2019년 겨울 1주차 과제

성적 관리 프로그램

Assignment #1

**담당교수: 윤은영**

**번호: 050223**

**과정: 청년 AI – Big data 5기**

**이름: 최진호**

**LMS ID: s\_0258**

명예서약(Honor code)

“나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.”

Problem 1: 성적 관리 프로그램 만들기

1. 문제의 개요

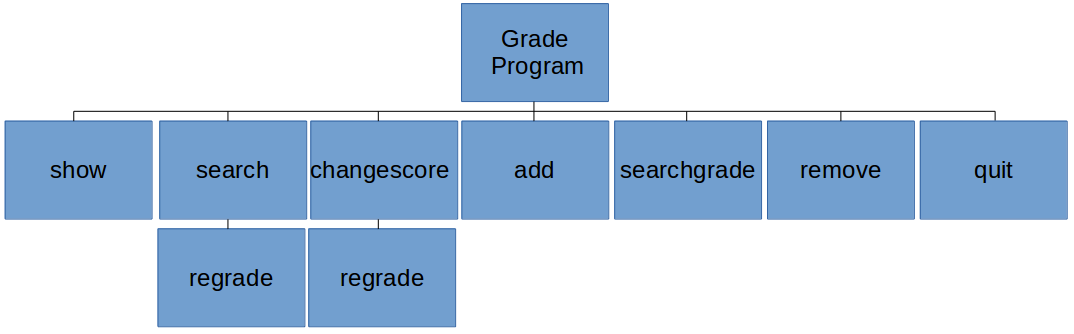
본 프로그램을 간략히 설명하면 다음과 같다.

- 입력받은 학생 성적 파일을 읽어 평균과 학점을 산출하고 성적표 양식으로 출력한다.

- 필요에 따라 데이터 추가, 제거, 변경 등을 통해 성적표를 수정하고 용이하게 만든다.

- 모든 수정이 마무리된 후에 현재 저장된 상황을 파일로 저장시킨다.

이 때 사용되는 구상 가능한 구조 차트(structure chart)는 아래와 같이 표현될 수 있다.



- 입력부: 사용자가 입력하는 파일명에서 학생 정보를 불러오고 각 텍스트 줄을 리스트 형태로 저장한다.

- 처리부: 점수 출력, 학생 추가, 점수 변경, 학생 삭제, 학점 점검 등 다양한 기능을 시행한다.

출력(show)으로 현재 저장된 리스트 상황을 확인하고 출력한다.

- 출력부: 처리가 완료된 현재 학생들의 명단을 다시 원래 형태의 텍스트 파일로 저장한다.

2. 알고리즘

본 프로그램 작성을 위한 알고리즘을 Pseudo 코드 형태로 나타내면 다음과 같다.

Pseudo-algorithm for grade program

// 프로그램에 필요한 변수들은 미리 선언해놓은 것으로 가정한다.

1. 포함 sys

2. 입력한 파일명 읽기, 공백시 students.txt를 디폴트로 엶

3. 빈 리스트d를 만들고 성과 이름을 합쳐서 하나의 원소로 만듦

4. show 함수를 호출해 현재 리스트 d(초기 파일)의 평균을 계산하고 학점을 부여

평균 기준 내림차순 순서로 소팅, 성적표 형태(평균과 학점이 기재)를 출력

5. while 무한루프에서 명령어를 대기

---아래는 while루프 안의 명령어(함수)의 로직-- 6.~12.는 함수 종료후에도 다시 명령어를 대기

6. show 함수 호출 시, 현재 리스트 d를 평균 기준 내림차순 순서로 소팅 후, 성적표 형태로 출력

7. search 함수 호출 시, 학번을 입력받아 해당 학생이 있으면 해당 학생의 성적표 형태를 출력

만일 없으면NO SUCH PERSON.을 출력

8. regrade 함수 호출 시, 실수를 입력받아 각 학점 경계 점수대와 비교해 해당 학점을 반환

9. changescore 함수 호출 시, 학번을 입력받아 해당 학생이 있으면 언제 시험인 지와 새 점수를 입력

성적표 양식을 출력하고, 새로 평균과 학점을 재산정하는 regrade함수를 호출

해당 학번의 학생이 없으면 NO SUCH PERSON.을 출력

10. add 함수 호출 시, 학번을 입력받아 해당 학생이 있으면 ALREADY EXISTS.를 출력

만일 없으면 빈 리스트k에 학생의 이름, 중간점수, 기말점수를 입력하고 저장, 이 k를 다시 리스트d에 추가

11. searchgrade 함수 호출 시, 검색할 학점(문자) 입력, 해당 학점의 학생들을 성적표 형태로 출력

만일 해당 학점의 학생이 없으면 NO RESULTS.을 출력

12. remove 함수 호출 시, 해당 학번의 모든 정보(이름, 점수, 평균, 학점)를 제거

만일 현재 리스트 상태가 비어있으면 List is empty.를 출력

만일 해당 학번의 학생이 없으면 NO SUCH PERSON.을 출력

13. quit 함수 호출 시, yes or no를 물어보고 해당 문자열이 아니면 다시 물어봄

yes 선택 시, 저장할 파일명을 입력받고 현재 리스트d를 초기 파일형태로 4가지 정보

(학번, 이름, 중간점수, 기말점수)만 저장시키고 $를 출력후 프로그램 종료

no 선택 시, 저장하지 않고 $를 출력 후 프로그램 종료

quit을 어떤 형식으로 종료하든 break를 연속으로 시행해서 무한루프 벗어남

위의 의사 알고리즘을 Flowchart를 통해 표현하면 아래와 같다.

FLOW CHART \_

3.프로그램 구조 및 설명

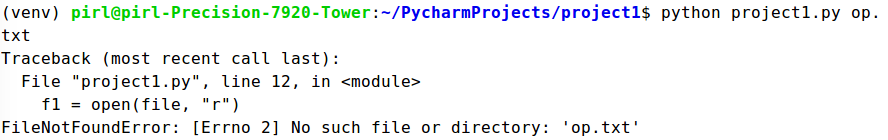
a) 파일명 입력

- 터미널에서 python project1.py “파일명”혹은 윈도우 프롬프트에서 project1.py “파일명”을 입력한다.

- sys.argv를 이용해 파일명을 입력 받는다.

- 만일 파일명이 공백이라면(sys.argv의 길이가 1이라면) 디폴트로 students.txt를 불러온다.

- 없는 파일명에 대해서는 이미 FileNotFoundError가 뜨므로 별도 처리하지 않는다.



b) 파일을 읽어 데이터로 저장, 정리, 추가

- file을 open을 통해 읽기모드로 열고 data에 모든 문자열을 입력 받는다.

- data를 카피해 k를 만들고, list\_d라는 빈 리스트를 만든다.

- k의 갯수만큼 각 행에 대해서 공백을 기준으로 split해서 1행을 5개의 원소를 분할시키고 list\_d에 추가시킨다.

- for i in list\_d:에서 list\_d의 갯수만큼 성과 이름이 분리된 것을 띄어쓰기를 넣어 한 문자열로 통합시킨 후,

원래 이름의 원소(i[2])를 삭제한다.

- i[2]에는 중간고사 점수가 들어가게 되고, i[3]의 기말점수와 평균을 내서 리스트에 i[4]로 추가시킨다.

이 과정에서 원래 i의 원소인 str에 대해 +와 / 연산이 정상 작동하지 않으므로, float로 형변환을 한 후,

실수로 추가시킨다.

c) 함수 정의

c-1) show : input도 없고 output도 없다. (조건에 따라 print형태의 출력만 있다)

- for i in list\_d:에서list\_d의 수만큼 i[4]에 i[2]와 i[3]의 평균을 넣는다. 이 과정에서 원래 i의 원소가 str일 수 있다. str에 대해 +와 / 연산이 정상 작동하지 않으므로, float로 형변환을 한 후, 실수로 추가시킨다.

- i[4]를 학점 경계에 대해 비교해서 점수에 맞는 학점을 i[5]에 추가시킨다

- list\_d.sort(key=**lambda** e: e[4], reverse=**True**)의 기능 :

list\_d의 리스트에 대해서 소팅한다.

4번째 원소인 평균을 기준으로 reverse=True이므로 내림차순으로 정렬한다.

해당 코드의 결과 list\_d가 평균을 기준으로 내림차순으로 정렬이 완료된다.

**람다함수란 ? lambda 인자 : 표현식 형태로 표현한다.**

람다형식은 인공지능 분야나 AutoCAD라는 설계프로그램에서 쓰이는Lisp언어에서 물려받아 이름이 없다.

함수의 이름이 없어 다른 함수에서 쓰일 때 코드가 간결해진다.

여러 인자를 받을 수 있으나, 표현식은 단 1개이다.

(출처 : <https://www.w3schools.com/python/python_lambda.asp>, <https://wikidocs.net/64>)

- print로 list\_d의 원소명들을 각각 적어주고 밑줄을 인쇄한다. (성적표 양식 형태 출력)

- for i in range(len(list\_d)):에서 list\_d의 행 갯수만큼 (현재 있는 학생수만큼) 각 행의 원소를 출력한다.

- %14s로 오른쪽 기준 14칸, 공백 4칸을 부여해 이름에 해당하는 칸 만큼 여유있게 출력한다.

- %3s로 0~100점을 받을 수 있게 만든다. %5.1lf로 0.0~100.0까지 받을 수 있게 만든다.

c-2) search : input도 없고 output도 없다. (조건에 따라 print형태의 출력만 있다)

- input\_id로 학번을 입력받는다. 동시에 임시 카운팅변수 temp=0을 만든다.

- for i in list\_d:에서 만일 input\_id가 각 행 i에 대해 존재한다면, 해당 학생의 성적표 양식 형태로 출력한다.

- 이런 출력이 이루어지면 temp=1이 되고, NO SUCH PERSON을 출력하지 않는다.

- 만일, 해당 학생이 각 행 i에 대해 없다면 여전히 temp=0이 되므로 NO SUCH PERSON.을 출력한다.

c-3) regrade : input은 실수, output은 해당 학점(문자열)

- 문제에서 요구하는 기능은 아니나, changescore시 평균이 바뀔 수 있고 동시에 학점도 변할 수 있다.

- 따라서 실수를 인자로 받고 해당하는 학점을 반환하는 함수를 만들어 코드의 간결성과 가독성을 높였다.

- 변환된 평균(실수)을 대입하고 changescore내부에서만 쓰이므로 다른 데이터 형태나 범위를 벗어난 값이 들어

갈 오류가 없다.

c-4) changescore : input도 없고 output도 없다. (조건에 따라 print형태의 출력만 있다)

- input\_id로 학번을 입력받는다. 동시에 임시 카운팅변수 temp=0을 만든다.

- for i in list\_d:에서 만일 input\_id가 각 행 i에 대해 존재한다면, temp에 1을 대입한다.

- 동시에 input\_term으로 mid나 final을 입력받는다. 그 외의 값은 실행시키지 않는다.

- 0이상 100이하의 input\_score를 입력받았다면, 성적표 양식을 출력하고, 해당 학생의 input\_term의

시험점수(mid면 i[2], final이면 i[3]를 수정한다.)

- 변한 점수로 평균을 재산출하고 regrade함수로 i[5]에 대입해 학점도 재산출한다.

- 점수가 변했음을 알리고, 수정된 점수, 수정된 평균, 수정된 학점으로 수정된 정보로 출력한다.

- 학번이 리스트에 없다면 temp가 0이므로 NO SUCH PERSON.을 출력한다.

c-5) add : input은 없고 조건에 따라 함수 종료 목적의 return output이 있다. (조건에 따라 print형태의 출력만 있다)

- input\_id로 학번을 입력받는다. 동시에 임시 카운팅변수 temp=0을 만든다.

- for i in list\_d:에서 만일 input\_id가 각 행 i에 대해 존재한다면, temp에 1을 대입한다.

- 만일 temp가 1이면, 이미 학생이 존재한다는 뜻이므로 ALREADY EXISTS.를 출력하고 return으로 함수

를 종료한다.

- temp가 0이라면, 즉 리스트에 해당 학번이 없다면 5개의 원소를 갖는 k를 만든다.

- 순서대로 학번, 이름, 중간점수, 기말점수를 입력한다.

- 평균을 계산해 k[3]에 입력하고, regrade를 사용해 k[5]에 학점을 대입한다.

- 이를 원래 학생 리스트 list\_d에 추가시키고 학생이 추가되었음을 알린다.

c-6) searchgrade : input은 없고 조건에 따라 함수 종료 목적의 return output이 있다. (조건에 따라 print형태의 출력만 있다)

- input\_grade로 검색할 학점을 입력받는다. 동시에 임시 카운팅변수 temp=0을 만든다.

- 빈 new\_list를 만들고, ABCDF가 순서대로 입력된 gra\_list를 만든다.

- if input\_grade in gra\_list:에서 입력한 학점이 ABCDF중 하나면 검색을 시행한다.

- for i in list\_d:에서 만일 각 행 i에 대해 입력한 학점이 해당 학생의 학점 i[5]와 같다면 new\_list에 i를

추가시키고 temp에 1을 대입한다.

- 만일 input\_grade가 gra\_list에 없으면 return으로 함수를 종료한다.

- 함수가 종료되지 않았다면, 즉 ABCDF중 하나를 입력했고 해당 학점의 학생이 있다면 temp가1이 되고

해당 학생들의 성적표 양식 형태를 출력한다.

- 만일 ABCDF중 하나를 입력했지만, 해당 학점의 학생이 없다면 temp는 0이므로 NO RESULTS.를 출력한

다.

c-7) remove : input은 없고 조건에 따라 함수 종료 목적의 return output이 있다. (조건에 따라 print형태의 출력만 있다)

- list\_d의 길이가 0이라면 List is empty.를 출력하고 return으로 함수를 종료한다.

- 만일 함수가 종료되지 않았다면 input\_id로 학번을 입력받고, 동시에 임시 카운팅변수 temp=0을 만든다.

- for i in list\_d: 에서 만일 input\_id가 i에 있다면, list\_d에서 i행을 삭제하고 학생이 제거되었음을 알리고

temp에 1을 대입하고 해당 for 문을 나간다.

- temp가 1이면 밑의 if문장을 시행하지 않는다.

- temp가 0이면, 즉 리스트의 크기가 0이 아니라 학생리스트가 있었지만 해당 학번의 학생이 리스트에 없다면

temp가 0이므로 NO SUCH PERSON.을 출력한다.

c-8) quit : input은 없고 함수 종료 목적의 return output이 있다. 현재의 리스트를 새로운 파일에 저장하고 프로그램을 종료한다.

- while 1: 무한 루프에서 input\_yon을 받는다.

- 문제에서 요구하지는 않았지만, 대소문자와 상관없이 모두 소문자로 변환해 yes와 no에 대응시킨다.

(yes나 no의 의도가 있다고 판단)

- 만일 yes라면 파일명을 물어보고 그 문자열대로 파일을 g=open(파일명, “w”) 쓰기 모드로 연다.

- 해당 리스트대로 학번, 이름, 중간점수, 기말점수만 취해서 새로운 변수(girok)에 입력시키고

- 이를 g에 쓰고 행 끝에 개행문자를 추가시킨다.

- 모든 행의 입력이 끝났으면 $를 출력하고 g를 닫고 return으로 quit함수를 종료한다.

- no를 입력했다면, $만 출력 후 return해 quit함수를 종료한다.

- 만일 yes나 no가 아니라면, input\_yon을 계속해서 입력받는다.

(프로그램을 종료시키는 함수이므로 실수로 입력했더라도 yes로 저장해서 데이터를 지키는 것을 의도)

----- 함수 정의 끝-----

d) 초기 학생(파일) 상태 출력

- show를 프로그램 실행 후 기본적으로 한 번 시행한다.

- 현재 읽은 파일의 리스트에서 평균과 학점을 계산하고, 평균 기준 내림차순으로 출력한다.

c-1의 show 참조.

e) while 1 : 무한루프 안에서 명령어(함수)를 실행시킬 준비를 하고 실행한다.

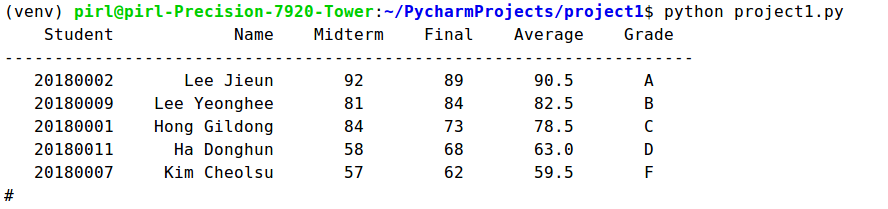
- #뒤에 명령어를 대기한다. 동시에 이를 소문자로 모두 치환시킨다.

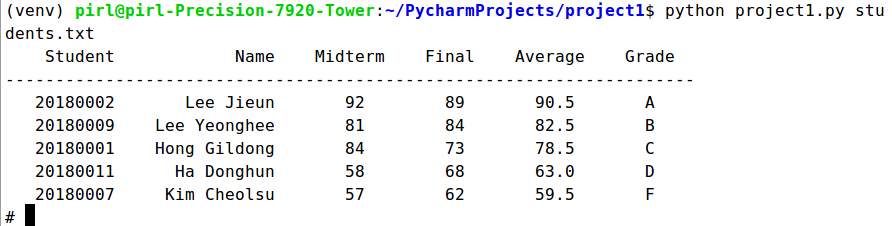
- 해당 소문자를 각각의 함수명과 비교해 일치하면 해당 함수를 실행시킨다

- quit은 실행을 한 후에 모두 return으로 종료한다. 그 뒤에 break실행으로 무한루프를 탈출한다.

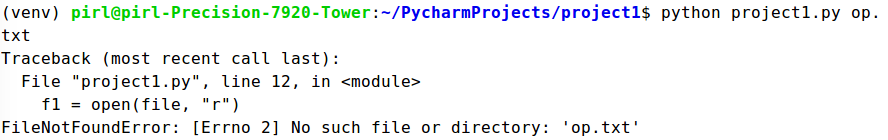
- 만일 일치하는 명령어가 없으면 다시 #뒤에 명령어를 대기 받는다.

4.프로그램 실행방법 및 예제

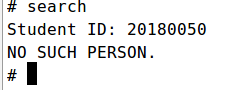
• python 환경에서 파일명을 입력하지 않아도 디폴트인 students.txt를 받아와서 내림차순 정렬 출력



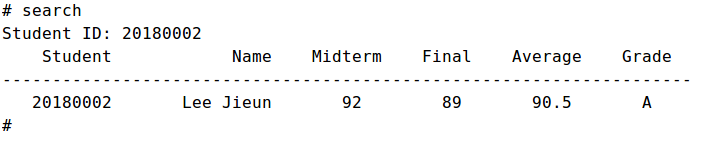
•students.txt를 입력했을 때와 동일 결과

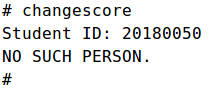
• 존재하지 않는 파일을 열려고 하면 에러가 뜨면서 프로그램 종료

정상적으로 파일이 열리면 #로 명령어 대기 모드로 들어간다.

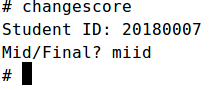


• search시 없는 학번이면 없다고 출력

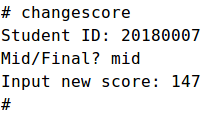
• search시 있는 학번이면 해당 학생의 성적표 양식 형태 출력



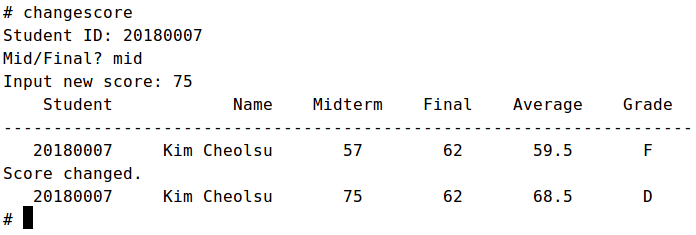
• changescore시 없는 학번이면 없다고 출력

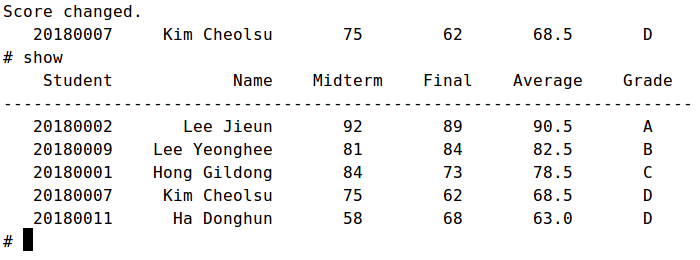


•changescore시 있는 학번이면 mid/fianl 중 하나 입력하라고 출력, miid라고 잘못된 문자 입력시 # 명령어 대기 모드

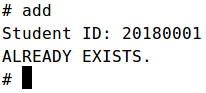


•0과 100을 벗어난 범위의 점수를 입력해도 # 명령어 대기 모드

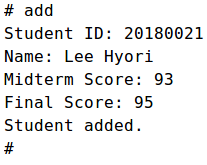
•학번, 시험일시, 점수범위가 모두 정상범주면 해당 학생의 현재 성적표 형태를 출력하고, 점수와 평균 학점이 아래와 같이 변했음을 알려줌

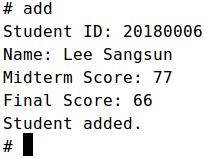


•show로 확인하면 바뀐 점수로 적용

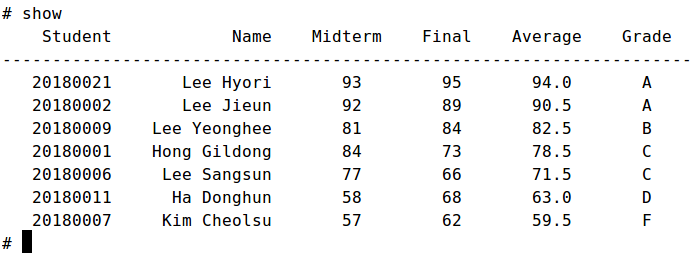


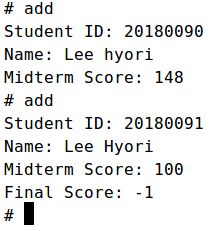
•add시, 학번이 있다면 이미 있다고 출력



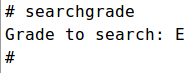


• 없는 학번이고 중간 점수와, 기말점수가 정상범주면 학생이 추가됐다고 알려줌

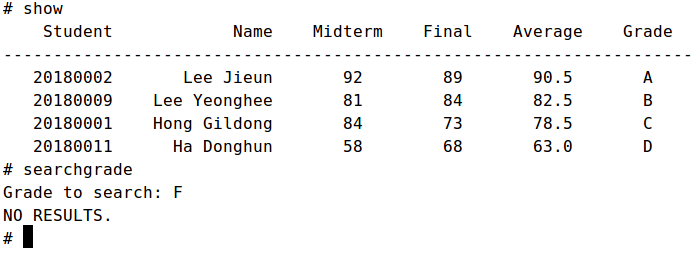
• 정상적으로 추가

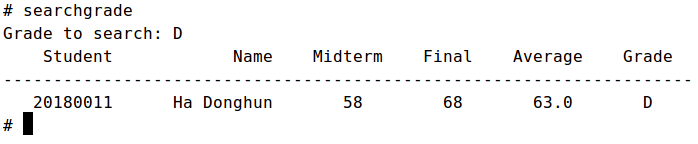


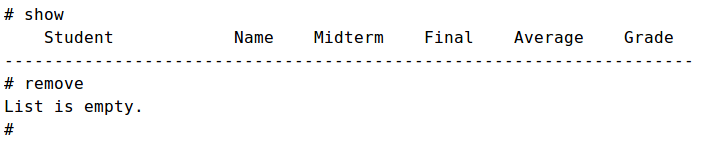
•점수가 정상범주를 벗어나면 그 상태로 add함수 종료하고 다시 #명령어 대기 모드 상태

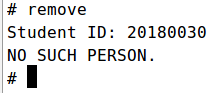


•searchgrade시 학점 문자가 아니면 #명령어 대기 모드 상태

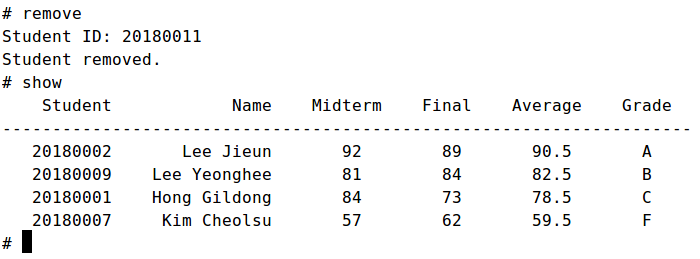
• 학점문자를 입력했으나 해당 학점의 학생이 없으면 결과가 없다고 출력

•해당 학점의 학생이 있으면 해당 학생들을 출력

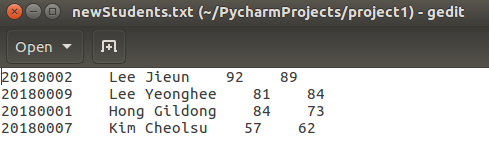
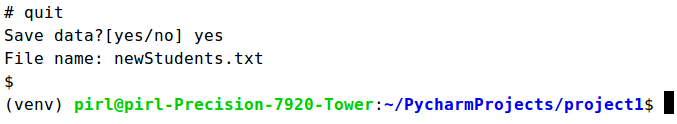
•remove시 빈 리스트면 리스트가 비었다고 출력



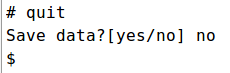
•해당 학번이 없으면 없다고 출력



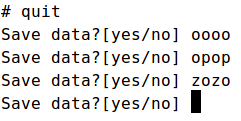
•해당 학번이 있으면 제거됐다고 출력하고, show로 확인시 정상 제거된 모습



• quit실행후 yes시 $출력후 프로그램 종료, 현재 리스트만 새로운 파일로 정상 저장된 모습



• no 입력시 $ 출력 후 정상 종료



• yes나 no가 아닐시 데이터를 저장할지 계속 물어

5.토론

-

문자의 ASCII 코드를 알아내는 프로그램을 작성할 때, scanf를 연속적으로 사용할 경우

정상적으로 동작하지 않는 문제가 발생하였다.

-

이는 문자를 입력 받을 때 사용한 ‘엔터’가 입력 버퍼에 남아있기 때문에 발생함을 알게

되었다.

-

문제를 해결하기 위해서는 ASCII 코드를 알아내는 프로그램을 필요한 만큼 반복 실행하

거나, flushing을 통해 입력버퍼를 비워주는 작업(fflush(stdin))을 매 scanf 앞에서 행하는

방식이 요구된다.

6.결론

본 과제에서는 prinf와 scanf를 사용하는 방법을 익히는데 유용했으며, ASCII 코드를 사용

하는 과정을 통해 문자 및 정수 변수의 관계를 학습할 수 있게 되었다.

7.개선방향

본 과제는 요구사항만 적절히 만족시켜주면 완성할 수 있기 때문에 프로그램 기능 자체

의 개선보다는 향후 학습하게 될 함수의 사용 또는 반복문/조건문 등을 통해 프로그램이

더욱 유동적이 되도록 할 수 있을 것으로 예상한다.

출처 : https://wikidocs.net/64