**2019년 청년 AI Big Data 아카데미 7기**

**파이썬 프로젝트**

**Assignment #1**

**담당교수:** 윤은영

**학번:** s\_0381

**이름:** 이세원

**이메일:**

kanday5665@gmail.com

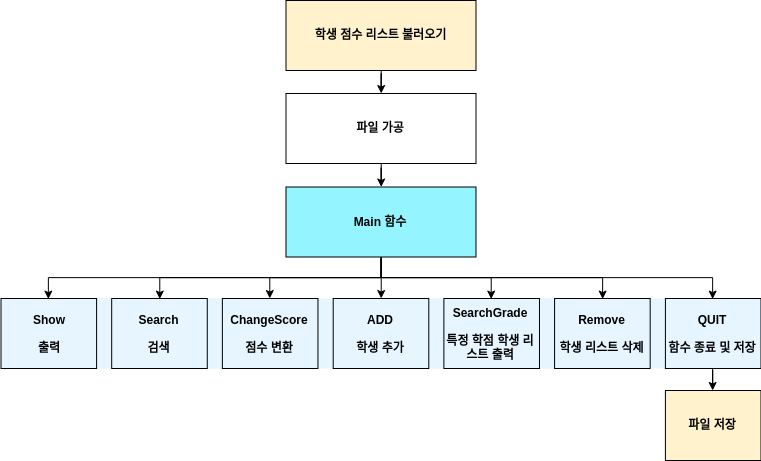
|  |
| --- |
| **명예서약(Honor code)**  **“나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.”** |

1. 문제의 개요

본 프로그램을 간략히 설명하면 다음과 같다.

* 입력된 학생 점수 파을 불러와서 사용자가 보기 편하게 가공 및 정렬하여 나타내준다.
* 학생 점수 리스트를 사용자의 7가지 명령어 입력에 따라 가공하고, 가공된 데이터를 원하는 파일명으로 저장한다.

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



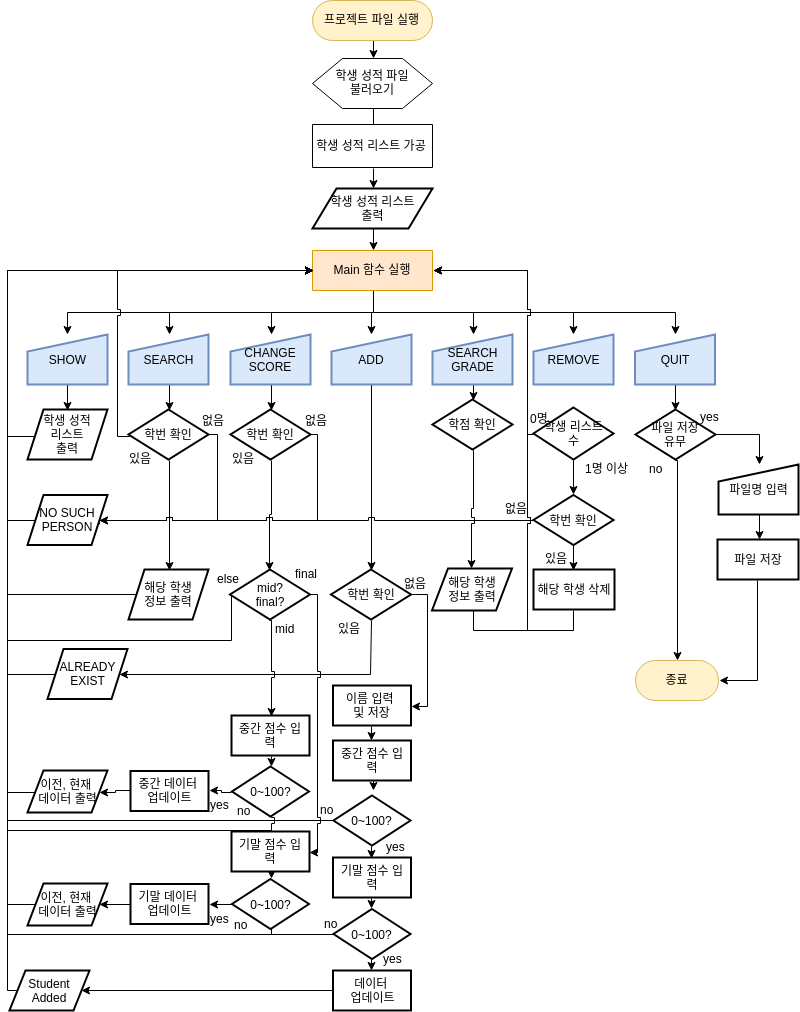
* 입력부 : 파일 불러오기 및 파일 가공하여 Main 함수에서 처리할 수 있도록 처리
* 처리부(Main 함수/Sub 함수) : Main함수에서 가공된 데이터를 출력하고 7가지 Sub 함수 및 기능과 연결되어 데이터를 변경 및 가공할 수 있도록 하는 함수
* 출력부(QUIT 함수) : 가공 완료된 데이터를 원하는 newStudents.txt 파일로 저장

2. 알고리즘

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

|  |  |
| --- | --- |
| **Pseudo-algorithm for dot-line distance** | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | **파일 불러오기** : 파일 명 (입력 값, 없을 경우 students.txt 기본 값으로)  **불러온 파일을 변수 단위 저장하기 :**  기존 변수 : student : 학번 / name : 이름 / midterm : 중간고사 점수 / final : 기말 점수  생성 필요 변수 : average : 평균점수 / grade : 학점  **불러온 파일 가공하기**: 리스트로 제작 [student, name, midterm, final, average, grade]  **변수별 계산 함수 작성 – score\_cal(list) :**  average = (midterm + final) / 2  if average >= 90 : grade = "A"  elif average >= 80 : grade = "B"  elif average >= 70 : grade = "C”  elif average >= 60 : grade = "D"  else : grade = "F"  **가공된 데이터 포맷 함수 작성 formats([list]) :**  “ Student Name Midterm Final Average Grade  ---------------------------------------------------------------------------------”  리스트로 제작된 데이터를 넣으면 상단에 위 형태로 나올 수 있도록 함수 제작  **기존 학생 리스트에서 기본적인 정보 가공 및 포맷팅 하여 출력**  부가 기능을 한 번에 출력할 수 있도록 연결해놓은 Main 함수 생성 :  def Main():  formats (list)  while True :  funtion = input(“원하는 함수 명령어”).upper() #함수는 대문자로 인해서 출력되도록  SHOW, SEARCH, CHANGESCORE, ADD, SEARCHGRADE, REMOVE, QUIT  **def SHOW:**  average 기준, 내림차순으로 학생 리스트를 formats 함수와 함께 나타내줌  **def SEARCH:**  student에 검색이 필요한 학생의 학번을 입력할 경우, 해당 학생의 정보 나타내줌  **def CHANGESCORE:**  student에 점수 변경이 필요한 학생의 학번을 입력하게되면, 중간/기말고사 중 변경할 부분을  확인한 후, 해당 점수를 입력받아 해당 학생의 점수를 변경해준다.  **def ADD:**  추가할 학생의 학번을 입력받은 후, 해당 학생의 이름, 중간/기말고사 점수를 입력받아 해당 학   생의 정보를 학생 성적 목록에 추가한다.  **def SEARCHGRADE:**  특정 학점을 입력하면 해당 학점에 해당하는 학생이 전부 출력되도록 나타낸다.  **def REMOVE:**  특정 학생의 학번을 입력하면 해당 학생이 성적 목록에서 전부 지워지도록 한다.  **def QUIT:**  해당 프로그램을 중지, yes 입력 시, 변경된 사항이 저장될 수 있도록, 파일 이름을 지정하  는 입력을 받도록 하며, no 입력 시 변경된 내용 저장하지 않고 해당 프로그램을 중지시킨다. |

위의 의사 알고리즘을 Flowchart를 통해 표현하면 아래와 같다.



3. 프로그램 구조 및 설명

a) 기존 학생 데이터 불러와서 가공

* 프로그램을 실행하면 가공이 필요한 데이터를 불러와서, 사용자가 보기 편하도록 포맷팅하여 화면에 출력해준다.
* 이후 평균점수와 학점을 자동으로 계산해준다.

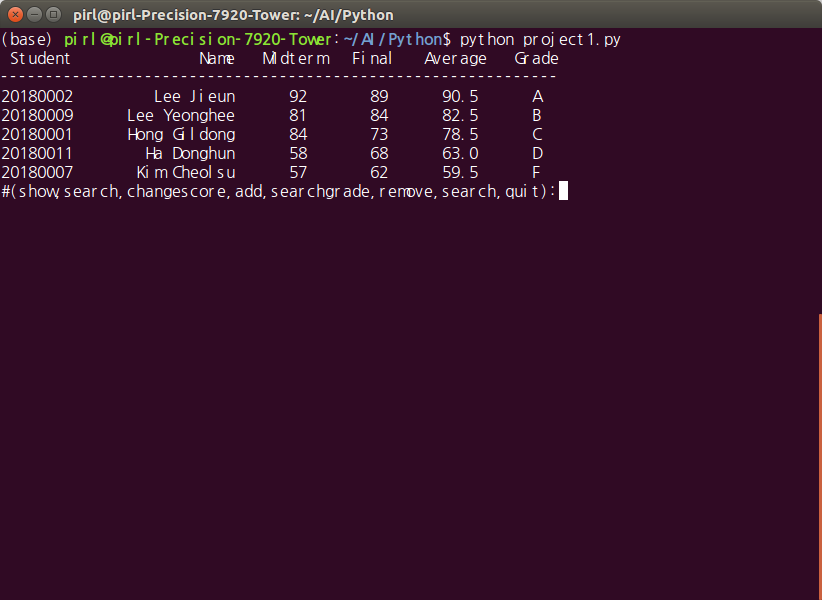
b) 학생 성적 출력 / 추가 / 삭제 / 변경 기능

* 사용자가 입력된 학생 리스트를 보면서, 원하는 방향으로 목록을 수정할 수 있다.
* 총 7가지 기능 보유
  + Show 기능 : 전체 학생정보 출력 (평균점수 기준 내림차순 정렬)
  + Search 기능: 학생의 학번을 기준으로 학번/이름/중간, 기말고사 점수/ 평균/학점 출
  + ChangeScore기능: 목록에 저장된 학생의 중간고사/기말고사 점수 수정 후 점수 변경 및 재산출
  + ADD 기능 : 목록내 신규 학생 및 중간/기말 점수 추가
  + SearchGrade 기능: 특정 Grade에 해당하는 학생 모두 출력
  + Remove 기능: 목록 내에서 학번 기준으로 학생 삭제
  + QUIT 기능: 프로그램 종료기능, 종료하면서 파일 저장 유무 확인

c) 수정 완료된 학생 성적 목록 저장

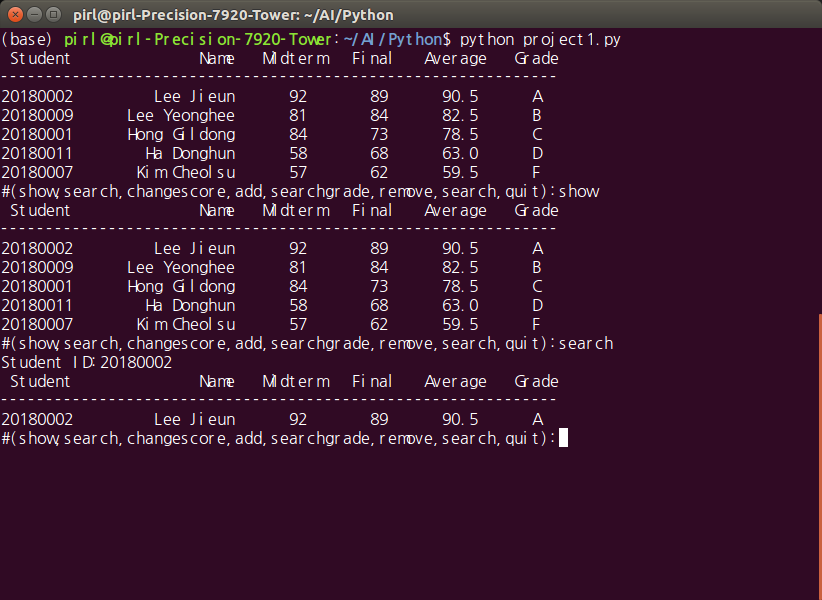
* 모든 처리가 완료된 학생 성적 목록을 다른 파일 이름을 지정하여 해당 파일에 저장한다.
* 기존 파일은 원본으로 유지하므로, 수정된 부분과 오류가 있었던 부분을 확인 가능하다.

4. 프로그램 실행방법 및 예제

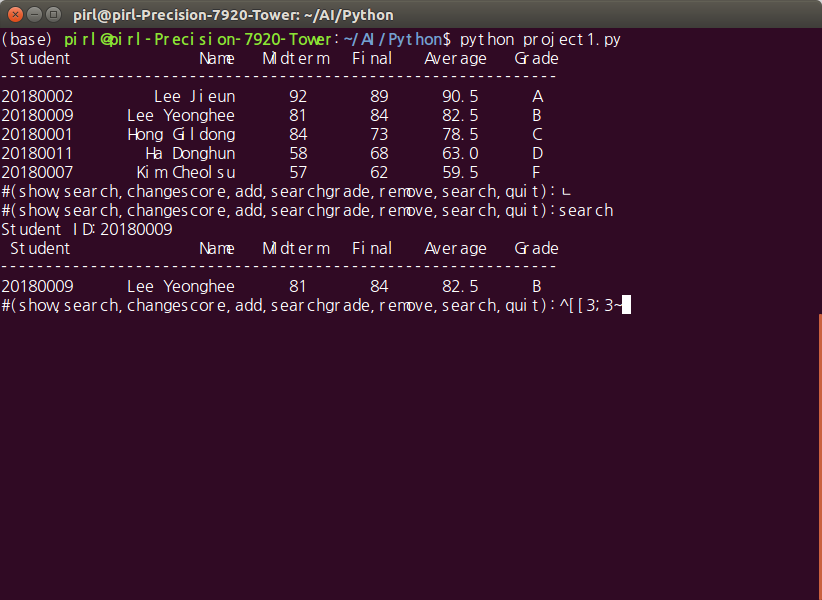


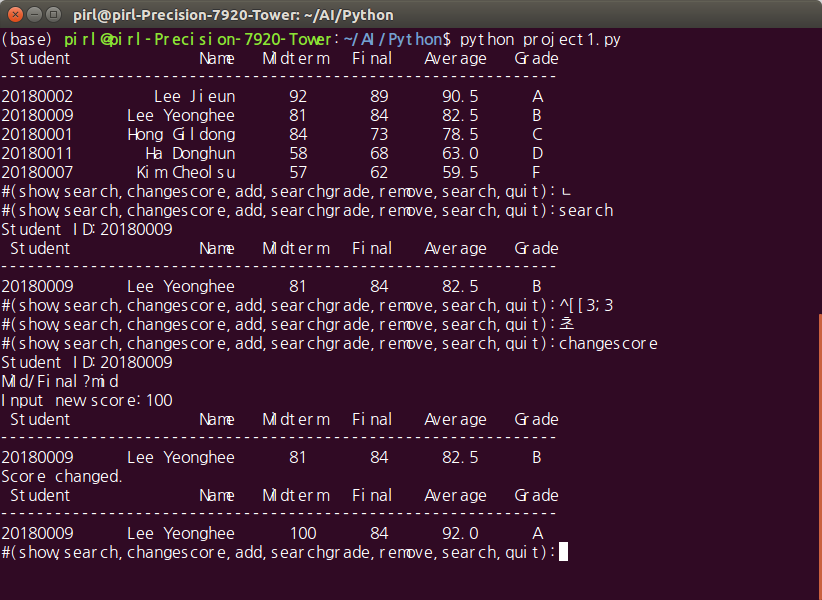
1) 가상환경(터미널)에서 해당 프로그래밍 파일이 있는 폴더로 접근한 후, python project1.py 입력하여 실행한다.

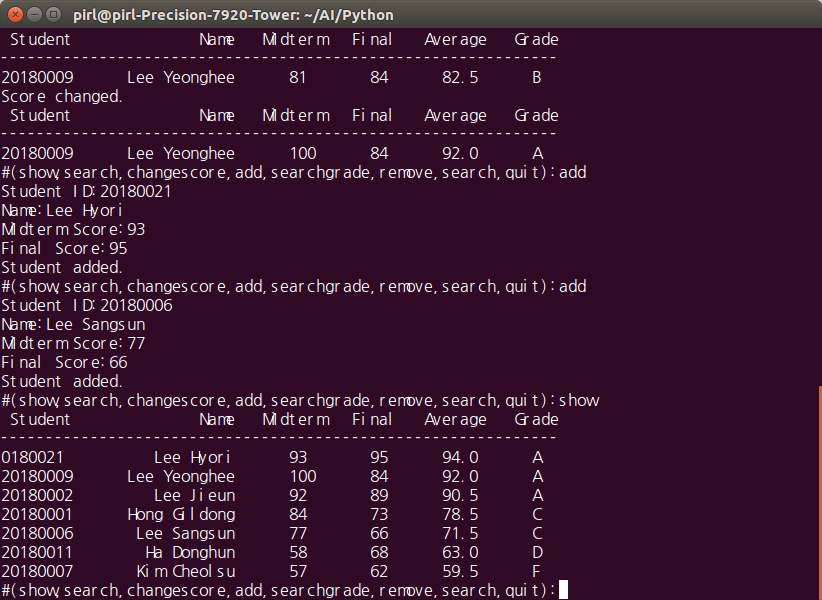
2) 실행 시, 위와 같이 총 7가지 명령어를 입력할 수 있는 명령창이뜬다.

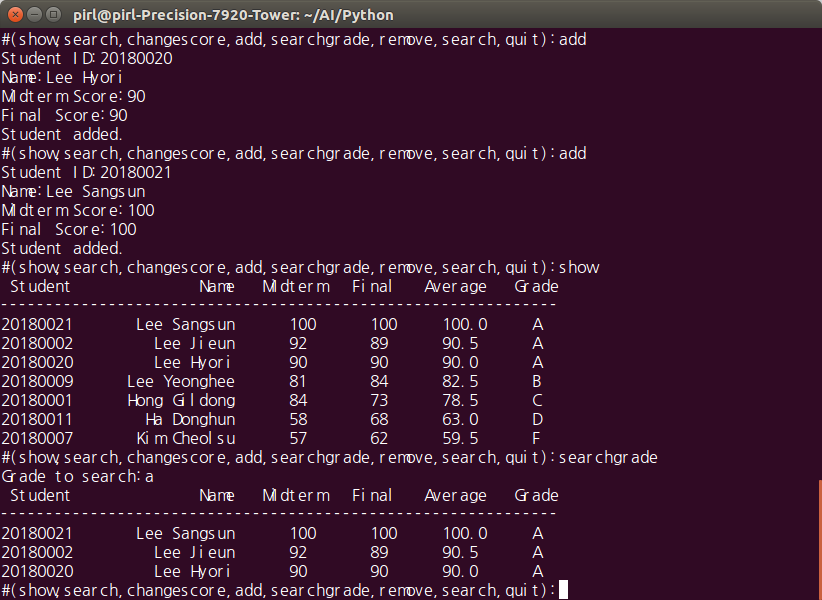


3) show 명령어를 입력할 경우, 평균 점수를 기준으로 학생 목록을 내림차순하여 출력해준다.

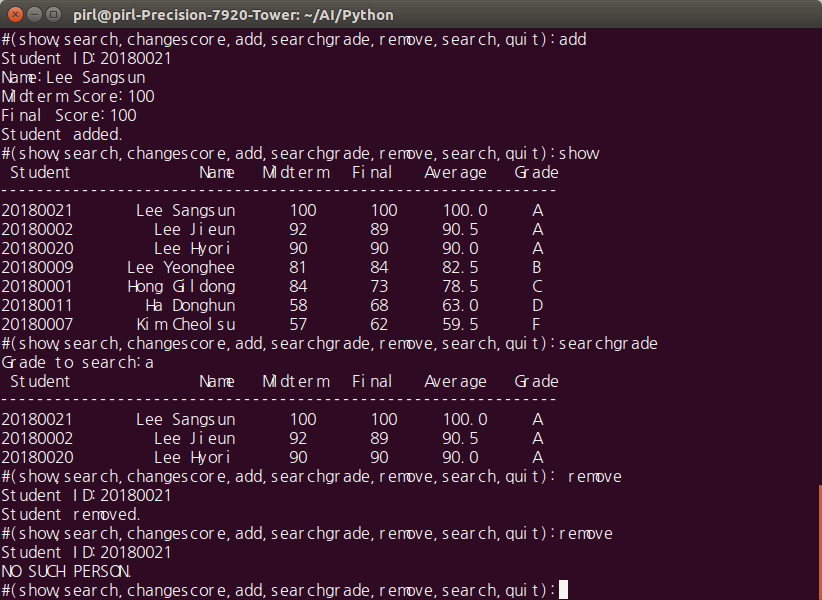
4) search 명령어를 입력할 경우, 찾고자하는 Student ID를 입력할 수 있는 창이 나오며, ID 입력시 위와 같이, 해당 학번에 해당하는 학생의 정보가 출력된다. 이때 학생이 없으면 ‘NO SUCH PERSON’ 출력

5) changescore 명령어를 입력할 경우, 점수를 변경할 학생의 학번을 확인하고, 중간고사/기말고사를 선택한 후 점수를 입력하면, 해당 학생의 이전 점수와, 변경된 점수를 출력하여 보여준다. 이 때, 점수가 100점 초과, 0점 미만일 경우, 점수의 변경 없이 명령어 입력창으로 되돌아온다.

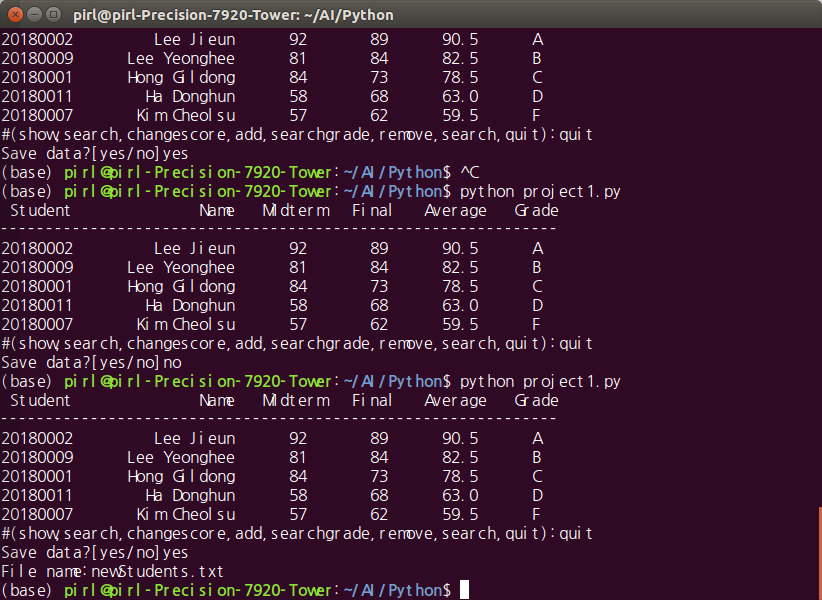
6) add 명령어 입력 시, 학번, 이름, 중간고사 점수, 기말고사 점수를 차례로 입력받고, 추가한 이후에 “Student added.”를 위와 같이 출력해준다. 평균점수와 학점은 자동으로 계산된다.



7) 위와 같이 searchgrade 명령어 입력 시, 특정 grade를 입력받은 후, 해당 grade에 해당하는 학생을 모두 출력, 해당 학점 학생이 없을 경우 “NO RESULTS.” 출력



8) 위와 같이, remove 명령어 입력 시, 삭제하고자 하는 학생의 학번을 입력받아, 학생이 목록에 있는 경우 해당 학생의 기록을 삭제한다. 삭제 시, ‘Student removed’ 출력되며, 학생이 없을 경우, ‘NO SUCH PERSON’이 출력된다.



9) 위와 같이 quit 명령어 입력 시, yes/no 두 가지 입력해도 프로그램이 종료되지만, yes 입력 시, 저장할 파일 명을 받아, 현재까지 편집된 학생 리스트를 해당 이름으로 저장해준다.

5. 토론

* 처음에는 구현해야하는 기능에 따라, 함수 개수를 정하고 제작하려고 했으나 계속적으로 반복되는 출력 포맷과 관련한 부분이나, 점수를 계산하는 부분을 함수화 하는 것이 효율적이라고 생각했으며, 처음부터 해당 지식이 있었다면 사전 설계 시 고려해볼 수 있었을 것 같다.
* 새롭게 학생 데이터를 추가해야 할 때, 입력해야하는 데이터 공간을 새롭게 지정하여 만들어주는 것이 생소하여 어려웠다.
* 코드를 짰을 때 가독성이 좋지 않아, 짜고 난 후 디버깅을 할 때, 또 다시 코드를 한 동안 들여다보아야 하는 번거로움이 있었다. 주석 처리와 변수명을 조금 더 명확하고 간결하게 짤 필요가 있을 것 같다.

6. 결론

* 본 과제를 통해 여러가지 함수를 구현해서, 한 가지 프로그램으로 모아서 구현하는 방법에 익숙해질 수 있었다.
* 해당 프로젝트는 혼자서 작업하였으나, 다수의 사람과 GitHub를 통해 메인 기능만 함께 구현하고 각각의 기능은 나눠서 구현하는 방법도 가능할 것 같다는 생각을 해보았다.

7. 개선방향

* 점수 입력 시에, 정해졌던 예외사항이 이외에 추가할 수 있는 예외의 경우의 수가 더 많기 때문에, 해당 경우의 수를 꼼꼼히 파악해서 에러 메시지를 띄워준다면 조금 더 사용자의 편의성이 높을 것이라고 생각된다. (예를 들면, 학번 입력 장소에, 학번의 규격과 맞지 않는 다른 데이터가 들어가더라도 입력이 되는 오류 등의 예외사항을 더 추가해야할 것 같다.)
* 프로그래밍 코드가 효율적이지 못하고, 단순하게 짜여진 부분이 많아 단순히 반복되는 부분은 함수를 새롭게 짜서 대체하는 방법이 좋을 것 같다
* 리스트를 이용해 프로그래밍하였으나, 딕셔너리를 통해 작업하는 것이 조금 더 효율적일 것 같다.