**POSCO 청년 AI/Big Data아카데미**

**성적 관리 프로그램**

**20기 C3조**

**성명: 신주연**

**이메일: yje0502@naver.com**

**Problem: 성적 관리 프로그램**

1.문제의 개요

본 프로그램을 간략히 설명하면 다음과 같다.

* 파일로부터 데이터(학번, 이름, 중간고사 점수, 기말고사 점수)를 읽어서 성적 목록을 작성한다.
* 이 프로그램은 사용자로부터 7개의 명령어(search , changescore , searchgrade, add, remove, quit)을 입력 받아 각 기능을 수행하게 된다.
* 목록을 기반으로 학번별 평균과 학점을 입력 받아 출력한다.

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

Show

Quit

Search

Remove

Searchgrade

Add

Changescore

Students

New Students

* 입력부: student.txt를 부르고 연다.
* 처리부: 사용자가 입력하는 7개의 명령어에 따라 실행한다.
* 출력부: 명령어 실행 후 결과 값을 반영하고 Newstudents.txt로 저장한다.

2.알고리즘

본 프로그램 작성을 위해 필요한 변수들의 정의는 다음과 같다.

* 학번(Student ID)

학생을 구분하는 고유한 식별자이다

* 이름

학생의 이름

* 중간고사 점수

학생의 중간고사 점수

* 기말고사 점수

학생의 기말고사 점수

* 평균

중간고사 점수와 기말고사 점수의 평균

* 학점

학점 부여 기준은 다음과 같다

A: 평균이 90점 이상

B: 평균이 80점 이상, 90점 미만

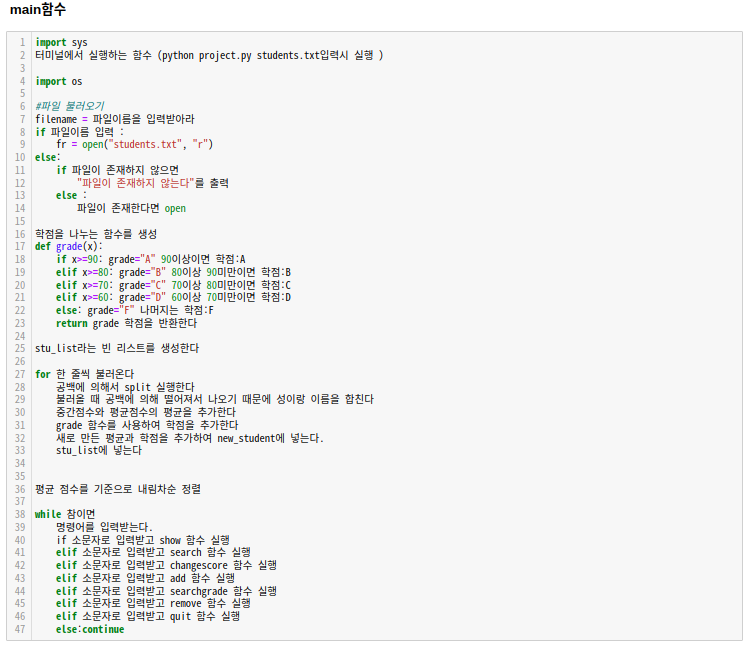
C: 평균이 70점 이상, 80점 미만

D: 평균이 60점 이상, 70점 미만

F: 평균이 60점 미만

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

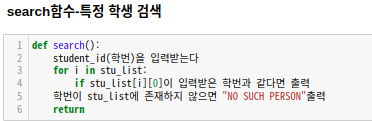
1. main함수



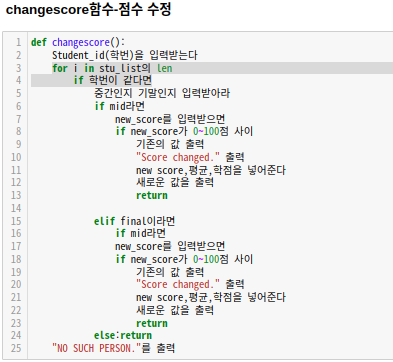
2. show 함수



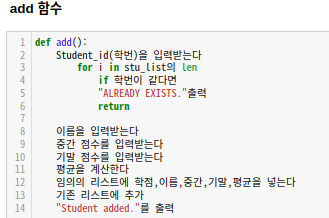
3. search 함수



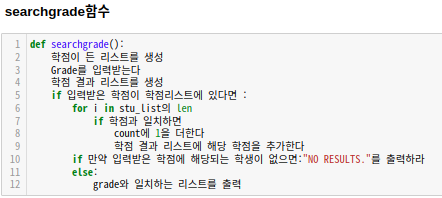
4. changescore 함수



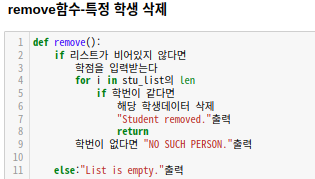
5. add함수



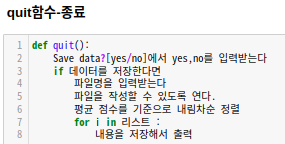
6. searchgrade 함수



7. remove함수



8. quit 함수



3. 프로그램 구조 및 설명

* show

show 입력 시, 저장되어 있는 전체 목록을 평균 점수를 기준으로 내림차순으로 출력한다

단, 평균 점수는 소수점 이하 첫째 자리까지만 표시한다

* search

검색하고자 하는 학생의 학번을 입력 받아 학번, 이름, 중간고사 점수, 기말고사 점수, 평균, 학점을 출력한다

단, 찾고자 하는 학생이 목록에 없는 경우에는 “NO SUCH PERSON.”이라는 에러 메세지를 출력한다

* changescore

수정하고자 하는 학생의 학번, 수정하고자 하는 점수가 중간고사인지 기말고사인지와 수정하고자 하는 점수를 순서대로 입력 받아 해당 학생의 점수와 학점을 수정한다

단, 학번이 목록에 없는 경우에는 “NO SUCH PERSON.”이라는 에러 메세지를 출력하며 “mid”또는 “final”외의 값이 입력된 경우와 점수에 0~100 외의 값이 입력된 경우에는 실행되지 않는다.

* Add

학생의 학번, 이름, 중간고사 점수, 기말고사 점수를 차례로 입력 받아 평균과 학점을 계산하여 추가하여 저장한다

학생 추가 후 show 명령을 사용하면 평균을 기준으로 내림차순으로 출력한다

단, 목록에 있는 학생의 학번을 입력 시, “ALREADY EXISTS.”이라는 에러 메세지를 출력한다

* searchgrade

특정 grade를 입력 받아 그 grade에 해당하는 학생을 모두 출력한다

단, A, B ,C ,D ,F외의 값이 입력된 경우 실행되지 않는다

해당 grade의 학생이 없는 경우 메시지 “NO RESULTS.”를 출력한다.

* REMOVE

삭제하고자 하는 학생의 학번을 입력 받은 후, 학생이 목록에 있는 경우 삭제한다

삭제하면, 메시지 “Student removed”를 출력한다

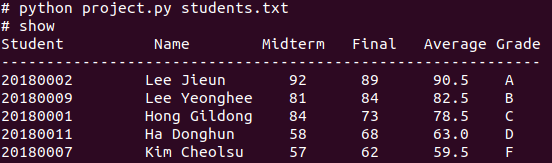
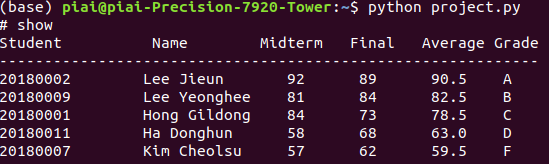
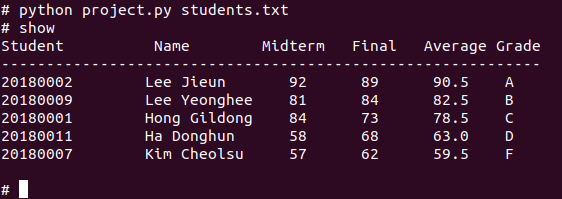
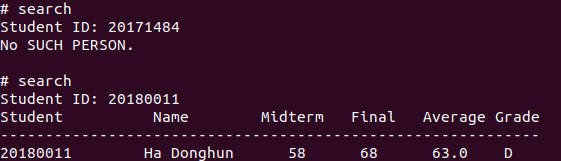
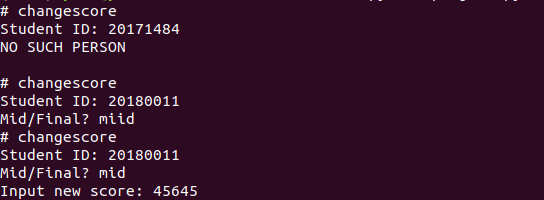
단, 목록에 아무도 없을 경우 “List is empty” 메세지를 출력한다

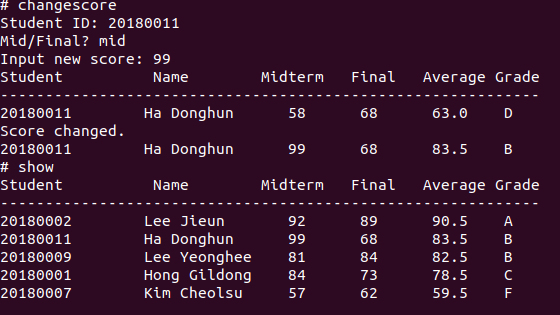
* quit

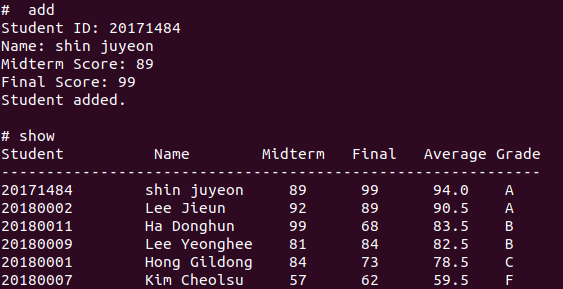
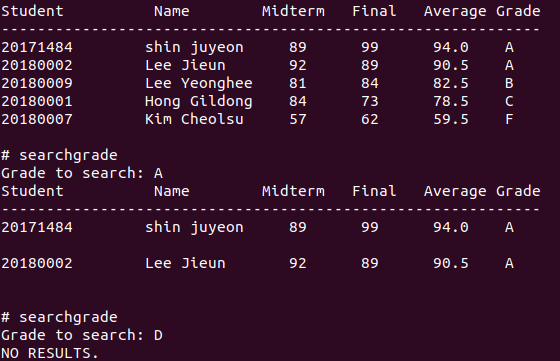
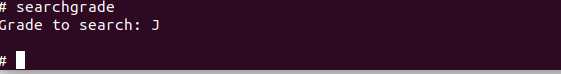
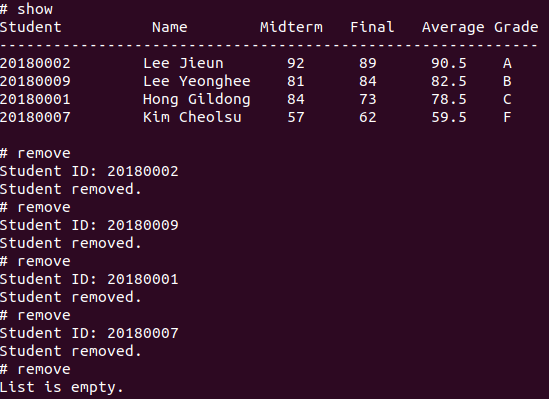
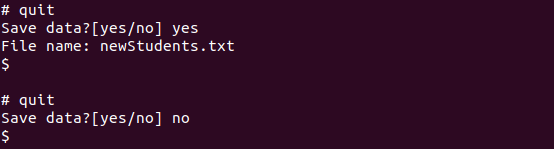
현재까지 편집한 내용의 저장 여부를 묻고, 저장을 선택(yes 입력)할 경우 파일명을 입력 받아서 저장한다

저장할 때 목록의 순서는 평균을 기준으로 내림차순하며 앞서 본 “students.txt”와 같이 내용을 구성한다.

4. 프로그램 실행방법 및 예제

* 파일명을 입력하지 않을 경우, default로 “students.txt” 로부터 데이터를 읽는다.
* 파일명이 입력될 경우, 입력된 파일로부터 데이터를 읽는다.
* Show 입력 시(대소문자 상관 없음), 리스트가 출력된다.
* search입력 시 학번(Student ID)를 입력 받는다.
* 해당 학번이 존재하지 않는다면, “NO SUCH PERSON” 출력한다.
* 존재한다면 해당 학번과 일치하는 값의 데이터를 출력한다.
* changescore입력 시, 학번( Student ID)를 입력 받는다. 해당 학번이 존재하지 않으면 “NO SUCH PERSON)을, 있다면 mid/Final을 입력 받는다.
* mid나 final이 아닌 오타를 입력 받는다면 아무것도 실행되지 않는다.
* New score을 0~100이 아닌 다른 숫자를 입력 받는다면 아무것도 실행되지 않는다.



* Changescore 명령어에 적절한 값들이 들어오면 변경 전 학생의 데이터와 변경 후 학생의 데이터가 함께 출력한다. 사이에 “score changed”라는 문구도 함께 출력한다.
* changescore의 결과 값은 show에도 반영된다. 변경 된 값으로 실행됨을 확인할 수 있다.
* add 입력 시, 학번( Student ID), 이름(Name),.중간고사 점수, 기말고사 점수를 차례로 입력 받는다.
* 적절한 데이터 타입의 입력 값들이 들어오면 “Student added.”라는 메세지와 함께 값이 추가된다.
* show 입력 시, 데이터가 추가된 것을 확인할 수 있다.
* searchgrade 입력 시, 찾을 학점을 입력 받는다. 입력 받은 학점이 해당되는 학생들의 데이터를 모두 출력한다.
* 입력 받은 학점이 A, B, C, D, F중 하나에 해당하더라도 , 그 학점에 해당되는 학생이 존재하지 않는다면 “NO RESULTS.”라는 메시지를 출력한다.
* 입력 받은 학점이 A, B, C, D, F이외의 값이면 실행되지 않는다.
* remove 입력 시, 존재하지 않는 학번을 입력 받는다면 “NO SUCH PERSON.”메시지를 출력한다
* remove입력 시, 존재하는 학번을 입력 받으면 “Student removed”라는 메시지와 함께 해당 학번 데이터를 삭제한다.
* show를 통해 데이터가 삭제된 것을 확인할 수 있다.
* 현재 리스트에 존재하는 데이터를 모두 삭제하면 “List is empty.”라는 메시지를 출력한다.
* quit 입력 시, 저장 여부에 대한 입력을 받으며 yes입력 시 추가적으로 저장할 파일명을 물어본다. 파일명 입력 시 새로운 파일이 생성된다
* no입력 시 데이터는 저장되지 않고 해당 프로그램은 종료된다.

5. 토론

* Searchgrade 생성 시 새로운 리스트를 생성하지 않으면 출력에서 문제가 발생하였다. 그리하여 새로운 리스트를 생성하는 방식으로 풀었지만 다른 방법으로 처리하는 방법도 있을 것이라고 생각된다.
* 이중리스트를 사용하여 풀었지만 딕셔너리를 사용하는 방법 또한 익힐 예정이다.
* 본 과제는 주어진 명령에만 국한되어 실행할 수 있는 프로그램이기 때문에 실제 활용하기 위해서는 다른 명령어 추가 등 고려를 해야 될 것이다.
* 또한 함수 생성시, return을 어느 부분에 넣어야 되는지 고민이 많이 됐다. 반환값에 대해 많이 배울 수 있는 기회였다.

6. 결론

* 본 과제에서는 프로그램 작성 시에 필요한 함수들을 작성하고 사용하는 방법을 주로 학습하게 되었다. 특히, 함수의 선언에 따른 반환형 및 반환되는 값이 중요함을 알게 되었다. 이중 리스트 생성 시 고려해야 할 사항이 늘어나는 것을 확인하였고 이중 리스트에서 정렬하는 방법을 익힐 수 있었다. 뿐만 아니라, 원하는 값이 입력될 때까지 입력부를 반복 출력하는 기본적인 예외 처리에 대해 알 수 있었다. 이번에는 flow chart를 작성하지 못했지만 여유가 생긴다면 꼭 작성해볼 것이다.