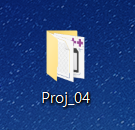
Project #4:: 프로그램 보고서:: 각 프로그램 기능에 대한 보고서

**< 4번 프로그램 구성 >**

****

**- 텍스트 문서(.txt)**



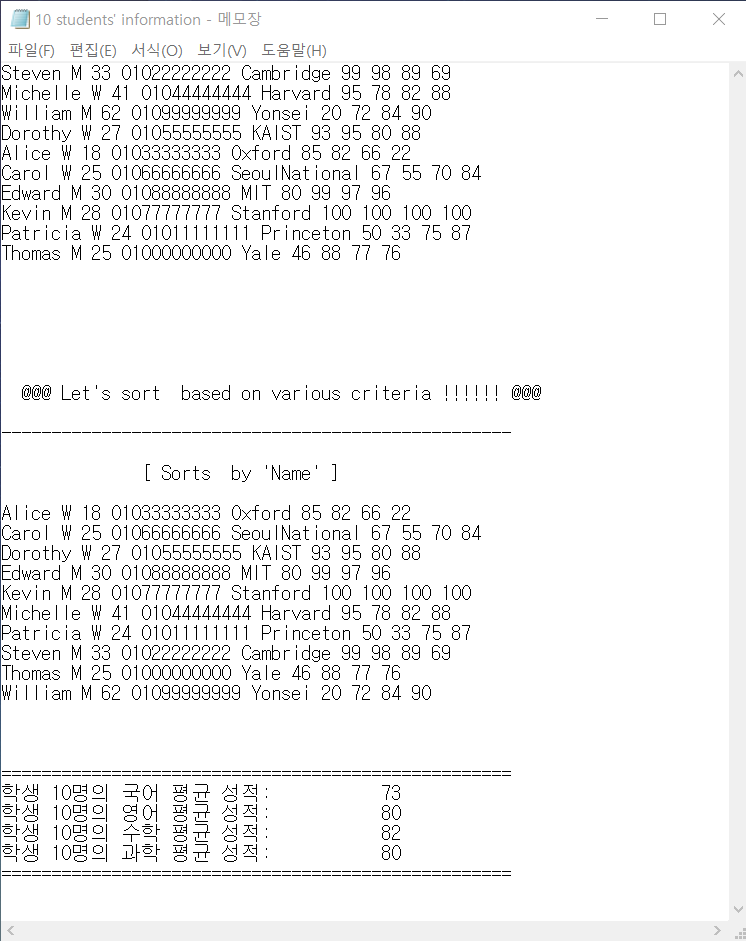
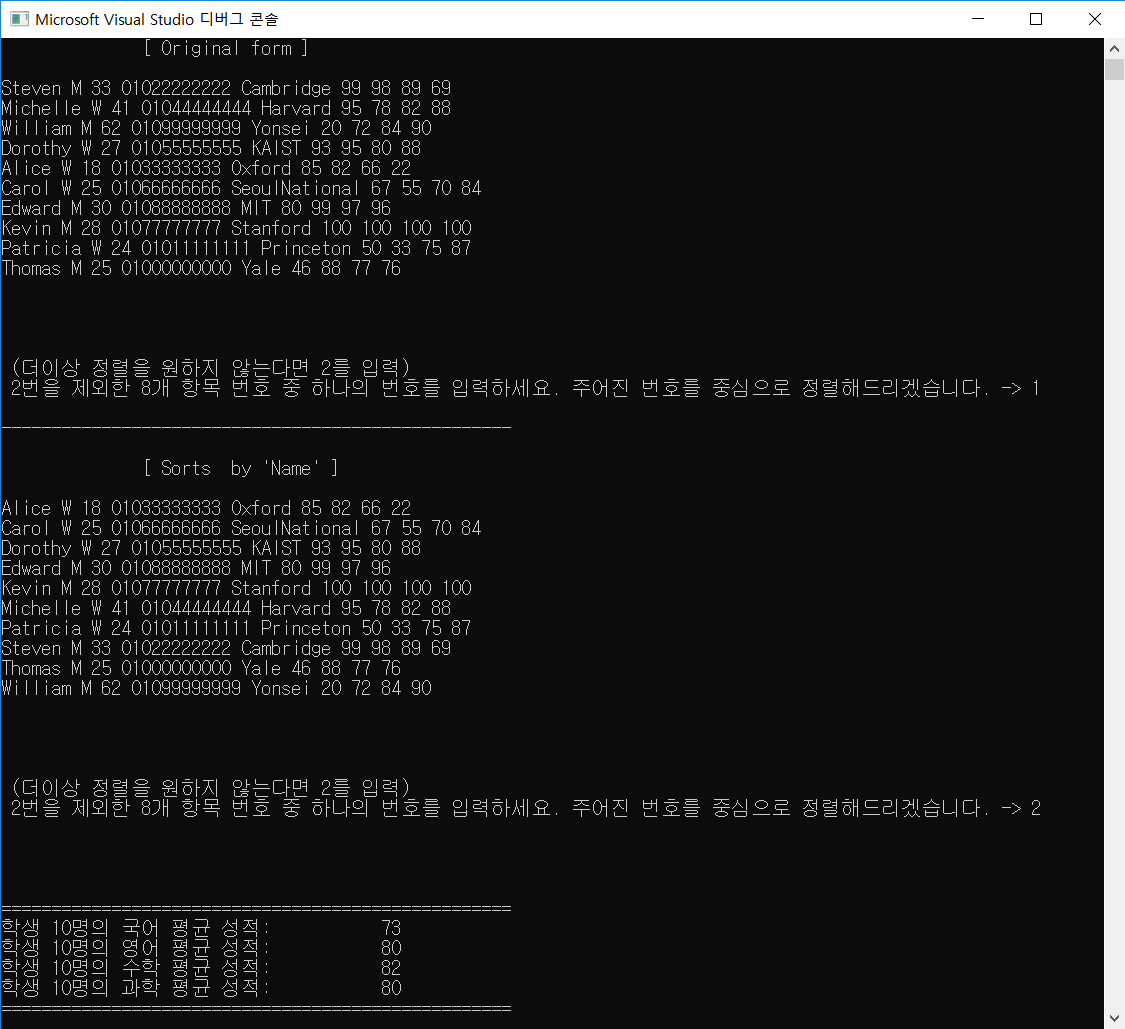
**- Source Code**



4번 프로그램에서는 ‘파일 입출력’ 개념이 등장한다. 파일에 있는 데이터들을 읽어와서, 그 데이터들을 가지고 각종 처리를 진행한 다음, 그 내용을 다시 파일에 출력시킬 것이다. 따라서 Proj\_04 프로젝트 폴더 안에는 텍스트 파일도 같이 들어 있다.

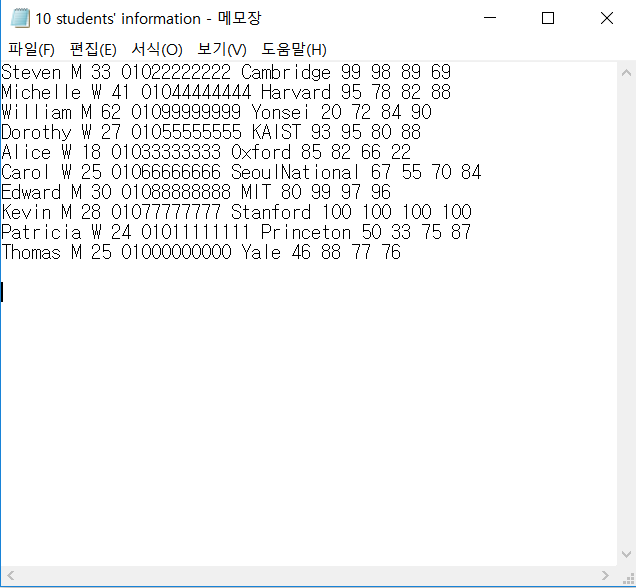
**< 출력 방식 >**

나는 파일에서 읽어온 데이터들을 가지고 정렬 처리와 평균 계산을 진행한 다음, 그 내용을 **실행 결과 화면에서도 출력시키고, 파일에서도 출력 시켰다.**

****

**실행 결과 화면 단계별 Capture (작은 size 화면으로 Capture)**

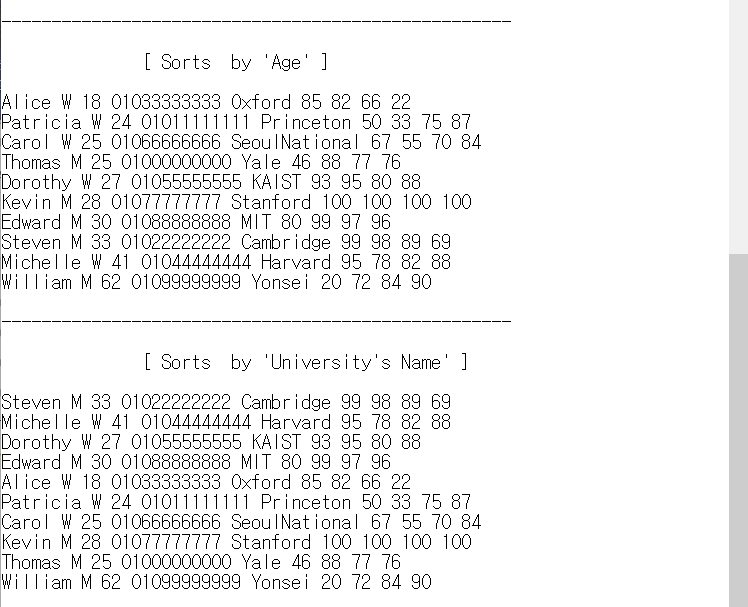
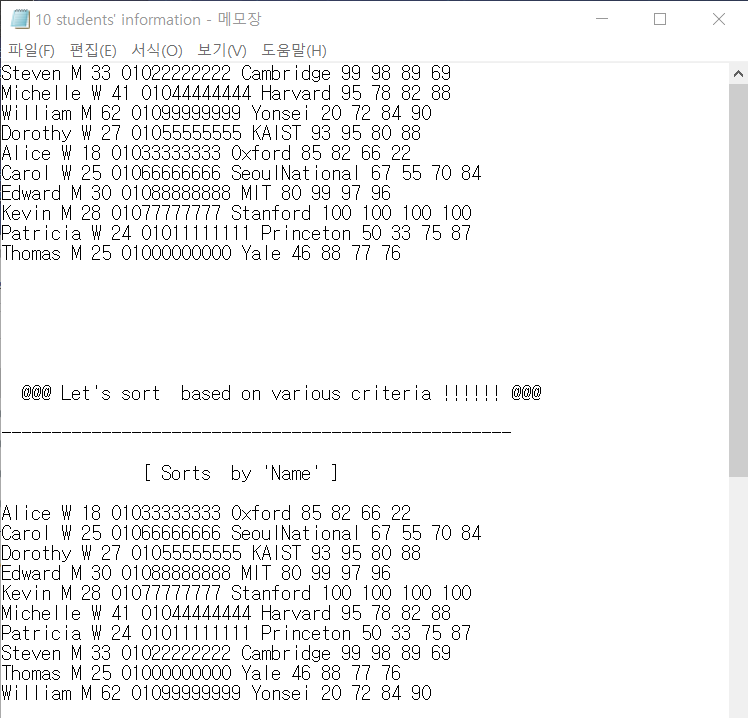
* **프로그램을 실행시키기 전, 텍스트 파일의 모습**

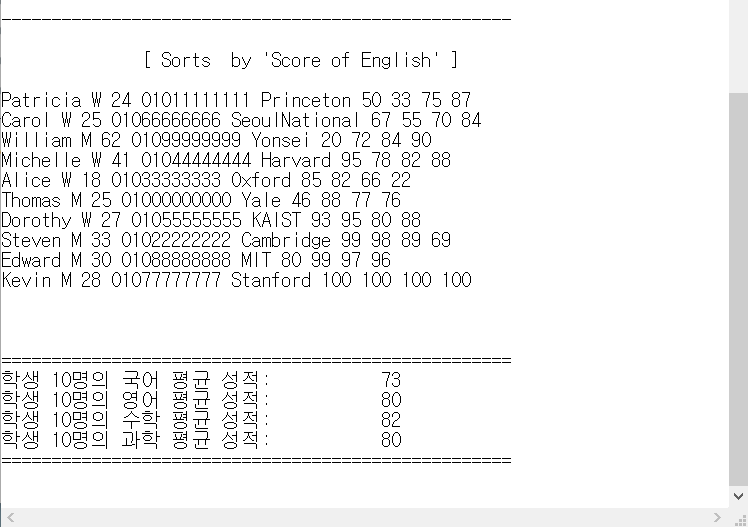


나는 텍스트 파일 하나를 생성하여 학생 10명의 데이터들을 미리 입력해두었다. 4번 프로그램에서는 fscanf함수를 이용하여 저 데이터들을 차례대로 읽어낼 것이다.

이제 4번 프로그램을 모두 실행시키고나면, 실행 결과의 내용들이 저 텍스트 파일 기존 내용 뒤에 그대로 이어서 출력될 것이다. 파일을 열 때 모드를 “a+”로 지정해놓았기 때문에, 데이터 읽기와 추가하여 쓰기 둘 다 가능하다. 즉, 이 텍스트 파일 하나에서 ‘읽기’와 ‘출력’을 같이 진행한다.

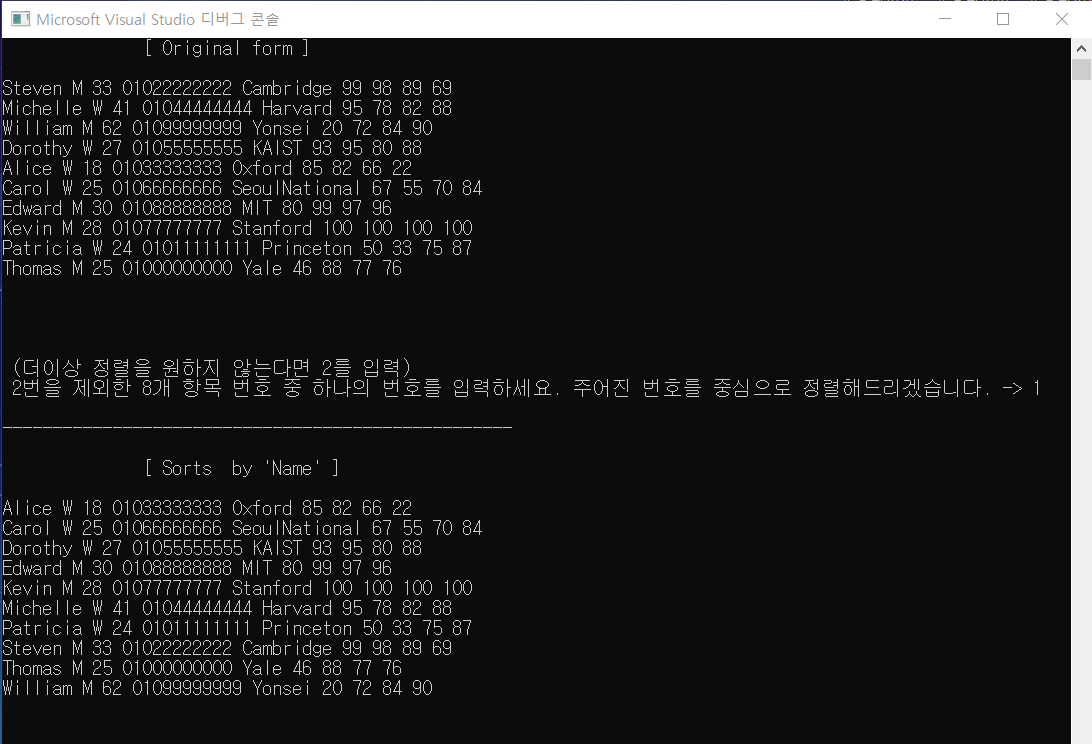
* **프로그램을 실행시키고난 후, 텍스트 파일 모습의 변화**

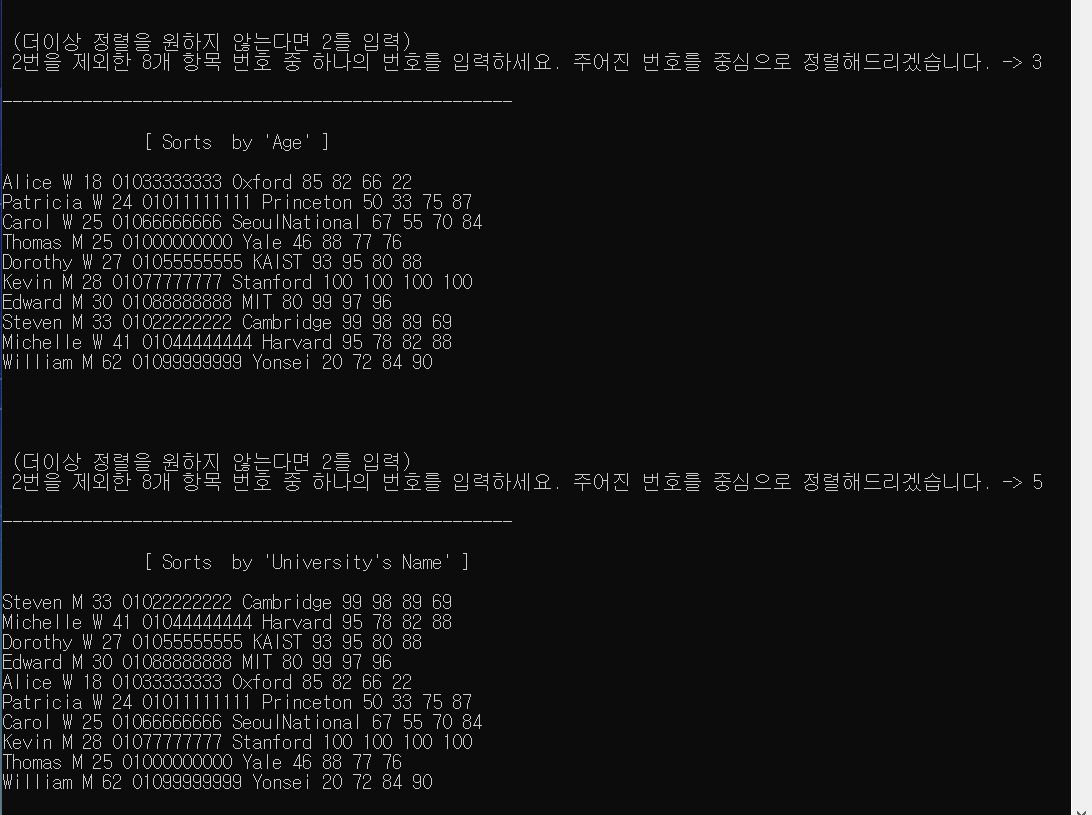


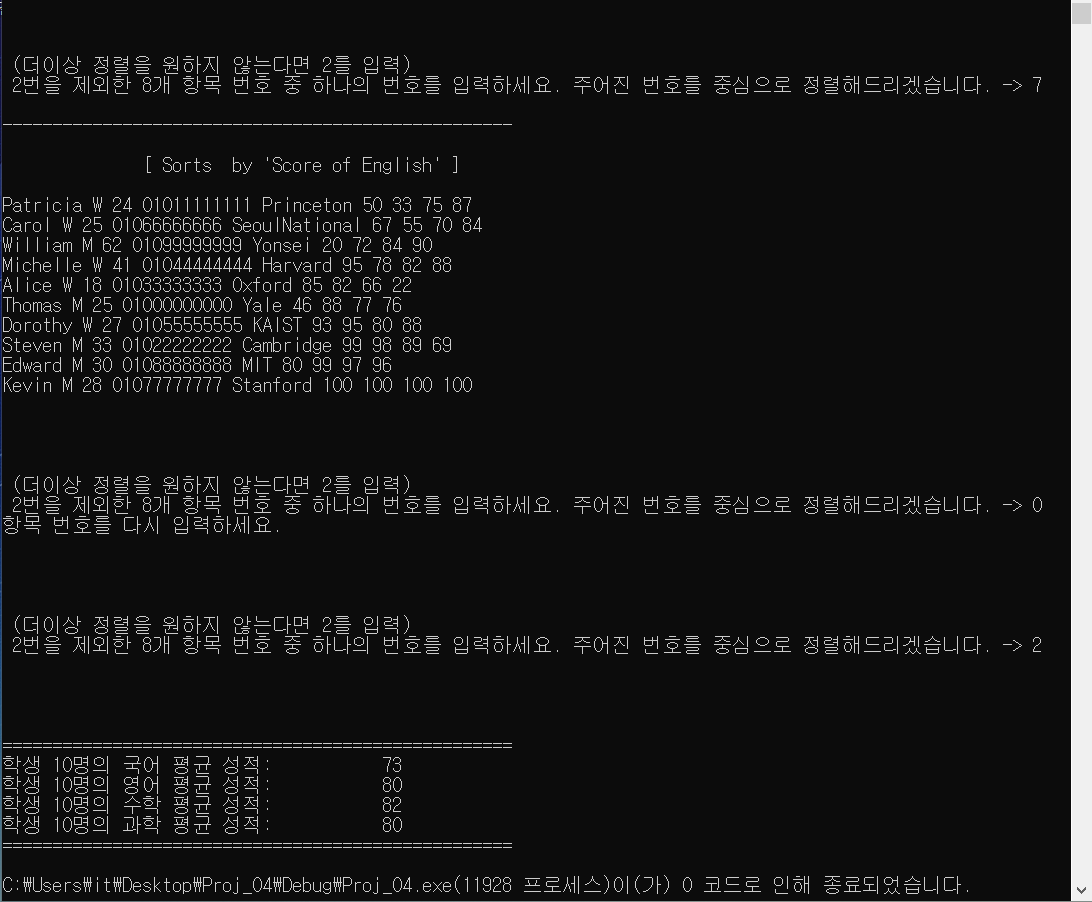


프로그램을 실행시키고나서 해당 텍스트 파일을 다시 열어보면 위와 같은 모습으로 텍스트 파일의 모습이 바뀌어 있을 것이다. 실행 결과의 내용이 기존에 있던 내용 뒤에 그대로 이어서 출력된 것이다.

* **프로그램을 실행시킨 커맨드 창의 모습**







나는 실행 내용을 파일 뿐만 아니라, 커맨드 창에서도 출력시키도록 하였다. 커맨드 창을 보면 프로그램이 어떻게 흘러가는지 한 눈에 파악할 수 있을 것이다. 파일에 있던 데이터들을 잘 읽어왔는지 확인하기 위한 출력부터 시작해서, 사용자가 항목 번호를 입력할 때마다 그 번호를 중심으로 정렬된 모습을 계속해서 보여주는 출력, 2를 눌러서 루프를 빠져나오면 마지막으로 평균 모습도 보여주는 출력까지. 컴파일 할 때마다 프로그램이 어떻게 흘러가는지 확인하기 위한 이유로 커맨드 창에서도 출력시키도록 하였다. 커맨드 창에서 출력시킨 또 다른 이유 중 하나는 조금 더 완성도 있는 프로그램을 제출하고 싶은 마음 때문인 것도 있다.

*# 위 결과 화면 은 하나의 예시이다. 모든 번호 항목을 기준으로 한 정렬 출력이 가능하다. 프로그램을 실행시켜서 원하는 기준 항목 번호를 입력하여 그를 기준으로 정렬된 모습을 확인할 수 있다.*

*# 참고로 텍스트 파일 같은 경우, 제출은 프로그램 실행시키기 전의 모습으로 보낼 것이다. 프로그램을 실제로 돌리고나면, 그 텍스트 파일의 내용이 커맨드 창에서 실행시킨대로 출력되어있을 것이다!*

**사용한 각 함수에 대한 설명 (Input, Output, 알고리즘 설명)**

**< student\_management.cpp >**

4번 프로그램의 소스 코드가 너무 길어 소스 코드 캡쳐는 생략하고 큰 구조 흐름을 설명하도록 할 것이다.

4번 프로그램 student\_managemant.cpp 파일의 흐름은 다음과 같다.

1. **파일로부터 학생 10명의 데이터들을 입력 받아 구조체 배열 변수에 저장한다.**
2. **사용자로부터 2번을 제외한 8개 항목 번호를 입력 받을 때마다 그 주어진 번호를 중심으로 정렬한 모습을 출력시킨다. 이는 루프를 돌려서 사용자가 계속 항목 번호를 입력하고 정렬된 모습을 각각 바로바로 확인할 수 있도록 하게 한다. 루프를 빠져나오려면 2를 입력.**
3. **루프를 빠져나오게 되면, 마지막으로 평균도 출력하여 준다.**

< 더 디테일한 소스 코드의 흐름 >

1. **student 구조체 정의**
2. **메인 함수 시작.**
3. **student구조체 배열 선언 (10개)**
4. **파일 “a+”모드로 열기**
5. **fscanf함수로 파일에 있는 데이터 읽어와서, 앞서 선언한 구조체 배열에 저장**
6. **이제 구조체 배열에 저장된 값들을 가지고 정렬 처리를 진행할 것**
7. **다중 if-else문을 사용하여 항목 번호대로 상황을 나눈다.**
8. **각 조건문의 내용에 맞게 정렬 코드를 기술한다.**
9. **위 과정을 무한 루프를 돌려서, 사용자가 항목 번호를 계속 입력할 때마다 그 번호를 중심으로 정렬된 모습을 출력 받을 수 있도록 한다.**
10. **루프를 나오고 싶으면 2를 입력하면 됨.**
11. **루프를 빠져나오게 되면, 마지막으로 평균도 출력하여 준다.**
12. **파일 닫기**
13. **메인 함수 종료**

4번 프로그램은 위와 같이 흘러간다. 소스 코드에 주석을 모두 달아 놓았으니 자세한 설명은 주석을 참고하기!

참고로 구조체 배열 변수에 값을 저장시키거나 출력시킬 때에는, 점을 이용한 ‘직접 접근’ 방식을 이용하였다. 구조체 ‘배열’이기 때문에(배열 자체가 주소를 상징하기 때문) 포인터를 사용한 ->를 이용한 ‘간접(참조) 접근’을 이용할 수도 있다. 하지만 내게 아직은 직접 접근이 더 익숙하기 때문에 전자의 방식을 이용하였다.