

CSED101. Programming & Problem solving

Fall, 2016

Programming Assignment #5 (70 points)

한상욱(han403@postech.ac.kr)

■ **Due:** 2016.12.16 23:59

■ **Development Environment:** Windows Visual Studio 2015

■ 제출물

- **C Code files (*.c, *.h)**
 - 프로그램의 소스 코드를 이해하기 쉽도록 반드시 주석을 붙일 것.
- **보고서 파일** (.doc(x) or .hwp) 예) assn5.doc(x) 또는 assn5.hwp
 - AssnReadMe.pdf 를 참조하여 작성할 것.
 - 프로그램 실행 화면을 캡처하여 보고서에 포함시키고 간단히 설명할 것.
- 소스코드와 보고서 파일을 LMS를 이용하여 제출한다.

■ 주의사항

- 각 문제에 해당하는 요구사항을 반드시 지킬 것.
- 모든 문제의 출력 형식은 아래의 예시들과 동일해야 하며, 같지 않을 시는 감점이 된다.
- 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. (LMS의 과목 공지사항의 제목 [document about cheating]의 첨부파일인 disciplinary.pdf를 참조할 것.)
- 과제를 작성하는데 있어서 전역변수를 선언하여 사용할 수 없다.
- 이번 과제에서는 추가 기능 구현에 대한 추가 점수는 없다.
- 이번 과제는 구조체와 연결리스트 활용을 목표로 한다. 따라서, 문제에서 지정한 부분에서 구조체와 연결리스트를 사용하지 않고 배열을 통해서 해결할 경우 감점 처리한다.

■ Problem 1: 개설 과목 관리 프로그램(35점)

(문제)

파일로부터 개설 과목이 저장된 파일을 읽어서 개설 과목을 관리하는 프로그램을 작성한다.

(목적)

- Structure와 linked list의 사용법을 익힌다.
- 명령줄인수(argc, argv)의 활용 방법을 익힌다.
- 다중 소스파일의 사용법을 익힌다.

(주의사항)

- 보고서는 assn5.doc 또는 assn5.hwp로 저장할 것(보고서는 통합하여 작성)
- 이번 과제는 여러 개의 파일로 분할해서 작성한다.
 - **subjectList.h** (수강과목의 구조체를 정의하고 링크드 리스트 구현을 위한 함수 선언)
 - **subjectList.c** (링크드 리스트 구현을 위한 함수 정의)
 - **assn5_1.c** (main() 함수를 포함하여 필요한 함수들 정의)
- string.h 에서 제공하는 라이브러리 함수를 사용할 수 있다.
- 프로그램은 사용자로부터 4개의 명령어를 입력 받아 각 기능을 수행하게 된다. 파일로부터 데이터를 읽어 리스트를 생성하는 부분과 각 명령어는 독립적인 함수를 정의하여 사용한다. 즉, main()함수를 제외하고 최소한 5개의 함수가 정의되어야 한다.
- 문제의 출력 형식은 채점을 위해 아래의 실행 예시와 최대한 비슷하게 작성한다.

(설명 및 요구사항)

- 윈도우의 커맨드 라인에서 다음과 같이 입력하여 프로그램 실행은 다음과 같이 한다. 노란색으로 표시된 문자는 사용자 입력에 해당한다.

```
C:\> assn5_1.exe subject.txt
```

- 파일명을 입력하지 않은 경우, default로 "subject.txt"로부터 데이터를 읽는다.
- 파일명이 입력된 경우, 입력된 파일로부터 데이터를 읽는다.
- 파일명의 최대길이는 20자이며 파일명에는 공백이 없다고 가정한다.
- 예외처리) 읽을 파일이 존재하지 않는 경우: 아래의 메시지 출력 후, 프로그램을 종료한다.

```
파일을 읽어올 수 없습니다.
```

```
계속하려면 아무 키나 누르십시오 . . .
```

- 프로그램 실행시, 수강 과목 정보를 가진 텍스트 파일(subject.txt)로부터 수강 과목 목록 작성을 위한 데이터를 읽으며, 파일의 내용 및 구성은 아래와 같다.

```
PHYS106 일반물리개론II 3
PHYS104 일반물리실험II 1
CSED103 프로그래밍입문 3
```

LIFE103 일반생명과학 3

GEDU101 글쓰기 2

MATH120 응용선형대수 3

가) 각 과목에 대한 **과목코드, 과목이름, 학점**으로 구성되어 있으며, 각 항목 사이는 공백으로 구분된다.

- 과목과 과목 사이는 줄 바꿈 문자('\n')로 구분된다.
- 위의 텍스트 파일을 읽어 과목 목록을 **링크드 리스트**로 구현 시, 과목 코드를 기준으로 **알파벳 순서**로 정렬된 리스트를 갖도록 한다.

나) **과목 목록**은 링크드 리스트로 구현한다. 아래 예시의 구조체 SUBJECT 를 이용해서 관리한다.

```
typedef struct subject
{
    char sub_code[8];
    char sub_name[41];
    int credit;
    struct subject *next;
} SUBJECT;
```

- 각 과목에 대한 SUBJECT 구조체는 과목이 추가될 때마다 각각 동적할당 받아 사용한다.
- 과목명(sub_name)은 공백 없는 한글 문자열로 최대 20자
- 학점(credit)은 한 자리 숫자
- 과목코드명(sub_code)은 알파벳과 숫자의 조합으로 7개의 문자열
 - 코드명은 식별자로 중복되지 않는다.
 - 코드 목록은 알파벳 순서로 항상 정렬된 상태를 유지한다.

과목코드는 네 자리의 문자와 세 자리의 숫자로 구성된다. 앞의 네 자리 문자는 교과목 개설학과, 뒤의 세 자리의 숫자는 임의의 고유번호를 나타낸다. 개설학과가 같은 경우에는 뒤의 세자리 숫자의 오름차순 정렬을 한다. 예를 들면, PHYS104, PHYS106 순서 이다

다) 프로그램 실행 시, 초기화면은 아래와 같이 과목 목록이 출력된다. 출력 된 후에는 명령어를 입력 받을 준비를 한다.

과목 목록

```
1 CSED103 프로그래밍입문 (3학점)
2 GEDU101 글쓰기 (2학점)
3 LIFE103 일반생명과학 (3학점)
4 MATH120 응용선형대수 (3학점)
5 PHYS104 일반물리실험II (1학점)
6 PHYS106 일반물리개론II (3학점)
```

>>

➤ 수강 과목 관리를 위해 명령어 **show, add, delete, quit** 을 입력 받아 처리하도록 한다. 명시

된 4가지 명령어 외의 명령어가 입력될 경우 무시하고 다시 명령어 입력을 위해 기다린다.

- **과목 목록 보기: show** 명령어 입력 시, 전체 과목 목록을 과목코드를 기준으로 알파벳 순으로 출력해준다

```
>> show
과목 목록
1 CSED103 프로그래밍입문 (3학점)
2 GEDU101 글쓰기 (2학점)
3 LIFE103 일반생명과학 (3학점)
4 MATH120 응용선형대수 (3학점)
5 PHYS104 일반물리실험II (1학점)
6 PHYS106 일반물리개론II (3학점)

>>
```

- **과목 삭제: delete** 명령어 입력 시, 삭제하고자 하는 과목 코드를 입력 받아 해당하는 과목의 항목을 삭제한다.
예외처리) 입력한 과목코드를 가진 과목이 존재하지 않는 경우 "Not found"를 출력한다.

```
>> delete
과목코드: CSED101
Not found

>> delete
과목코드: CSED103

>> show
과목 목록
1 GEDU101 글쓰기 (2학점)
2 LIFE103 일반생명과학 (3학점)
3 MATH120 응용선형대수 (3학점)
4 PHYS104 일반물리실험II (1학점)
5 PHYS106 일반물리개론II (3학점)

>>
```

- **과목 추가: add** 명령어 입력 시, 아래 예제와 같이 과목코드, 과목명, 학점을 순서대로 입력 받아 새로운 과목을 추가한다.
예외처리) 중복되는 과목코드가 입력된 경우 더 이상 진행되지 않고 다시 명령어를 입력 받을 준비를 한다.

```

>> add
과목코드: PHYS104
<-- 과목코드가 이미 있는 경우

>> add
과목코드: CSED101
과목명: 프로그래밍과문제해결
학점: 3

>> add
과목코드: GEDU152
과목명: 검도
학점: 1

>> show
과목 목록
1 CSED101 프로그래밍과문제해결 (3학점)
2 GEDU101 글쓰기 (2학점)
3 GEDU152 검도 (1학점)
4 LIFE103 일반생명과과학 (3학점)
5 MATH120 응용선형대수 (3학점)
6 PHYS104 일반물리실험II (1학점)
7 PHYS106 일반물리개론II (3학점)

>>

```

- **수강 과목 관리 종료:** quit 명령어 입력 시, 프로그램을 종료한다.
 프로그램을 종료하기 전,
 - 수정한 과목 목록을 파일이름을 입력 받아 파일에 저장한다.
 - 동적할당을 받아 사용했던 모든 메모리를 반드시 할당해제 하도록 한다.

```

>> quit
파일명: new_subject.txt
계속하려면 아무 키나 누르십시오 . . .

```

new_subject.txt

```

CSED101 프로그래밍과문제해결 3
GEDU101 글쓰기 2
GEDU152 검도 1
LIFE103 일반생명과과학 3
MATH120 응용선형대수 3
PHYS104 일반물리실험II 1
PHYS106 일반물리개론II 3

```

■ Problem 2: 성적 처리 프로그램(35점)

(문제)

파일로부터 학생에 대한 정보와 학생이 수강하는 과목 및 성적이 저장된 파일을 읽어서 성적을 처리하는 프로그램을 작성한다.

(주의사항)

- 보고서는 assn5.doc 또는 assn5.hwp로 저장할 것(보고서는 통합하여 작성)
- Problem 2의 프로그램은 3개의 메뉴로 이루어져 있다. 파일을 읽어 학생 목록 리스트를 생성하는 부분과 각 메뉴는 각각의 독립적인 함수로 작성 하도록 한다.
- Problem 1에서 작성한 **subjectList.h**를 include하여 사용한다.
- 여러 개의 파일로 분할해서 작성한다
 - **studentList.h** (학생 정보의 구조체를 정의하고 링크드 리스트 구현 관련 함수 선언)
 - **studentList.c** (링크드 리스트 구현을 위한 함수 정의)
 - **assn5_2.c** (main() 함수를 포함하여 필요한 함수들 정의)

(설명 및 요구사항)

- 윈도우의 커맨드 라인에서 다음과 같이 입력하여 프로그램 실행은 다음과 같이 한다. 노란색으로 표시된 문자는 사용자 입력에 해당한다.

```
C:\> assn5_2.exe new_subject.txt student.txt
```

- 명령줄 인수는 실행파일명을 제외하고 2개의 입력을 가지는데, 순서는 수강과목정보파일("new_subject.txt"), 학생정보파일("student.txt") 순이며, 정보가 바뀌어 입력되는 경우는 없다고 가정한다. 또한 과목정보파일 또는 학생정보파일만 입력되는 경우 역시 없다고 가정한다.
- 파일명의 최대길이는 20자이며 파일명에는 공백이 없다고 가정한다.
- 예외처리 1) 파일 읽기에서 읽을 파일이 존재하지 않을 경우
아래의 메시지 출력 후, 프로그램을 종료한다.

```
파일을 읽어올 수 없습니다.
```

```
계속하려면 아무 키나 누르십시오 . . .
```

- 예외처리 2) 프로그램이 제시한 선택 목록에 없는 정수 범위의 입력이 들어온 경우
예를 들면, 사용자가 1~4 사이의 선택을 해야 하나, 5나 6 등의 입력을 넣은 경우를 말하며, "**올바른 입력이 아닙니다.**" 라는 메시지 출력 후, 다시 입력을 받는다.
- 프로그램 실행 시, 읽어 들인 과목정보파일(new_subject.txt)과 학생정보파일(students.txt)로부터 수강 과목 목록과 학생 목록의 2개의 링크드 리스트를 생성한다.
과목정보파일에 없는 과목을 학생이 수강하는 경우는 없다고 가정한다.
 - 과목 목록을 링크드 리스트로 구현한다.
 - problem1 에서 설명한 파일의 구성과 내용이 동일하다, subjectList.h 를 include 하여 과목 목록을 링크드 리스트로 생성한다.

➤ 학생 목록을 링크드 리스트로 구현한다.

(1) 학생 정보 파일의 구성과 내용은 아래와 같다.

- 각 학생에 대한 정보는 한 줄 띄움으로 구별한다.
- 각 학생에 대한 정보 중 첫 번째 줄은 다음과 같은 형태로 주어진다. (각 데이터들은 하나의 공백으로 구분된다.)

이름 학번 학과 학년

- 두 번째 줄부터는 그 학생이 수강하는 과목에 대한 정보로 아래의 형태로 주어지며, 최소한 1과목 이상을 수강한다.

수강과목코드 성적

- 한 학생이 수강하는 과목수는 최대 8과목으로 제한한다.
- 학생 목록은 파일에서 읽은 순서대로 목록을 관리하도록 한다. 즉, 출력시 파일에서 읽은 순서대로 출력되도록 한다.

```
20160001 홍길동 컴공 1
CSED101 A0
PHYS106 B+
PHYS104 A0
LIFE103 A-
GEDU101 B0
MATH120 A+
GEDU152 A+
```

```
20160023 김기리 컴공 1
MATH120 C0
CSED101 F
LIFE103 B-
GEDU101 B+
```

```
20160135 김지민 화공 1
LIFE103 A0
PHYS106 B-
PHYS104 A0
CSED101 A-
MATH120 C0
```

(2) 아래 예시의 구조체 STUDENT 를 이용해서 관리한다.

```
typedef struct
{
    char sub_code[8];
    char grade[3];
}TAKING_SUBJECTS;

typedef struct student
{
    int id;
    char name[11];
    char dept[7];
    int level;
    int sub_num;
    TAKING_SUBJECTS subjects[8];
    struct student *next;
}STUDENT;
```

- 각 학생에 대한 STUDENT 구조체는 학생이 추가될 때마다 각각 동적할당 받아 사용한다.
- 학번(id)은 8자리의 정수(ex. 20160001)로 구성되며, 학번은 식별자(identifier)로 중복되지 않는다. (힌트: **atoi()** 10진 문자열을 int 형 정수값으로 변환하는 함수 사용 가능, 사용시 **stdlib.h**를 포함시킬 것)
- 이름(name)은 공백 없는 한글 또는 영어 문자열(한글 최대 5글자, 영어 최대 10글자)
※ 한글 문자 1개는 2byte의 저장공간이 필요하다.
- 학과(dept)는 공백 없는 한글 문자열로 최대 3글자
- 학년(level)은 1 ~ 4 까지의 숫자
- sub_num은 현재 이수 중인 과목 수
- 수강과목(subjects)은 최대 8과목으로 구조체 TAKING_SUBJECTS 형의 배열로 구성
 - 수강과목코드(sub_code)는 알파벳과 숫자의 조합으로 7개의 문자열
 - 성적(grade)은 A+, A0, A-, ... , F 형태로 구성된다.

➤ 프로그램이 실행되면, 아래와 같은 메뉴 화면이 출력되고, 메뉴 입력을 기다린다.

```
### 성적 관리 프로그램 ###
1. 학생별 성적
2. 과목별 성적
0. 종료
>>
```

➤ 학생별 성적

- 숫자 1 을 입력 시, 출력 예시는 아래와 같다.
각 학생 별 학번, 이름, 학과, 학년, 과목 코드 및 각 성적, 평점 평균을 출력한다.

>> 1

학번	이름	학과(학년)	CSED101	GEDU101	GEDU152	LIFE103	MATH120	PHYS104	PHYS106	평점평균
20160001	홍길동	컴공(1)	A0	B0	A+	A-	A+	A0	B+	3.76
20160023	김기리	컴공(1)	F	B+	-	B-	C0	-	-	1.88
20160135	김지민	화공(1)	A-	-	-	A0	C0	A0	B-	3.17

성적 관리 프로그램

1. 학생별 성적

- 과목코드 목록은 알파벳 순서로 정렬되어 출력한다.
- 학업성적에 따른 평점은 아래와 같다.

성적	A+	A0	A-	B+	B0	B-	C+	C0	C-	D+	D0	D-	F
평점	4.3	4.0	3.7	3.3	3.0	2.7	2.3	2.0	1.7	1.3	1.0	0.7	0.0

- 각 학생별 평균 평점은 각 과목에 대한 (평점x학점)을 하여 합한 후, 전체 이수 학점으로 나눈 값이다.
- 학생별 성적이 모두 출력된 후, 메뉴 화면이 다시 출력되고 메뉴 입력을 기다린다.

➤ 과목별 성적

- 숫자 2 를 입력 시, 출력 예시는 아래와 같다.

서브 메뉴로 과목 목록이 아래와 같이 출력된다. 0을 선택하면 상위 메뉴로 돌아간다. 해당 과목의 번호를 입력하면 해당 과목의 이름, 과목 코드, 학점을 출력한다. 해당 과목 정보와 과목을 듣는 학생들의 명단과 학점이 출력되고 마지막에 과목평균이 출력된다. 아래의 예제는 과목번호로 1을 선택했을 경우이다.

```
>> 2
과목 목록
1 CSED101 프로그래밍과문제해결 (3학점)
2 GEDU101 글쓰기 (2학점)
3 GEDU152 검도 (1학점)
4 LIFE103 일반생명과학 (3학점)
5 MATH120 응용선형대수 (3학점)
6 PHYS104 일반물리실험II (1학점)
7 PHYS106 일반물리개론II (3학점)
나가기(0) >> 1

과목명: 프로그래밍과문제해결(CSED101), 학점: 3학점

학번      | 이름      | 학과(학년) | 학점
20160001  | 홍길동    | 컴공(1)    | A0(4.0)
20160023  | 김기리    | 컴공(1)    | F(0.0)
20160135  | 김지민    | 화공(1)    | A-(3.7)

** 과목평균: 2.57

과목 목록
1 CSED101 프로그래밍과문제해결
```

과목 정보로 과목명, 과목 코드, 학점이 출력되어야 한다.

그리고, 그 아래에 각 수강 학생의 정보로 학번, 이름, 학과, 학년, 학점이 출력되어야 한다

실행 후, 과목 목록이 다시 출력되고, 0을 누른 경우 상위 메뉴로 돌아간다.

➤ 종료

- 최상위 메뉴에서 0을 선택하면, 프로그램을 종료한다.

프로그램을 종료하기 전, 동적할당을 받아 사용했던 모든 메모리를 반드시 할당해제하도록 한다.