_input	album_stock_input
album_name[20]	char	album_stock_input	album_stock_input
stock_container[]	album_stock	album_stock_sale	album_stock_sale

이름	타입	범위	관련 모듈
transaction_container[]	album_transaction	album_stock_sale	album_stock_sale
transaction_f	FILE*	album_stock_sale	album_stock_sale
num1	int	album_stock_sale	album_stock_sale
sell_num	int	album_stock_sale	album_stock_sale
turnover	int	album_stock_sale	album_stock_sale
i	int	album_stock_sale	album_stock_sale
j	int	album_stock_sale	album_stock_sale
function_exit_condition	char	album_stock_sale	album_stock_sale
stock_container[]	album_stock	album_stock_return	album_stock_return
transaction_container[]	album_transaction	album_stock_return	album_stock_return
transaction_f	FILE*	album_stock_return	album_stock_return
i	int	album_stock_return	album_stock_return
j	int	album_stock_return	album_stock_return
return_Album	int	album_stock_return	album_stock_return
return_Amount	int	album_stock_return	album_stock_return
turnover	int	album_stock_return	album_stock_return
function_exit_condition	char	album_stock_return	album_stock_return

3.2 입력 및 출력 창 디자인

-프로그램 시작 화면

처음 프로그램이 시작되면 사용자가 저장된 파일을 불러올 것인지 선택할 수 있도록 화면을 출력하고 저장된 파일을 불러온 다음 재고현황을 더 추가할 수 있도록 프로그램을 구성한다.

-음반 관리 메뉴

사용자가 어떤 관리를 할 것인지 선택 가능하도록 메뉴를 화면에 출력한다.

-재고현황 출력

수량이 변경되거나 신규 입고된 재고가 반영된 재고현황이 출력된다.

-거래내역 출력

사용자가 판매하거나 반품 받은 앨범 현황을 거래내역서에 기록하여 사용자가 원할 때 화면출력 할 수 있다.

-앨범 입고

입고 메뉴를 선택하면 우선 현재 재고현황이 화면에 출력된 다음 입고할 앨범 명을 입력한다. 기존 재고현황에 입력된 앨범 명이 존재하면 현재 재고가 있다는 설명이 나오고 입고할 수량을 입력하라고 나온다. 수량을 입력하면 합산된 재고수량 설명이 나오고 변경된 재고현황표가 화면에 출력된다.

-앨범 신규입고

입력 받은 앨범 명을 기존 재고현황표의 앨범명과 비교하여 존재하지 않는 앨범 명이면 신규입고로 처리한다. 신규입고를 받으면 변경된 재고현황표가 출력된다.

-앨범 판매

사용자가 판매 메뉴를 선택하면 판매할 앨범 번호를 입력 받고 입력 받은 앨범번호와 기존 재고현황 앨범번호를 비교해서 판매할 앨범을 찾는다. 그런 다음 판매할 앨범수량을 입력해서 기존 재고수량에서 수량을 감산한다. 감산 후에는 판매 내역이 화면에 출력되고 판매된 수량과 단가를 곱하여 총 판매액을 보여준다. 판매가 끝난 후에는 신규 거래 내역이 생겼으므로 거래내역을 거래내역 데이터에 추가하고 거래내역 데이터를 거래내역 파일에 추가모드로 저장한다.

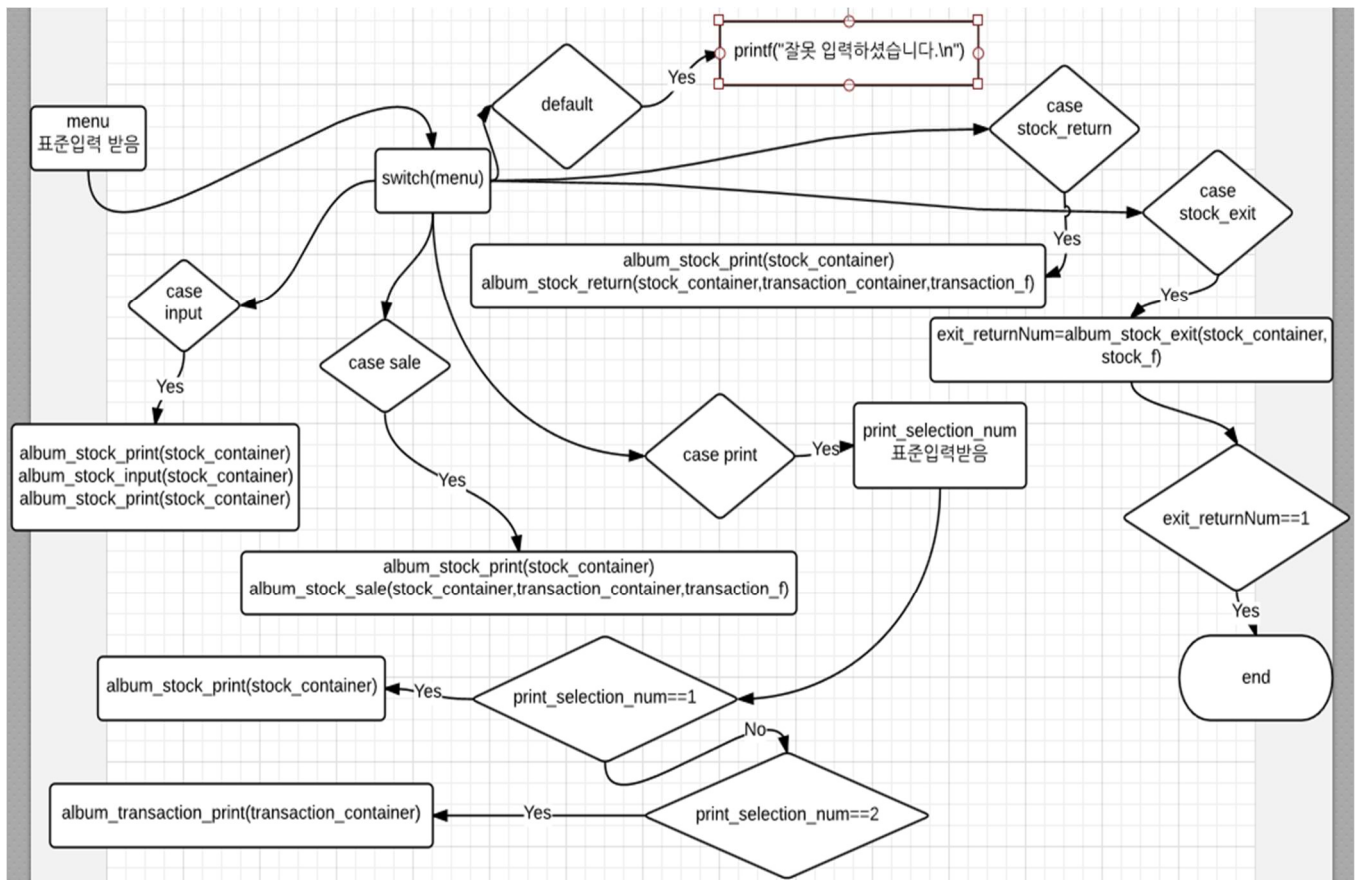
-앨범 반품

사용자가 반품 메뉴를 선택하면 반품할 앨범 번호를 입력 받고 입력 받은 앨범번호와 기존 재고현황 앨범번호를 비교해서 반품할 앨범을 찾는다. 그런 다음 반품할 앨범수량을 입력해서 기존 재고수량에서 새로운 수량을 합산한다. 합산 후에는 반품 내역이 화면에 출력되고 반품된 수량과 단가를 곱하여 총 반환액을 보여준다. 반품이 끝난 후에는 신규 거래내역이 생겼으므로 거래내역을 거래내역 데이터에 추가하고 거래내역 데이터를 거래내역파일에 추가모드로 저장한다

-음반관리 프로그램 종료

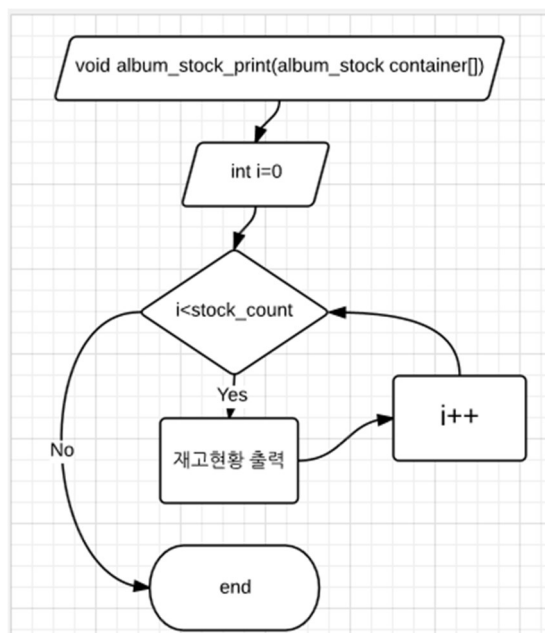
프로그램은 사용자가 종료 메뉴를 선택해야지만 종료된다. 종료메뉴를 선택하면 변경된 재고수량과 거래내역이 파일로 저장된다.

3.3 프로그램 동작 절차

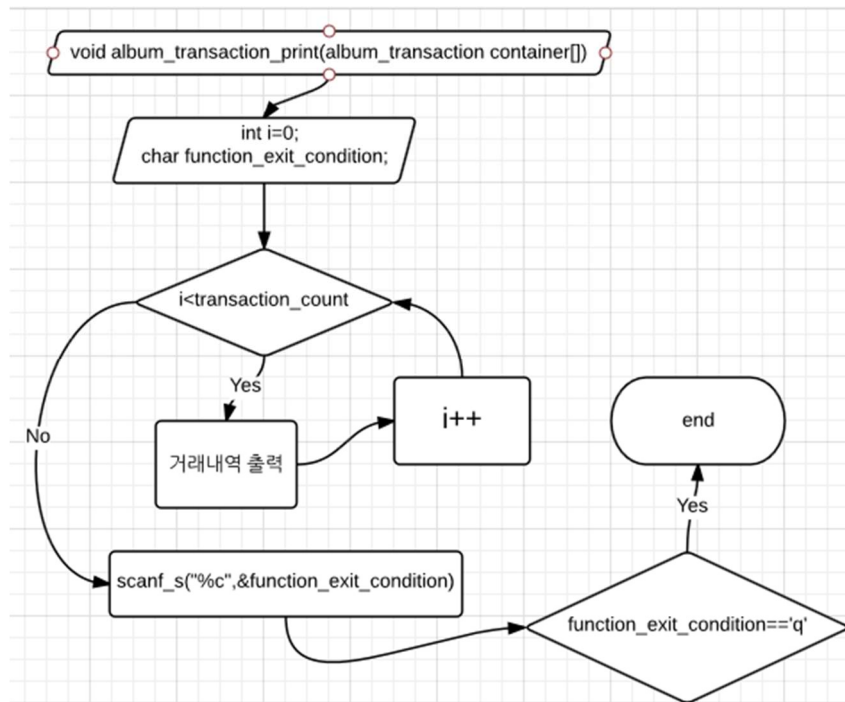


3.4 모듈별 동작 절차

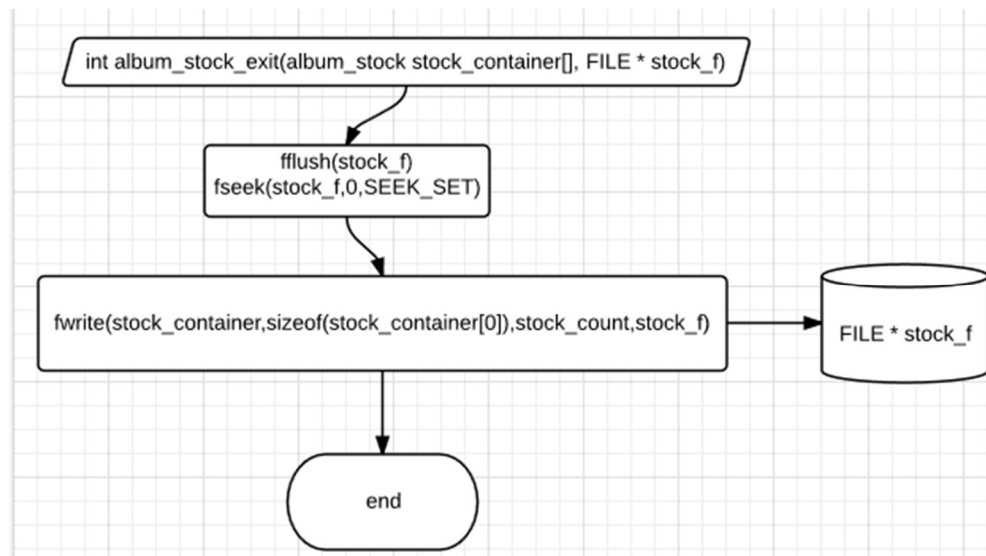
-재고현황 출력 플로우차트



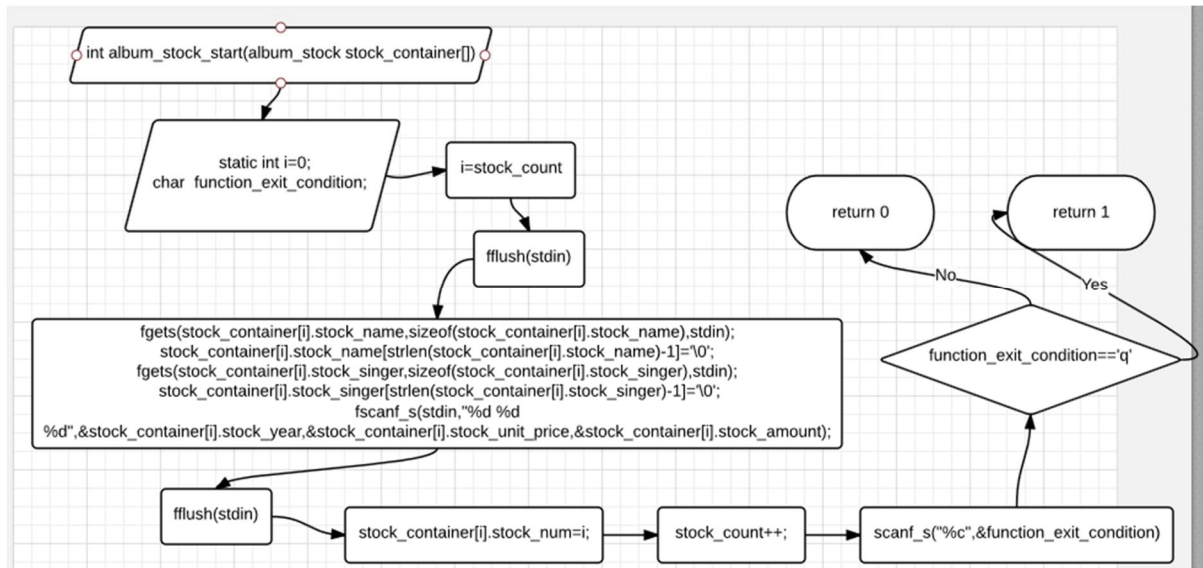
-거래내역 출력 플로우차트



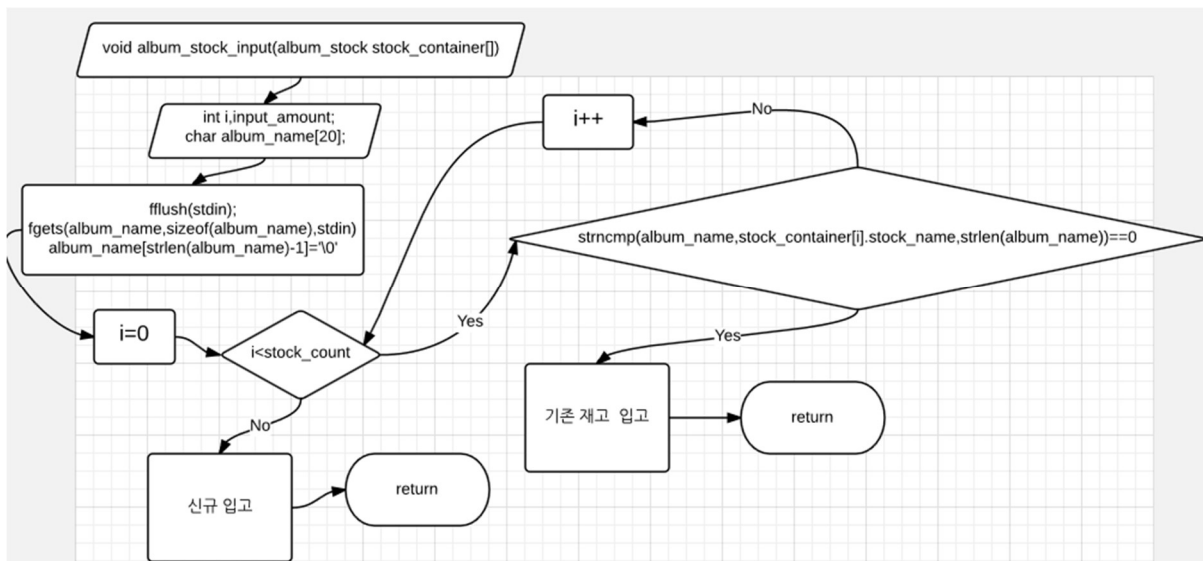
-종료 플로우차트



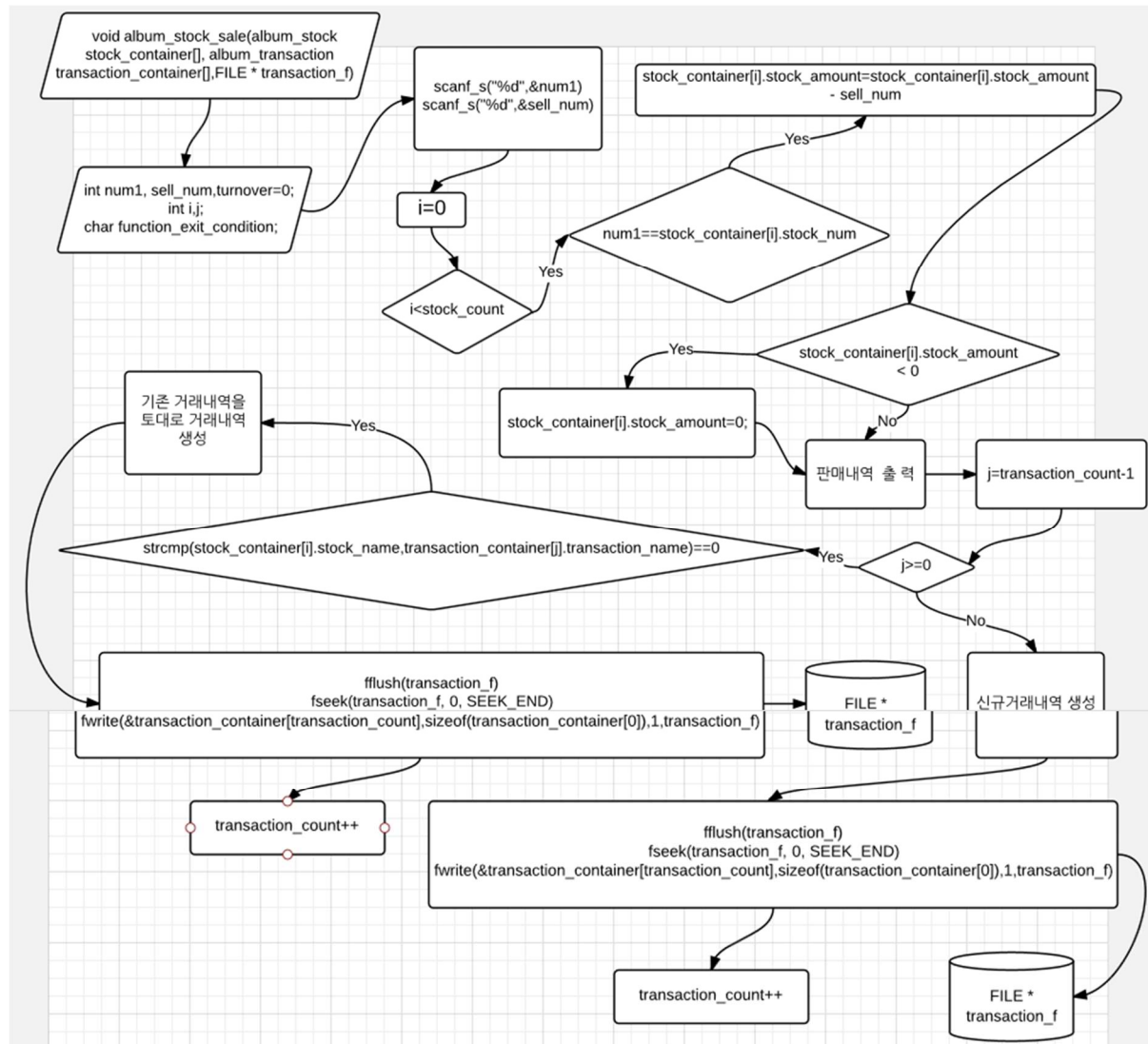
-프로그램 시작 시 재고현황 입력 플로우차트



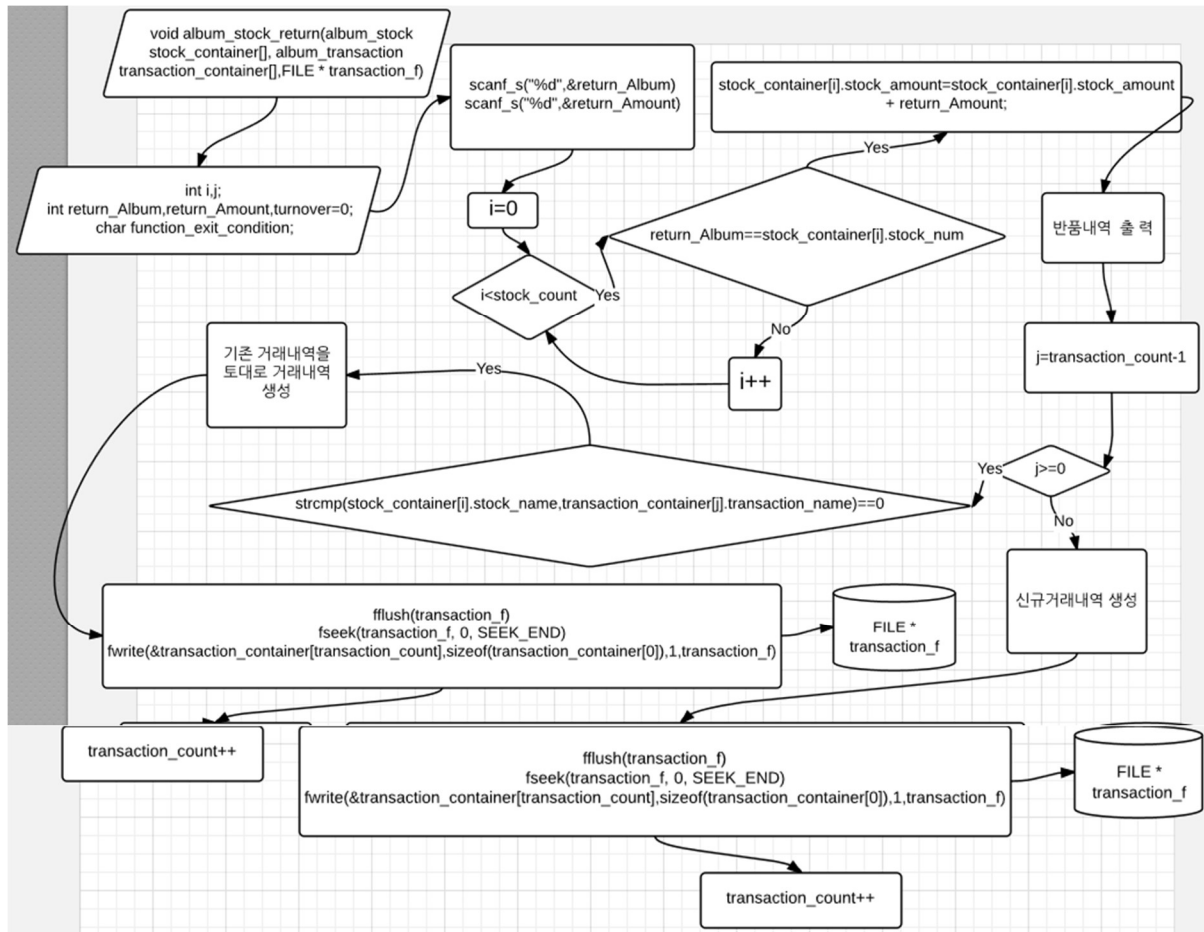
-재고현황 입고 플로우차트



-음반 판매 플로우차트



-음반 반품 플로우차트



3.5 동작 규칙

- 프로젝트 폴더에 `albumdata.bin`, `transaction.bin` 파일이 있어야지 내용 저장 및 불러오기가 가능하다.
- 거래내역은 판매, 반품 함수에서 자동으로 거래내역파일에 추가되지만 임시저장이기 때문에 메인 메뉴에서 종료를 선택해야지 제대로 저장된다. 정상적인 종료를 하지 않으면 내용이 파일에 저장되지 않는다.
- 전처리 설정에 `#define MAX 25`을 설정했는데 MAX는 재고현황과 거래내역을 저장할 수 있는 구조체 배열 길이이다. 저장된 개수가 25를 넘어서 저장하면 memory 오류가 생긴다. 그러므로 더 많은 내용을 저장하고 싶으면 MAX의 설정 값을 변경해주면 된다.
- 앨범 명은 공백도 입력이 되므로 사용자가 원하면 공백을 입력해도 된다. 또한 재고현황을 입력할 때 프로그램에서 입력 방법을 제시하는데 제시되는 방법대로 입력해야 한다.

4.구현

4.1 개발형식

- 프로젝트 이름: Basic_C_project

4.2 소스코드

```
/*
  작성자: 임준수_2010111661_바이오환경과학과
  작성일: ~ 2014.12.12
  주요내용: 기초프로그래밍 프로젝트
*/

#include<stdio.h>
#include<stdlib.h>//exit함수 헤더파일
#include<string.h>//문자열 함수 헤더파일
#define MAX 25//재고현황,거래내역서 구조체 배열 길이

char fname_stock[20]="albumdata.bin";//재고현황 파일이름
char fname_transaction[20]="transaction.bin";//거래내역서 파일이름
char sale_str[15]="판매";
char return_str[15]="반품";

//열거형 상수 구현
typedef enum __Menu {input=1,sale,stock_return,pr int,stock_exit} Menu;

typedef struct album_stock//재고현황 구조체 정의
{
    int stock_num;//재고현황 번호
    char stock_name[20];//앨범명
    char stock_singer[20];//가수 이름
    int stock_year;//년도
    int stock_unit_price;//단가
    int stock_amount;//수량
}album_stock;//자료형 album_stock 정의

typedef struct album_transaction//거래내역 구조체 정의
{
    char transaction_date[15];//거래일자
    char transaction_name[20];//앨범명
    char transaction_singer[20];//가수 이름
    char transaction_type[15];//거래유형
    int transaction_unit_price;//단가
    int transaction_amount;//수량
```

```

    int transaction_money;//거래액
    int transaction_money_turnover;//거래총액
    int transaction_sur tax;//누적부가세
}album_transaction;//자료형 album_transaction 정의

int stock_count=0;//재고현황 개수를 세는 전역변수
int transaction_count=0;//거래내역 횟수를 세는 전역변수

int album_stock_start(album_stock stock_container[]);//재고현황 초기화 함수 선언
void album_stock_input(album_stock stock_container[]);//재고현황 입고 함수 선언
void album_stock_print(album_stock container[]);//재고현황 출력 함수 선언
void album_transaction_print(album_transaction container[]);//거래내역 출력 함수 선언
void album_stock_sale(album_stock stock_container[], album_transaction
transaction_container[],FILE * transaction_f);//판매 함수 선언
void album_stock_return(album_stock stock_container[], album_transaction
transaction_container[], FILE * transaction_f);//반품 함수 선언
int album_stock_exit(album_stock stock_container[], FILE * stock_f);//종료 함수 선언

int main(void)
{
    Menu menu;//자료형이 Menu인 열거형 변수 선언
    FILE *stock_f;//재고현황 파일을 가리키는 파일포인터 변수선언
    FILE *transaction_f;//거래내역 파일을 가리키는 파일포인터 변수선언
    album_stock stock_container[MAX];//재고현황 구조체 배열
    album_transaction transaction_container[MAX];//거래내역 구조체 배열
    int func_return_number, import_selection_number, append_stock_num;
    int print_selection_num, exit_returnNum, i=0, j=0;

    if(fopen_s(&stock_f,fname_stock,"r+b")!=0)//재고현황 읽고 쓰기 모드로 오픈
    {
        fprintf(stderr,"STOCK FILE OPEN ERROR\n");//에러처리
        exit(1);
    }

    if(fopen_s(&transaction_f,fname_transaction,"a+b") != 0)//거래내역 읽고 추가 모드로
    오픈
    {
        fprintf(stderr,"TRANSACTION FILE OPEN ERROR\n");//에러처리
        exit(1);
    }

    printf("1번을 누르시면 저장된 파일을 불러옵니다>");
    scanf_s("%d",&import_selection_number);//파일 불러오기 선택 기능
    fflush(stdin);//입력버퍼 비우기

```

```

if(import_selection_number==1)
{
    //이진파일에 있는 재고현황 데이터 fread 함수로 읽어오기 성공 시 1반환,
실패or파일의 끝 도달 시 0반환
    while(fread(&stock_container[i],sizeof(stock_container[0]),1,stock_f)>0)
    {
        stock_count++;//파일에서 album_stock 구조체 한번 읽어올때 마다 1씩
증가
        i++;//제어변수 1증가
    }
    //이진파일에 있는 거래내역 데이터 fread 함수로 읽어오기 성공 시 1반환,
실패or파일의 끝 도달 시 0반환
    while(fread(&transaction_container[j],sizeof(transaction_container[0]),1,transaction_f
)>0)
    {
        transaction_count++;//파일에서 album_transaction 구조체 한번 읽어올때
마다 1씩 증가
        j++;//제어변수 1증가
    }
}

printf("재고현황을 추가하실려면 1번을 누르세요>");
scanf_s("%d",&append_stock_num);//재고현황 추가 선택기능
fflush(stdin);//\n을 비워준다.

if(append_stock_num==1)
{
    while(1)//무한루프로 구성해서 종료 조건 입력시에만 루프 종료
    {
        func_return_number=album_stock_start(stock_container);//재고현황
입력함수 호출
        if(func_return_number==1)
            break;
    }
}

while(1)//무한루프안에 메뉴를 삽입했기 때문에 종료기능을 선택해야지만 프로그램 종료
{
    //메뉴 출력
    puts("\n음반 판매 관리 프로그램");
    puts("1. 입고");
    puts("2. 판매");
    puts("3. 반품");
    puts("4. 출력(재고현황/거래내역)");
    puts("5. 종료");
}

```



```

printf("번호를 입력하고 ENTER를 누르세요> ");

scanf_s("%d", &menu); //메뉴 선택

switch(menu) //switch 조건에 열거형 변수 삽입
{
    case input: //레이블을 열거형 상수로 구현
        album_stock_print(stock_container); //재고현황 출력함수 호출
        album_stock_input(stock_container); //재고현황 입력함수 호출
        album_stock_print(stock_container); //재고현황 출력함수 호출
        break; //switch문 종료
    case sale: //판매 레이블
        album_stock_print(stock_container);

album_stock_sale(stock_container, transaction_container, transaction_f); //판매함수 호출
        break;
    case stock_return: //반품 레이블
        album_stock_print(stock_container);

album_stock_return(stock_container, transaction_container, transaction_f); //반품함수
호출

        break;
    case print: //출력 레이블
        puts("\n출력할 항목을 선택하세요.");
        puts("1. 재고 현황");
        puts("2. 거래 내역");

printf("번호를 입력하고 ENTER를 누르세요>");

//사용자가 입력한 값에 따라 출력할 항목 선택
scanf_s("%d", &print_selection_num);
if(print_selection_num==1)
{
    album_stock_print(stock_container);
    break;
}
else if(print_selection_num==2)
{
    album_transaction_print(transaction_container);
    break;
}
}
case stock_exit:
{
    exit_returnNum=album_stock_exit(stock_container, stock_f); //종료함수
호출

    if(exit_returnNum==1) //종료함수의 반환 값이 1이면 프로그램 종료

```

```

        {
            fclose(stock_f); //stock_f 파일 스트림 종료
            fclose(transaction_f); //transaction_f 파일 스트림 종료
            return 1;
        }

    }

    default: //예외처리
        printf("잘못 입력하셨습니다.\n");
    }

}

return 0;
}

void album_stock_print(album_stock container[]) //재고현황 출력 함수 정의
{
    int i=0; //제어변수

    printf("\n[[ 재고 현황]]\n");
    printf("No          앨범명          가수          년도  단가  수량\n");
    printf("-----Wn");

    for(i=0; i<stock_count; i++) //재고현황 수만큼 출력
    {
        //fprintf를 사용해서 stdout으로 모니터 출력

        fprintf(stdout, "%d%17s%17s%11d%6d%3dWn", container[i].stock_num, container[i].stock_name
, container[i].stock_singer, container[i].stock_year
, container[i].stock_unit_price, container[i].stock_amount);
    }
    puts("");

    return;
}

void album_transaction_print(album_transaction container[]) //거래내역 출력 함수 정의
{
    int i=0;
    char function_exit_condition; //종료조건을 입력받기 위한 변수

    printf("\n[[ 거래내역서 ]]\n");
    printf("거래일자          앨범명          가수          유형  단가  수량  거래액  거래총액\n");
    printf("-----Wn");
    printf("-----WnWn");

    for(i=0; i<transaction_count; i++) //거래내역 수만큼 출력
    {

```

```

        fprintf(stdout, "%s%15s%11s%7s%6d%3d%8d%8d%8d%8d\n", container[i].transaction_date, container[i].transaction_name, container[i].transaction_singer,
        container[i].transaction_type, container[i].transaction_unit_price, container[i].transaction_amount, container[i].transaction_money,
        container[i].transaction_money_turnover, container[i].transaction_surtax);
    }
    puts("");

    fflush(stdin); //입력 버퍼 비우기
    printf("Wnq를 누르시면 메인 화면으로 돌아갑니다>");
    scanf_s("%c", &function_exit_condition);

    if(function_exit_condition=='q')
        return;
}

int album_stock_exit(album_stock stock_container[], FILE * stock_f) //종료 함수 정의
{
    fflush(stock_f); //main함수에서 입력스트림으로 값을 읽어왔으므로 버퍼를 비워준다
    fseek(stock_f, 0, SEEK_SET); //파일지시자를 파일의 시작으로 이동
    //stock_container 구조체 배열 시작 주소부터 stock_container 구조체 변수 사이즈 *
stock_count 만큼 파일에 데이터 저장
    fwrite(stock_container, sizeof(stock_container[0]), stock_count, stock_f); //출력스트림을
형성해서 파일에 데이터를 저장한다.
    printf("Wn재고현황을 파일에 저장했습니다.Wn");
    puts("프로그램을 종료합니다.Wn");

    return 1;
}

int album_stock_start(album_stock stock_container[]) //재고현황 입력 함수 정의
{
    static int i=0; //변수 i는 static변수로 선언해서 스택영역이 아니라 데이터영역에 올린다
    그러므로 함수가 종료되도 값이 유지된다.
    char function_exit_condition;

    i=stock_count; //재고현황 개수 i에 대입, 만약 파일로부터 3개를 입력 받아오면 배열
인덱스는 0부터 시작하므로 0,1,2가 된다.
        //현 데이터의 마지막에 추가하기 위해서 i에 값 대입
    printf("Wn재고 현황을 입력합니다.Wn");
    printf("앨범명과 가수는 엔터로 구분 년도, 단가, 수량은 스페이스로 구분 입력Wn");

    fflush(stdin); //입력버퍼 비우기

    fgets(stock_container[i].stock_name, sizeof(stock_container[i].stock_name), stdin);
    //fgets함수는 Wn까지 입력받으므로 Wn을 W0로 변경해준다.
    stock_container[i].stock_name[strlen(stock_container[i].stock_name)-1]='W0';

```

```

fgets(stock_container[i].stock_singer,sizeof(stock_container[i].stock_singer),stdin);
stock_container[i].stock_singer[strlen(stock_container[i].stock_singer)-1]='\0';

fscanf_s(stdin,"%d %d %d",&stock_container[i].stock_year,&stock_container[i].stock_unit_price,&stock_container[i].stock_amount);
fflush(stdin);//fscanf는 \n이 입력되어야지 버퍼에 있던 데이터가 변수로 저장된다. 근데 \n은 계속 버퍼에 있으므로 버퍼를 비워줘야 한다.

stock_container[i].stock_num=i;//재고현황 번호 지정

stock_count++;//재고현황을 추가하였으므로 stock_count 1증가

printf("\nq를 입력하시면 재고현황 입력이 종료됩니다>");

scanf_s("%c",&function_exit_condition);
if(function_exit_condition=='q')
    return 1;
else
    return 0;
}
void album_stock_input(album_stock stock_container[])//재고현황 입고 함수 정의
{
    int i,input_amount;//제어변수 i와 사용자로부터 수량을 입력받을 변수 input_amount선언
    char album_name[20];//사용자로부터 입력받은 앨범명을 저장하기 위한 char형 배열

    printf("\n입고할 앨범의 정보를 입력하세요.\n");
    printf("\nt1.앨범명을 입력하고 ENTER를 누르세요>");

    fflush(stdin);//입력버퍼를 비워준다
    fgets(album_name,sizeof(album_name),stdin);//공백을 포함한 앨범명을 입력받는다.
    album_name[strlen(album_name)-1]='\0';//fgets는 \n도 문자로 취급하므로 \n을 찾아서 \0로 바꿔준다.

    for(i=0; i<stock_count; i++)//재고현황 개수-1 만큼 반복문 실행
    {
        //strcmp함수로 재고현황 앨범명과 입력받은 앨범명을 strlen(album_name)
        길이만큼 비교한다. 참이면 0을 반환해서 if문 실행
        if(strcmp(album_name,stock_container[i].stock_name,strlen(album_name))==0)
        {
            printf("\n%d번 앨범은 재고가
            있습니다.\n",stock_container[i].stock_num);
            printf("%d번 앨범에 입고할 수량 입력>",stock_container[i].stock_num);
            scanf_s("%d",&input_amount);//수량 입력

            stock_container[i].stock_amount=stock_container[i].stock_amount+input_amount;//입고할

```

수량하고 기존 수량하고 합산

```
printf("%d번 앨범의 입고된 수량과 합산한  
재고수량 %d개\n", stock_container[i].stock_num,  
stock_container[i].stock_amount);  
  
return;  
}  
  
}  
  
{  
printf("\n신규 입고입니다.\n");  
printf("앨범명과 가수는 엔터로 구분 년도, 단가, 수량은 스페이스로  
구분 입력.\n");  
  
fflush(stdin); //입력버퍼 비우기  
  
fgets(stock_container[stock_count].stock_name, sizeof(stock_container[stock_count].stock_name), stdin);  
stock_container[stock_count].stock_name[strlen(stock_container[stock_count].stock_name)-1] = '\0';  
  
fgets(stock_container[stock_count].stock_singer, sizeof(stock_container[stock_count].stock_singer), stdin);  
stock_container[stock_count].stock_singer[strlen(stock_container[stock_count].stock_singer)-1] = '\0';  
  
fscanf_s(stdin, "%d %d %d", &stock_container[stock_count].stock_year, &stock_container[stock_count].stock_unit_price, &stock_container[stock_count].stock_amount);  
fflush(stdin);  
  
stock_container[stock_count].stock_num = stock_count; //현 재고현황  
마지막에 추가하였으므로 stock_count 값 대입  
  
stock_count++; //stock_count 1증가  
  
return;  
}  
}  
  
//판매함수 정의, 매개변수는 album_stock 구조체 배열, album_transaction 구조체 배열, 거래내역 파일을 가리키는 포인터 변수  
void album_stock_sale(album_stock stock_container[], album_transaction transaction_container[], FILE * transaction_f)  
{  
int num1, sell_num, turnover=0;  
int i, j;
```

```

char function_exit_condition;

printf("판매할 앨범 번호를 입력하고 ENTER를 누르세요>");
scanf_s("%d",&num1);//사용자가 앨범 번호 입력
printf("판매할 앨범의 수량을 입력하고 ENTER를 누르세요>");
scanf_s("%d",&sell_num);//사용자가 수량 입력

for(i=0; i<stock_count; i++)//재고현황 개수-1 만큼 반복문 실행
{
    if(num1==stock_container[i].stock_num)//입력받은 앨범번호와 기존에 있던
    앨범번호가 같으면 if문 실행
    {
        stock_container[i].stock_amount=stock_container[i].stock_amount -
sell_num;//입력받은 수량을 뺀다

        if(stock_container[i].stock_amount < 0)//수량이 음수로 가면 0을
대입해서 수량 0으로 화면 출력
        stock_container[i].stock_amount=0;

        printf("\n[[ 판매 내역]]\n");//변경된 수량과 판매액 화면 출력
        printf("No          앨범명          가수          년도  단가 수량
판매액\n");
        printf("-----\n");

        fprintf(stdout,"%d%17s%17s%11d%6d%3d%8d\n\n",stock_container[i].stock_num,stock_contai
ner[i].stock_name,stock_container[i].stock_singer,stock_container[i].stock_year
        , stock_container[i].stock_unit_price,
sell_num,stock_container[i].stock_unit_price*sell_num);

        printf("거래내역 파일에 새로운 거래 내역을 추가합니다\n");
        printf("거래일자를 입력해주세요.\n");

        for(j=transaction_count-1; j>=0; j--)//거래내역 데이터 마지막에서
부터 1씩 빼면서 처음 거래내역까지 for문 실행
        {
            //기존의 거래내역이 있는지 strcmp 함수로 확인

            if(strcmp(stock_container[i].stock_name,transaction_container[j].transaction_name,str
len(stock_container[i].stock_name))==0)
            {
                //기존 거래내역이 있는 영역

                scanf_s("%s",transaction_container[transaction_count].transaction_date,sizeof(transact
ion_container[transaction_count].transaction_date));
                fflush(stdin);

```

```
strcpy_s(transaction_container[transaction_count].transaction_type, sizeof(transaction_
container[transaction_count].transaction_type), sale_str); //판매유형 복사
```

```
strcpy_s(transaction_container[transaction_count].transaction_name, sizeof(stock_contai
ner[i].stock_name), stock_container[i].stock_name); //이름 복사
```

```
strcpy_s(transaction_container[transaction_count].transaction_singer, sizeof(stock_cont
ainer[i].stock_singer), stock_container[i].stock_singer); //가수 이름 복사
```

```
transaction_container[transaction_count].transaction_unit_price=stock_container[i].sto
ck_unit_price; //재고현황 단가 거래내역 단가에 대입
```

```
transaction_container[transaction_count].transaction_amount=sell_num; //판매된 수량
재고현황 수량에 대입
```

```
transaction_container[transaction_count].transaction_money=transaction_container[trans
action_count].transaction_unit_price*sell_num; //신규거래액 대입
```

```
turnover+=transaction_container[j].transaction_money_turnover; //기존 거래액 가져오기
```

```
transaction_container[transaction_count].transaction_money_turnover=turnover+transacti
on_container[transaction_count].transaction_money; //기존 거래액과 새로운 거래액 합산
```

```
transaction_container[transaction_count].transaction_surtax=transaction_container[tran
saction_count].transaction_money_turnover/10; //누적부가세는 거래총액에서 10을 나눈다
```

```
fflush(transaction_f); //transaction_f 입력스트림을
비운다.
```

```
fseek(transaction_f, 0, SEEK_END); //파일지시자를
transaction_f 파일의 끝으로 이동한다.
```

```
//현재 추가된 거래내역 구조체 변수의 주소에서
거래내역 구조체 변수 사이즈만큼 한번만 파일에 저장
```

```
fwrite(&transaction_container[transaction_count], sizeof(transaction_container[0]), 1, tr
ansaction_f);
```

```
puts("거래 내역 파일에 거래 내역이 추가
되었습니다.");
```

```
transaction_count++; //거래내역 횟수 증가
```

```
fflush(stdin);
```

```
printf("Wnq를 누르시면 메인 화면으로 돌아갑니다>");
```

```
scanf_s("%c", &function_exit_condition);
```

```
if(function_exit_condition=='q')
```

```
return;
```

```
}
```

```

    }

    {
        //신규 거래내역 추가 영역

        scanf_s("%s", transaction_container[transaction_count].transaction_date, sizeof(transaction_
ion_container[transaction_count].transaction_date));
        fflush(stdin);

        strcpy_s(transaction_container[transaction_count].transaction_type, sizeof(transaction_
container[transaction_count].transaction_type), sale_str);

        strcpy_s(transaction_container[transaction_count].transaction_name, sizeof(stock_contai
ner[i].stock_name), stock_container[i].stock_name);

        strcpy_s(transaction_container[transaction_count].transaction_singer, sizeof(stock_cont
ainer[i].stock_singer), stock_container[i].stock_singer);

        transaction_container[transaction_count].transaction_unit_price=stock_container[i].sto
ck_unit_price;

        transaction_container[transaction_count].transaction_amount=sell_num;

        transaction_container[transaction_count].transaction_money=transaction_container[trans
action_count].transaction_unit_price*sell_num;

        turnover+=transaction_container[transaction_count].transaction_money; //신규 거래내역은
        기존 거래액이 없으므로 새로운 거래액만 총액에 합산

        transaction_container[transaction_count].transaction_money_turnover=turnover;

        transaction_container[transaction_count].transaction_surtax=transaction_container[tran
saction_count].transaction_money_turnover/10;

        fflush(transaction_f);
        fseek(transaction_f, 0, SEEK_END);

        fwrite(&transaction_container[transaction_count], sizeof(transaction_container[0]), 1, tr
ansaction_f);

        puts("거래 내역 파일에 거래 내역이 추가 되었습니다.");

        transaction_count++;

        fflush(stdin);
        printf("Wnq를 누르시면 메인 화면으로 돌아갑니다>");
        scanf_s("%c", &function_exit_condition);

        if(function_exit_condition=='q')
            return;
    }
}

```



```

    }
}

//반품함수 정의, 매개변수는 album_stock 구조체 배열, album_transaction 구조체 배열, 거래내역
파일을 가리키는 포인터 변수
void album_stock_return(album_stock stock_container[], album_transaction
transaction_container[], FILE * transaction_f)
{
    int i,j;
    int return_Album,return_Amount,turnover=0;
    char function_exit_condition;

    printf("반품할 앨범 번호를 입력하고 ENTER를 누르세요>");
    scanf_s("%d",&return_Album);//반품할 앨범 번호 입력
    printf("반품할 앨범의 수량을 입력하고 ENTER를 누르세요>");
    scanf_s("%d",&return_Amount);//앨범 수량 입력

    for(i=0; i<stock_count; i++)//재고현황 개수-1 만큼 반복문 실행
    {
        if(return_Album==stock_container[i].stock_num)//입력받은 앨범번호와 기존에
있던 앨범번호가 같으면 if문 실행
        {
            stock_container[i].stock_amount=stock_container[i].stock_amount +
return_Amount;//입력받은 수량을 기존 수량에 더한다.

            printf("\n[[ 반품 내역]]\n");
            printf("No          앨범명          가수          년도   단가   수량
반환액\n");
            printf("-----\n");

            fprintf(stdout,"%d%17s%17s%11d%6d%3d%8d\n\n",stock_container[i].stock_num,stock_contai
ner[i].stock_name,stock_container[i].stock_singer,stock_container[i].stock_year
            , stock_container[i].stock_unit_price,
            return_Amount,stock_container[i].stock_unit_price*return_Amount);

            printf("거래내역 파일에 새로운 거래 내역을 추가합니다\n");
            printf("거래일자를 입력해주세요.\n");

            for(j=transaction_count-1; j>=0; j--)//거래내역 데이터 마지막에서
부터 1씩 빼면서 처음 거래내역까지 for문 실행
            {
                //기존의 거래내역이 있는지 strcmp 함수로 확인

                if(strcmp(stock_container[i].stock_name,transaction_container[j].transaction_name,strlen(stock_container[i].stock_name))==0)
                {
                    //기존 거래내역이 있는 영역

                    scanf_s("%s",transaction_container[transaction_count].transaction_date,sizeof(transact

```

```

ion_container[transaction_count].transaction_date)); //거래일자 입력
        fflush(stdin);

        strcpy_s(transaction_container[transaction_count].transaction_type, sizeof(transaction_
container[transaction_count].transaction_type), return_str);

        strcpy_s(transaction_container[transaction_count].transaction_name, sizeof(stock_contai
ner[i].stock_name), stock_container[i].stock_name);

        strcpy_s(transaction_container[transaction_count].transaction_singer, sizeof(stock_cont
ainer[i].stock_singer), stock_container[i].stock_singer);

        transaction_container[transaction_count].transaction_unit_price = stock_container[i].sto
ck_unit_price;

        transaction_container[transaction_count].transaction_amount = return_Amount; //반품한
앨범수량 거래내역 수량의 대입

        transaction_container[transaction_count].transaction_money = transaction_container[trans
action_count].transaction_unit_price * return_Amount;

        turnover += transaction_container[j].transaction_money_turnover; //기존 거래액을 변수
turnover에 저장

        transaction_container[transaction_count].transaction_money_turnover = turnover + transacti
on_container[transaction_count].transaction_money; //기존 거래액과 새로운 거래액 합산

        transaction_container[transaction_count].transaction_surtax = transaction_container[tran
saction_count].transaction_money_turnover / 10; //누적부가세

        fflush(transaction_f);
        fseek(transaction_f, 0, SEEK_END);

        fwrite(&transaction_container[transaction_count], sizeof(transaction_container[0]), 1, tr
ansaction_f);

        puts("거래 내역 파일에 거래 내역이 추가
되었습니다.");

        transaction_count++;

        fflush(stdin);
        printf("Wnq를 누르시면 메인 화면으로 돌아갑니다>");
        scanf_s("%c", &function_exit_condition);

        if(function_exit_condition == 'q')
            return;
    }
}
{

```

//신규 거래내역 추가 영역

```
scanf_s("%s", transaction_container[transaction_count].transaction_date, sizeof(transaction_container[transaction_count].transaction_date));
fflush(stdin);

strcpy_s(transaction_container[transaction_count].transaction_type, sizeof(transaction_container[transaction_count].transaction_type), return_str);

strcpy_s(transaction_container[transaction_count].transaction_name, sizeof(stock_container[i].stock_name), stock_container[i].stock_name);

strcpy_s(transaction_container[transaction_count].transaction_singer, sizeof(stock_container[i].stock_singer), stock_container[i].stock_singer);

transaction_container[transaction_count].transaction_unit_price = stock_container[i].stock_unit_price;

transaction_container[transaction_count].transaction_amount = return_Amount;

transaction_container[transaction_count].transaction_money = transaction_container[transaction_count].transaction_unit_price * return_Amount;

turnover += transaction_container[transaction_count].transaction_money; //신규 거래내역은
//기존 거래액이 없으므로 새로운 거래액만 총액에 합산

transaction_container[transaction_count].transaction_money_turnover = turnover;

transaction_container[transaction_count].transaction_surtax = transaction_container[transaction_count].transaction_money_turnover / 10;

fflush(transaction_f);
fseek(transaction_f, 0, SEEK_END);

fwrite(&transaction_container[transaction_count], sizeof(transaction_container[0]), 1, transaction_f);

puts("거래 내역 파일에 거래 내역이 추가 되었습니다.");

transaction_count++;

fflush(stdin);
printf("Wnq를 누르시면 메인 화면으로 돌아갑니다>");
scanf_s("%c", &function_exit_condition);

if(function_exit_condition == 'q')
    return;
}
}
}
```

5. 결과

5.1 결과화면 및 설명

1번을 누르시면 저장된 파일을 불러옵니다>1
재고현황을 추가하실려면 1번을 누르세요>

처음 프로그램이 실행되었을 때 저장된 파일을 불러올 것인지 선택할 수 있는 기능과 재고현황을 더 추가할 것인지 선택하는 기능

```
음반 판매 관리 프로그램
1.인고
2.판매
3.반출
4.출력<재고현황/거래내역>
5.종료
번호를 입력하고 ENTER를 누르세요>
```

음반 판매 관리 프로그램 메뉴

```
출력할 항목을 선택하세요.
1.재고 현황
2.거래 내역
번호를 입력하고 ENTER를 누르세요>1

[[재고 현황]]
No      앨범명      가수      년도  단가  수량
-----
0      METEORA      Linkin Park      2003  19000  27
1      I'M DIFFERENT      HI SUHYUN      2014  18700  15
2      Coast to Coast      WESTLIFE      2000  23000  10
3      소녀시대      소녀시대      2010  13500  19
4      걸스데이      걸스데이      2013  24500  7
5      BIGBANG      BIGBANG      2011  30000  12
```

현재 가지고 있는 음반 재고현황 화면출력

출력할 항목을 선택하세요.

- 1.재고 현황
 - 2.거래 내역
- 번호를 입력하고 ENTER를 누르세요>2

[[거래내역서]]

거래일자	앨범명	가수	유형	단가	수량	거래액	거래총액	누적부가세
14.10.12	Coast to Coast	WESTLIFE	판매	23000	7	161000	161000	16100
14.12.01	걸스데이	걸스데이	판매	24500	5	122500	122500	12250
14.12.02	걸스데이	걸스데이	반품	24500	2	49000	171500	17150
14.12.05	BIGBANG	BIGBANG	판매	30000	5	150000	150000	15000
14.12.06	Coast to Coast	WESTLIFE	반품	23000	10	230000	391000	39100
14.12.07	I'M DIFFERENT	HI SUHYUN	반품	18700	5	93500	93500	9350
14.12.08	I'M DIFFERENT	HI SUHYUN	판매	18700	5	93500	187000	18700
14.12.09	BIGBANG	BIGBANG	판매	30000	2	60000	210000	21000
14.12.09	단발머리	AOA	판매	15000	5	75000	75000	7500

q를 누르시면 메인 화면으로 돌아갑니다>

판매, 반품 거래내역을 가지고 있는 거래내역서 화면출력

No	앨범명	가수	년도	단가	수량
0	METEORA	Linkin Park	2003	19000	27
1	I'M DIFFERENT	HI SUHYUN	2014	18700	15
2	Coast to Coast	WESTLIFE	2000	23000	10
3	소녀시대	소녀시대	2010	13500	19
4	걸스데이	걸스데이	2013	24500	7
5	BIGBANG	BIGBANG	2011	30000	12

입고할 앨범의 정보를 입력하세요.
1.앨범명을 입력하고 ENTER를 누르세요>METEORA

0번 앨범은 재고가 있습니다.
0번 앨범에 입고할 수량 입력>3
0번 앨범의 입고된 수량과 합산한 재고수량 30개

[[재고 현황]]

No	앨범명	가수	년도	단가	수량
0	METEORA	Linkin Park	2003	19000	30
1	I'M DIFFERENT	HI SUHYUN	2014	18700	15
2	Coast to Coast	WESTLIFE	2000	23000	10
3	소녀시대	소녀시대	2010	13500	19
4	걸스데이	걸스데이	2013	24500	7
5	BIGBANG	BIGBANG	2011	30000	12

입고 기능을 선택하면 앨범 명을 입력하는데 기존 재고현황에 앨범이 존재하면 앨범의 입고된 수량과 기존 수량을 합산해서 재고현황을 화면에 출력한다.

```

4      걸스데이      걸스데이      2013 24500 7
5      BIGBANG      BIGBANG      2011 30000 12

입고할 앨범의 정보를 입력하세요.
1.앨범명을 입력하고 ENTER를 누르세요>secret

신규 입고입니다.
앨범명과 가수는 엔터로 구분 년도, 단가, 수량은 스페이스로 구분 입력.
SECRET
SECRET
2013 26000 10

[[재고 현황]]
No      앨범명      가수      년도 단가 수량
-----
0      METEORA      Linkin Park      2003 19000 30
1      I'M DIFFERENT      HI SUHYUN      2014 18700 15
2      Coast to Coast      WESTLIFE      2000 23000 10
3      소녀시대      소녀시대      2010 13500 19
4      걸스데이      걸스데이      2013 24500 7
5      BIGBANG      BIGBANG      2011 30000 12
6      SECRET      SECRET      2013 26000 10

```

입력 받은 앨범이 기존 재고현황에 존재하지 않는 앨범이면 신규입고 기능을 이용해서 새로운 앨범을 입력 받은 후 변경된 재고현황을 화면에 출력한다.

```

[[재고 현황]]
No      앨범명      가수      년도 단가 수량
-----
0      METEORA      Linkin Park      2003 19000 30
1      I'M DIFFERENT      HI SUHYUN      2014 18700 15
2      Coast to Coast      WESTLIFE      2000 23000 10
3      소녀시대      소녀시대      2010 13500 19
4      걸스데이      걸스데이      2013 24500 7
5      BIGBANG      BIGBANG      2011 30000 12
6      SECRET      SECRET      2013 26000 10

판매할 앨범 번호를 입력하고 ENTER를 누르세요>0
판매할 앨범의 수량을 입력하고 ENTER를 누르세요>10

[[판매 내역]]
No      앨범명      가수      년도 단가 수량 판매액
-----
0      METEORA      Linkin Park      2003 19000 10 190000

거래내역 파일에 새로운 거래 내역을 추가합니다
거래일자를 입력해주세요.
14.12.10
거래 내역 파일에 거래 내역이 추가 되었습니다.

```

판매할 앨범번호를 선택하면 재고현황에 있는 앨범이 선택되고 판매할 앨범 수량을 입력하면 기존 재고수량에서 수량이 감산되고 단가와 판매수량을 곱한 후 판매 내역을 통해서 판매액을 보여준다.

[[재고 현황]]						
No	앨범명	가수	년도	단가	수량	
0	METEORA	Linkin Park	2003	19000	20	
1	I'M DIFFERENT	HI SUHYUN	2014	18700	15	
2	Coast to Coast	WESTLIFE	2000	23000	10	
3	소녀시대	소녀시대	2010	13500	19	
4	걸스데이	걸스데이	2013	24500	7	
5	BIGBANG	BIGBANG	2011	30000	12	
6	SECRET	SECRET	2013	26000	10	
반품할 앨범 번호를 입력하고 ENTER를 누르세요>4						
반품할 앨범의 수량을 입력하고 ENTER를 누르세요>3						
[[반품 내역]]						
No	앨범명	가수	년도	단가	수량	반환액
4	걸스데이	걸스데이	2013	24500	3	73500
거래내역 파일에 새로운 거래 내역을 추가합니다						
거래일자를 입력해주세요.						
14.12.10						
거래 내역 파일에 거래 내역이 추가 되었습니다.						

반품할 앨범번호를 선택하면 재고현황에 있는 앨범이 선택되고 반품할 앨범 수량을 입력하면 기존 재고수량과 수량이 합산되고 단가와 반품수량을 곱한 후 반품 내역을 통해서 반환액을 보여준다.

음반 판매 관리 프로그램	
1. 입고	
2. 판매	
3. 반품	
4. 출력<재고현황/거래내역>	
5. 종료	
번호를 입력하고 ENTER를 누르세요> 5	
재고현황을 파일에 저장했습니다.	
프로그램을 종료합니다.	
계속하려면 아무 키나 누르십시오 . . .	

사용자가 종료기능을 선택하면 프로그램 사용 중에 있었던 재고현황 추가, 변경과 생성된 거래내역이 파일에 저장되면서 프로그램이 종료된다.

5.2 평가

프로젝트 문제는 텍스트파일에 있는 정보를 불러오고 저장해서 사용하라고 명시되어있다. 하지만 텍스트파일로 프로젝트를 구현하니 공백처리에 많은 애로사항을 느꼈고 새로운 재고현황을 추가할 때 메모리에러가 생기게 되었다. 그래서 교수님께 여쭙보고 바이너리 파일로 프로젝트를 해도 된다고 하셔서 문제와는 조금 다르게 바이너리모드로 프로

젝트를 완성하였다. 텍스트모드로 프로그램을 구성하지 못한 것에 대해서는 문자열 처리의 실력 부족으로 느껴진다.

바이너리로 프로젝트를 구성하면서 내가 목표한 구현 수준에는 어느 정도 도달했다고 생각한다. 앨범 명의 공백이 삽입되어 있어도 공백을 문자열의 길이에 포함해서 재고현황을 추가할 수 있도록 구현하였고 거래내역을 추가할 때 기존 거래금액이 있는지 판단해서 거래총액 계산을 가능하도록 구현하였다. 기초프로그래밍 수업을 듣기 전에는 단순한 계산기도 구현할까 말까 했던 실력이 이 정도 난이도의 프로젝트를 내가 구현한 것에 대해서는 어느 정도 만족하고 싶다.

6. 결론

6.1 소감 및 향후 계획

프로그램의 라인수가 늘어나면서 함수간에 논리적인 연관 관계와 변수의 쓰임에 대해서 많은 것을 배우게 되었다. 또한 프로그래밍 강의를 들으면서 매번 느끼는 것이 있는데 프로그래밍 실력은 하루 아침에 이루어 지는 것이 아니라 오랜 시간 노력이 필요하다고 느끼게 되었다. 예전까지는 대강하면 되겠지 라는 안이한 생각을 가지고 있었다면 요번 프로젝트를 수행하고 공부를 함에 있어서 나의 부족한 실력을 뼈저리게 느끼게 되었고 프로그래밍을 더 공부하고자 하는 열정이 생기게 되었다. 우선 컴퓨터공학과로 전과를 신청하였는데 전과가 된다면 학교에서는 주는 커리큘럼대로 학업을 수행하고 전과가 되지 못한다면 그래도 프로그래밍을 계속 공부해 나갈 것이다. 프로그래밍을 하면서 다른 공부를 할 때는 시간이 너무 느리게 가고 지루했는데 프로그래밍을 할 때는 시간이 가는 줄도 모르고 책상에 주구장창 앉아서 프로그래밍을 하고 있고 잘 때나 밖에서 돌아다닐 때 구현이 되지 않는 부분을 생각하면서 어떻게 하면 구현이 될까? 라는 생각이 머리 속에서 계속 맴돌았다. 아직까지는 기초가 제대로 잡혀있지 않아 어려움이 많고 잘 안 풀리는 경우가 많지만 꾸준히 하다 보면 어느 순간 영감이 올 것이라고 생각하고 계속 공부해나갈 것이다.