

# CSED101. Programming & Problem solving

## Spring, 2018

### Programming Assignment #5 (70 points)

여진영 (jinyeo@gmail.com)

■ **Due:** 2018.05.31 23:59

■ **Development Environment:** Windows Visual Studio 2017

■ **제출물**

- **C Code files (\*.c, \*.h)**
  - 제출시, 모든 파일을 하나의 파일로 압축해서 제출할 것. (압축파일명: asn5.zip)
  - 프로그램의 소스 코드를 이해하기 쉽도록 반드시 주석을 붙일 것.
- **보고서 파일** (.doc(x) or .hwp) 예) asn5.doc(x) 또는 asn5.hwp
  - AssnReadMe.pdf 를 참조하여 작성할 것.
  - 프로그램 실행화면 캡처하여 보고서에 포함시키고 간단히 설명 할 것!!
  - 명예서약(Honor code): 표지에 다음의 내용을 포함한다. "나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다." 보고서 표지에 명예서약이 없는 경우는 과제를 제출하지 않은 것으로 처리한다.
  - 소스코드와 보고서 파일을 LMS를 이용하여 제출한다.

■ **주의사항**

- 문제의 요구사항을 반드시 지킬 것.
- 에러 메시지를 포함하여 모든 문제의 출력 형식은 공백의 개수를 제외하고 아래의 예시들과 완전히 동일해야 하며, 같지 않을 시는 감점이 된다.
- 이번 과제를 작성하는 데 있어서 전역변수를 사용할 수 없다.
- 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.)
- 이번 과제에서는 추가 기능 구현에 대한 추가 점수는 없습니다.
- 이번 과제는 구조체와 연결리스트를 활용하는 것이 목표이므로, 문제에 구조체와 연결리스트를 언급한 부분을 배열을 통해서 해결할 경우 감점 처리함.

## ■ 학급 관리 프로그램 (Class Management System)

### (문제)

파일로부터 학생 데이터를 읽어서 학급 관리하는 프로그램을 작성한다.

### (목적)

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

### (소개)

학급 관리 프로그램은 학생들의 이름, 키, 성적 등과 같은 정보가 담겨 있는 텍스트 파일을 유지/관리가 편리한 연결리스트 형태로 저장하는 프로그램입니다. 이때 연결리스트를 활용하는 이유는, 새로운 학생이 전학을 오거나 기존의 학생이 전학을 갈 때 등 학급에서 발생하는 변화점들을 연결리스트의 추가 및 삭제 기능으로 관리하기가 편리하기 때문입니다. 본 assignment에서는 연결리스트의 기본적인 기능들을 활용하는 법을 학급 관리 프로그램을 통해서 익히고 훈련하는 것을 목표로 합니다. **[경고: 본 문서를 끝까지 꼼꼼히 다 읽고 구현을 시작할 것]**

### (주의사항)

- 학급 관리 프로그램은 6개의 명령어(show, add, remove, exam, sort, quit)를 입력 받아 각 기능을 수행하게 된다. 최소한 각 명령어 별로 함수를 정의하여 사용한다. 즉 명령어 외에 필요한 함수는 정의하여 사용할 수 있다.
- 이번 과제는 여러 개의 **파일로 분할**해서 작성한다.
  - ※ 헤더 파일 작성은 마지막페이지의 "헤더 파일 작성"부분을 참고한다(추가 점수 있음).
  - **학생 정보 (student.h, student.c):** 이름, 성별, 키, 이메일 등의 정보를 가진 학생 정보를 구조체로 선언하고, 학생 정보와 관련된 함수들(학생 정보 입력 등)이 선언 및 정의되어야 한다.
  - **학급 리스트 (class.h, class.c):** 학급은 연결리스트 구조로 선언하고, 연결리스트의 삽입, 삭제 등의 함수들이 선언 및 정의되어야 한다.
  - **assn5.c**
    - main()함수를 포함하여 필요한 함수들 정의
    - main() 함수 내에서는 아래와 같이 사용자로부터 명령어를 입력 받아 처리한다.

Ex)

```
int main()
{
    ...
    if (show 명령어)
        show();
    else if (add 명령어)
        add();
    ...
}
```

- Assignment 문서에 명시된 경우를 제외하고 다른 예외처리를 할 필요는 없음.

## (설명)

- 프로그램 구현은 구조체와 연결리스트를 사용하여야 하며, 각 항목에 대한 제약은 다음과 같다.
  - **이름(name):** 영어 최대 20글자
  - **성별(gender):** 'm'은 남성, 'f'는 여성(영문 1글자)
  - **키(height):** 양의 정수(단위: cm)
  - **이메일(email):** 최대 30글자로 이루어진 string으로, 모든 학생들이 제각기 다른 이메일주소를 가지고 있으므로 식별자(identifier)로 활용한다. (이 문자열은 공백이 없다고 가정한다.)

- 윈도우의 커맨드 라인에서 다음과 같이 입력하여, 프로그램을 실행한다. 노란색으로 표시된 문자는 사용자 입력에 해당한다.

```
C:W>assn5.exe students.txt
```

- 파일명을 입력하지 않은 경우, default로 "students.txt"로부터 데이터를 읽는다.
- 파일명이 입력된 경우, 입력된 파일로부터 데이터를 읽는다. 파일명의 최대 길이는 30자이며, 파일명에는 공백이 없다고 가정한다.
- 예외처리)  
읽을 파일이 존재하지 않는 경우: 아래의 메시지 출력 후, 프로그램을 종료한다.

```
Could not open input file!
```

```
계속하려면 아무 키나 누르십시오 . . .
```

- 프로그램 실행 시, 텍스트 파일(students.txt)로부터 학생들에 대한 데이터를 읽으며, 텍스트 파일의 내용 및 구성은 아래와 같다.

Lee Chulhyung	m	179	chlee@gmail.com
Lee Moonhyung	m	179	mhlee@gmail.com
Yeo Jinyoung	m	180	jinyeo@gmail.com
Lee Woongsae	m	173	wslee@gmail.com
Choi Youjin	f	159	yjchoi@gmail.com
Oh Youjeong	f	163	sudal@gmail.com

- 이름, 성별, 키, 이메일로 구성되어 있으며 각 항목 사이는 tab(\t)으로 구분된다.
- 한 사람의 정보와 정보 사이는 줄 바꿈 문자(\n)으로 구분된다.  
Ex. [Name][\t][Gender][\t][Height][\t][Email][\n]
- 프로그램을 실행시키면 텍스트 파일로부터 데이터를 읽어 목록을 연결리스트에 저장하고, 전체 학생 목록을 연결리스트에 저장되어있는 순서대로, 아래의 예제처럼 출력한다.
- 전체 학생 목록을 출력한 후에는 명령어를 입력 받을 준비를 한다.

Name	Gender	Height	Email
Lee Chulhyung	m	179	chlee@gmail.com
Lee Moonhyung	m	179	mhlee@gmail.com
Yeo Jinyoung	m	180	jinyeo@gmail.com

Lee Woongsae	m	173	wslee@gmail.com
Choi Youjin	f	159	yjchoi@gmail.com
Oh Youjeong	f	163	sudal@gmail.com

>

## (기능)

- 학급 관리 프로그램은 아래와 같은 기능을 가진다.
- 명시된 6가지 명령어 외의 명령어가 입력될 경우 무시하고 다시 명령어 입력을 대기한다. 이 때, 6개의 명령어(show, add, remove, exam, sort, quit) 중 하나를 입력한다. 사용자가 명령어 입력 시, 대소문자를 구분하지 않고 동일한 명령어의 기능을 수행하도록 작성한다. 예를 들면, show, Show, SHOW는 동일한 동작을 수행한다. (단, 명령어는 최대 10글자이다.

## 1. SHOW 명령어

- show 입력 시, 전체 학생들의 정보를 출력시켜준다. 이때 출력되는 순서는 연결리스트의 head부터 저장되어있는 순서이다.

>show

Name	Gender	Height	Email
Lee Chulhyung	m	179	chlee@gmail.com
Lee Moonhyung	m	179	mhlee@gmail.com
Yeo Jinyoung	m	180	jinyeo@gmail.com
Lee Woongsae	m	173	wslee@gmail.com
Choi Youjin	f	159	yjchoi@gmail.com
Oh Youjeong	f	163	sudal@gmail.com

>

## 2. ADD 명령어

- 새로운 학생이 전학을 온 경우 등에 사용.
- add 입력 시, 아래 예제와 같이 본인의 이메일, 이름, 성별, 키를 순서대로 입력 받아 새 연락처를 추가한다 (키는 실제 수치를 입력할 필요는 없습니다). 새로운 학생은 리스트의 마지막에 추가 되도록 구현한다.
- show 입력 시, 새 학생의 정보가 포함되어 출력되어야 합니다.

>add

Email: gildong@gmail.com  
 Name: Hong Gildong  
 Gender: m  
 Height: 182  
 Added!

>show

Name	Gender	Height	Email
Lee Chulhyung	m	179	chlee@gmail.com
Lee Moonhyung	m	179	mhlee@gmail.com
Yeo Jinyoung	m	180	jinyeo@gmail.com
Lee Woongsae	m	173	wslee@gmail.com
Choi Youjin	f	159	yjchoi@gmail.com
Oh Youjeong	f	163	sudal@gmail.com
Hong Gildong	m	182	gildong@gmail.com

>

- 예외처리)

이미 있는 학생을 입력한 경우, 이미 있다는 메시지와 함께 정보를 출력해 준 후 명령어 입력을 기다린다. 아래의 예시를 참고한다.

>add

Email: jinyeo@gmail.com

Already Exists!

Name	Gender	Height	Email
Yeo Jinyoung	m	180	jinyeo@gmail.com

>

### 3. REMOVE 명령어

- 학생이 다른 학교로 전학을 간 경우 등에 사용.
- remove 입력 시, 삭제하고자 하는 이메일을 입력 받아 해당하는 학생 정보를 삭제한다.
- 예외처리)

아래의 예제는 목록에 없는 이메일을 입력한 경우의 예시이다.

>remove

Email: jinyoung@gmail.com

No Email!

>

- 아래의 예제는 목록에 있는 학생이 삭제된 예시로, show 입력 시 삭제된 학생 정보가 제외되어 출력된다.

>remove

Email: jinyeo@gmail.com

Removed!

>show

Name	Gender	Height	Email
Lee Chulhyung	m	179	chlee@gmail.com
Lee Moonhyung	m	179	mhlee@gmail.com
Lee Woongsae	m	173	wslee@gmail.com
Choi Youjin	f	159	yjchoi@gmail.com
Oh Youjeong	f	163	sudal@gmail.com
Hong Gildong	m	182	gildong@gmail.com

>

#### 4. EXAM 명령어

- 중간/기말시험을 치른 후, 아래와 같은 파일(exam\_result.txt)에 성적을 정리했다.  
전학년 학생의 시험 성적도 있을 수 있으며, 시험 성적이 없는 학생도 있을 수 있다.  
※ 시험 성적이 없는 학생: 학생 목록에는 있지만, 성적 파일(exam\_result.txt)에는 없는 학생을 말함
- exam\_result.txt 파일에 본인의 이름, 이메일과 함께 시험 성적(0~100점 사이)을 입력해야 한다. (이것은 C가 아니라 텍스트파일을 직접 수정하여 작성합니다. 가상의 점수를 넣으세요.)  
"exam\_result.txt"의 구성과 예시는 다음과 같다.

Hong Gildong	92	94	gildong@gmail.com
Lee Chulhyung	68	73	chlee@gmail.com
Lee Moonhyung	65	71	mhlee@gmail.com
Yeo Jinyoung	100	100	jinyeo@gmail.com
Lee Woongsae	82	84	wslee@gmail.com
Choi Youjin	45	47	yjchoi@gmail.com
Oh Youjeong	78	63	sudal@gmail.com

←여기 직접 입력하세요

- 이름, 중간고사 성적, 기말고사 성적, 이메일로 구성되어 있으며 각 항목 사이는 tab(Wt)으로 구분된다.
- 한 사람의 정보와 정보 사이는 줄 바꿈 문자(Wn)으로 구분된다.  
Ex. [Name][Wt][MidTerm\_Score][Wt][FinalTerm\_Score][Wt][Email][Wn]
- "exam\_result.txt"라는 파일과 함께 exam 입력 시, 학생들의 중간/기말 시험 평균 성적을 저장하여야 하고, 시험 평균 성적을 기준으로 내림차순으로 전체 학생 목록을 아래와 같이 출력한다. (추가된 여러분의 정보도 함께 포함 되어있어야 합니다.)  
시험 성적이 같은 학생의 경우, 이름을 기준으로 알파벳순으로 정렬한다. (동일한 이름이 존재하는 경우에는 누가 먼저 정렬되어도 상관없다.)
- (성적 처리 파일 생성)  
학생들의 평균성적은 "exam\_grade.txt"파일을 새로 만들어 아래와 같은 형태로 평균

점수가 추가되어 저장 되어야한다.

[Name][Wt][MidTerm\_Score][Wt][FinalTerm\_Score][Wt][Average\_Score][Wt][Email][Wn]

- 성적 파일에 시험 성적이 없는 학생의 경우 0점으로 처리하면 된다.
- 학생 목록에 없는 학생의 시험 성적은 무시한다.
- (힌트) Student 구조체에 score를 저장하는 부분을 추가하여 평균점수를 저장

```
>exam exam_result.txt
```

Name	Gender	Height	Score	Email
Hong Gildong	m	182	93.0	gildong@gmail.com
Lee Woongsae	m	173	83.0	wslee@gmail.com
Lee Chulhyung	m	179	70.5	chlee@gmail.com
Oh Youjeong	f	163	70.5	sudal@gmail.com
Lee Moonhyung	m	179	68.0	mhlee@gmail.com
Choi Youjin	f	159	46.0	yjchoi@gmail.com

```
>show
```

Name	Gender	Height	Email
Hong Gildong	m	182	gildong@gmail.com
Lee Woongsae	m	173	wslee@gmail.com
Lee Chulhyung	m	179	chlee@gmail.com
Oh Youjeong	f	163	sudal@gmail.com
Lee Moonhyung	m	179	mhlee@gmail.com
Choi Youjin	f	159	yjchoi@gmail.com

```
>
```

- 단순히 정렬하여 출력하는 것이 아니라 연결리스트 상에서 순서가 변했기 때문에 show 입력 시, 위의 같은 결과가 출력된다.
- 예외처리)
  - 파일명 없이 단순히 exam만 입력한 경우, 명령문을 무시한 뒤 다시 명령어 입력을 대기한다.
  - 읽을 파일(exam.txt)이 존재하지 않는 경우, 아래의 예시처럼 메시지 출력 후, 다시 명령어 입력을 대기한다.

```
>exam exam.txt
```

```
Could not open input file!
```

```
>
```

## 5. SORT 명령어

- 학습 효율이 좋아지도록 학생들의 자리를 배치하려고 한다.
- 정렬되는 학생은 키 순으로 (작은 학생부터 먼저, 큰 학생을 뒤에) 연결리스트 상에 존재하되, 여학생과 남학생이 번갈아가며 정렬되도록 한다. 만약 남녀의 수가 달라 더 이상 번갈아가며 배치할 수 없는 경우 키 순으로 정렬한다. 만약 키가 같은 경우는 이름을 기준으로 알파벳 순으로 정렬한다.  
(동일한 이름이 존재하는 경우에는 누가 먼저 배치되어도 상관없다.)
- show 명령 없이도, 정렬된 학생 목록을 아래와 같이 출력한다.

```
>sort
Name      Gender Height Email
Choi Youjin      f      159  yjchoi@gmail.com
Lee Woongsae     m      173  wslee@gmail.com
Oh Youjeong      f      163  sudal@gmail.com
Lee Chulhyung    m      179  chlee@gmail.com
Lee Moonhyung    m      179  mhlee@gmail.com
Hong Gildong     m      182  gildong@gmail.com

>show
Name      Gender Height Email
Choi Youjin      f      159  yjchoi@gmail.com
Lee Woongsae     m      173  wslee@gmail.com
Oh Youjeong      f      163  sudal@gmail.com
Lee Chulhyung    m      179  chlee@gmail.com
Lee Moonhyung    m      179  mhlee@gmail.com
Hong Gildong     m      182  gildong@gmail.com

>
```

- 단순히 정렬하여 출력하는 것이 아니라 연결리스트 상에서 순서가 변했기 때문에 show 입력 시 위와 같은 결과가 출력된다.

## 6. QUIT 명령어

- quit 입력 시, 프로그램을 종료한다.
- 해당 명령어를 실행할 때, 프로그램 실행 시, 읽었던 텍스트 파일의 이름을 저장한다.  
앞서 본 "students.txt"와 같이 내용을 구성한다.  
**[Name][Wt][Gender][Wt][Height][Wt][Email][Wn]**
- 저장할 때 목록의 순서는 현재 연결리스트 상의 순서와 동일하게 저장하도록 한다.
- 동적 할당된 memory를 모두 free 시킨 후, 종료한다.

```
>quit
Terminated!
계속하려면 아무키나 누르십시오...
```



## (헤더 파일 작성)

### 1. 헤더 파일 작성 예시 (student.h)

```
#ifndef STUDENT_H
#define STUDENT_H

// 구조체 정의
....
// 함수 정의
....

#endif
```

위의 사용자 정의 헤더 파일 student.h를 필요한 곳에서 include 하여 사용한다. 보고서에 왜 "#ifndef"와 "#define", "#endif"와 같은 부분이 들어가는지 간략하게 조사하여 그 이유를 서술한다.(추가점수 있음)