

# REPORT

## 보고서 작성 서약서

1. 나는 타학생의 보고서를 복사(Copy)하지 않았습니다.
2. 나는 타학생의 보고서를 인터넷에서 다운로드 하여 대체하지 않았습니다.
3. 나는 타인에게 보고서 제출 전에 보고서를 보여주지 않았습니다.
4. 보고서 제출 기한을 준수하였습니다.

나는 보고서 작성시 위법 행위를 하지 않고,  
성.균.인으로서 나의 명예를 지킬 것을 약속합니다.

과 목 : 전자전기 프로그래밍실습

과 제 명 : HW4

담당교수 : 민 형 복

학 과 : 전자전기공학부

학 년 : 2

학 번 : 2017311583

이 름 : 정해진

제 출 일 : 2018.4.16

## 1. Introduction

실습 2, 3주차 내용의 응용이다. text file과 binary file을 읽고 쓰기를 하여 주어진 문제를 해결한다.

## 2. Problem Statement

### ① Describe what is the problem.

학생 6명의 이름과 성별, 4개 과목의 점수가 들어있는 text file(infile.txt)이 주어진다. (숫자 6, 4는 command line argument에서 읽음)

- 이 text file을 읽어서, 그 내용과 4개 과목의 점수들의 평균값(자료형 : double)을 binary file(binary.dat)에 저장한다.
- 방금 저장한 binary file을 읽어서, 그 binary file의 크기를 측정하여 printf() 함수를 이용하여 출력하고, binary file의 내용은 다시 text file(outfile.txt)에 저장한다.

### ② Describe how do you solve the problem.

- Main 함수

Command line argument로부터 학생수, 과목수를 get\_positive\_integer 함수를 통해 num\_persons, num\_grades에 저장한다. 그 후, buildMemory 함수를 통해 메모리를 할당하고, writeBIN 함수를 통해 infile.txt 파일을 읽는다. 그 후 writeBIN 함수를 통해 이진 파일에 이진 형식으로 내용을 저장한다. 그리고 할당한 메모리를 disposeMemory 함수를 통해 돌려준다. 저장했던 이진 파일을 다시 readBIN 함수를 통해 읽어 여러 변수들에 저장한 후, writeTXT 함수를 통해 outfile.txt 파일에 내용들에 과목 평균을 더 추가하여 저장한다. 그리고 할당한

메모리를 disposeMemory 함수를 통해 돌려준다.

- disposeMemory 함수

input variable 들을 myfree 함수를 통해 할당했던 모든 메모리들을 돌려준다.

- readTXT 함수

infile.txt의 내용을 읽어 각 변수에 저장한다. infile.txt에는 이름, 성별, 4개 과목의 점수가 적혀있다.

- writeBIN 함수

readTXT를 통해 읽은 infile.txt의 내용을 토대로 이름의 길이, 4개 과목의 점수의 평균 항목을 더 추가하여 이진 형식으로 binary.dat에 저장한다.

- readBIN 함수

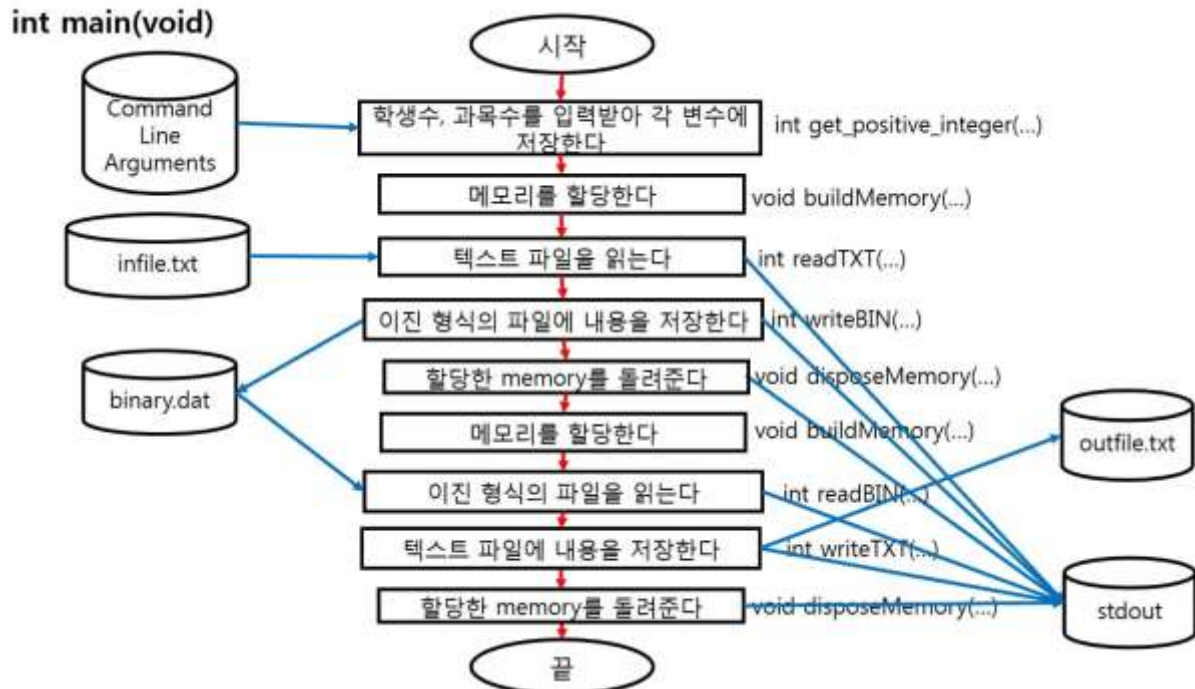
binary.dat의 내용을 이진 형식으로 다시 읽어서 각 변수에 저장한다.

- writeTXT 함수

readBIN을 통해 읽은 binary.dat의 내용을 토대로 4개 과목의 점수의 평균 항목을 더 추가하여 outfile.txt에 저장한다.

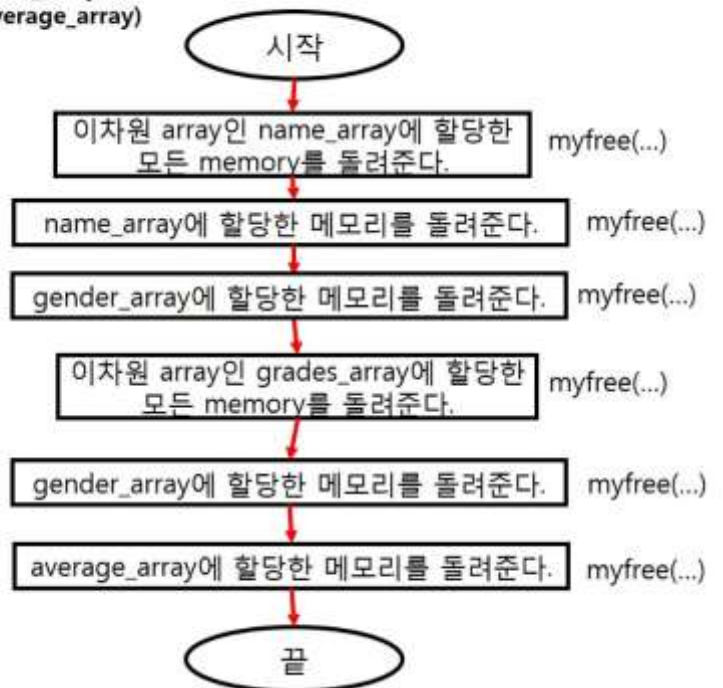
### ③ Draw a flowchart of your algorithm

#### 1. Main 함수



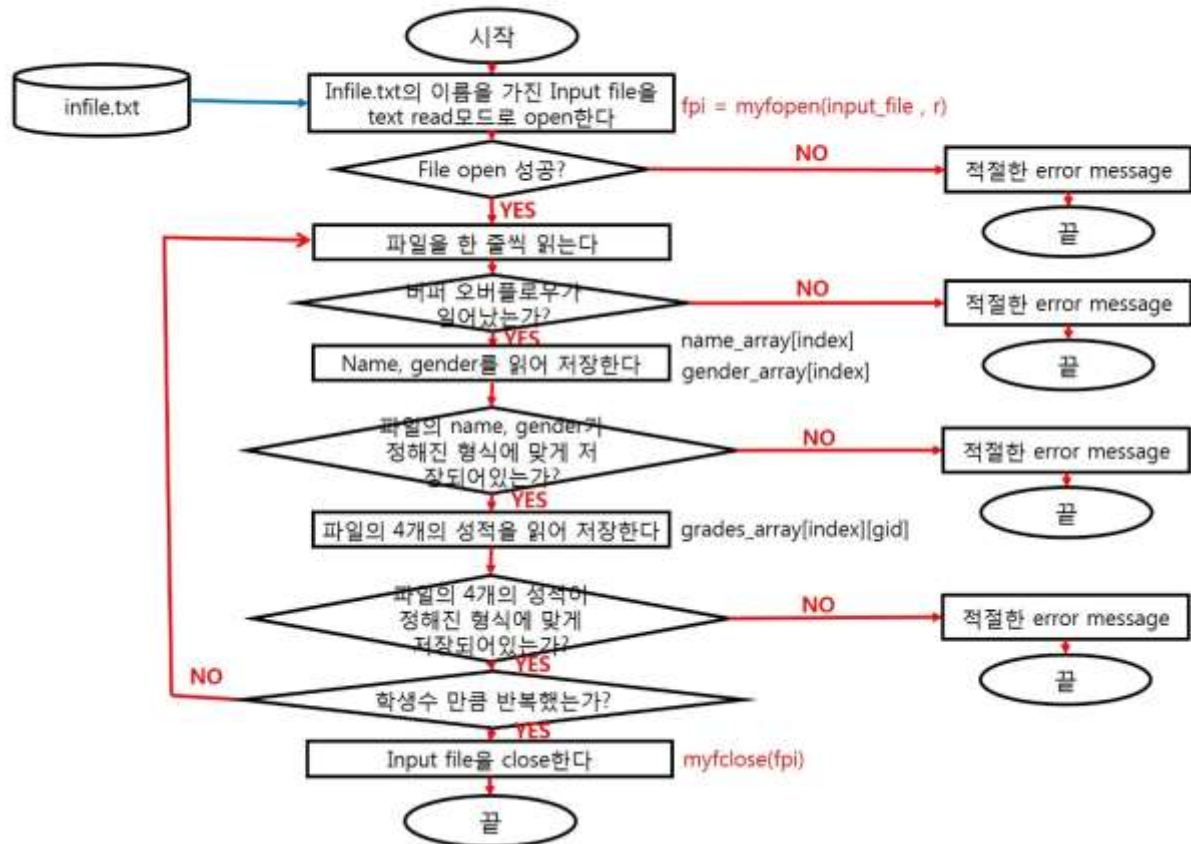
#### 2. disposeMemory 함수

**void disposeMemory(int num\_persons, int num\_grades,  
char \*\*name\_array, char \*gender\_array,  
int \*\*grades\_array, double \*average\_array)**



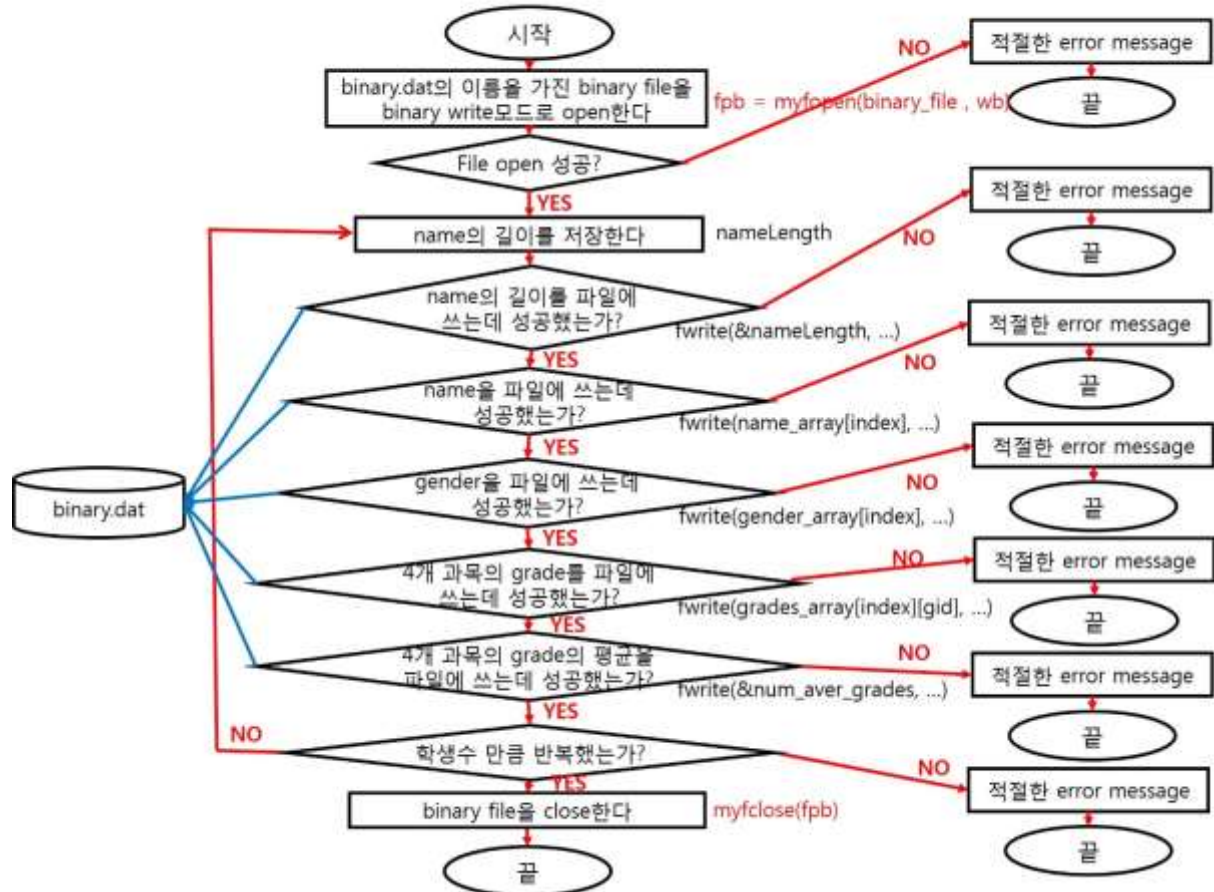
### 3. readTXT 함수

int readTXT (char \*input\_file, int num\_persons, int num\_grades,  
char \*\*name\_array, char \*gender\_array, int \*\*grades\_array)



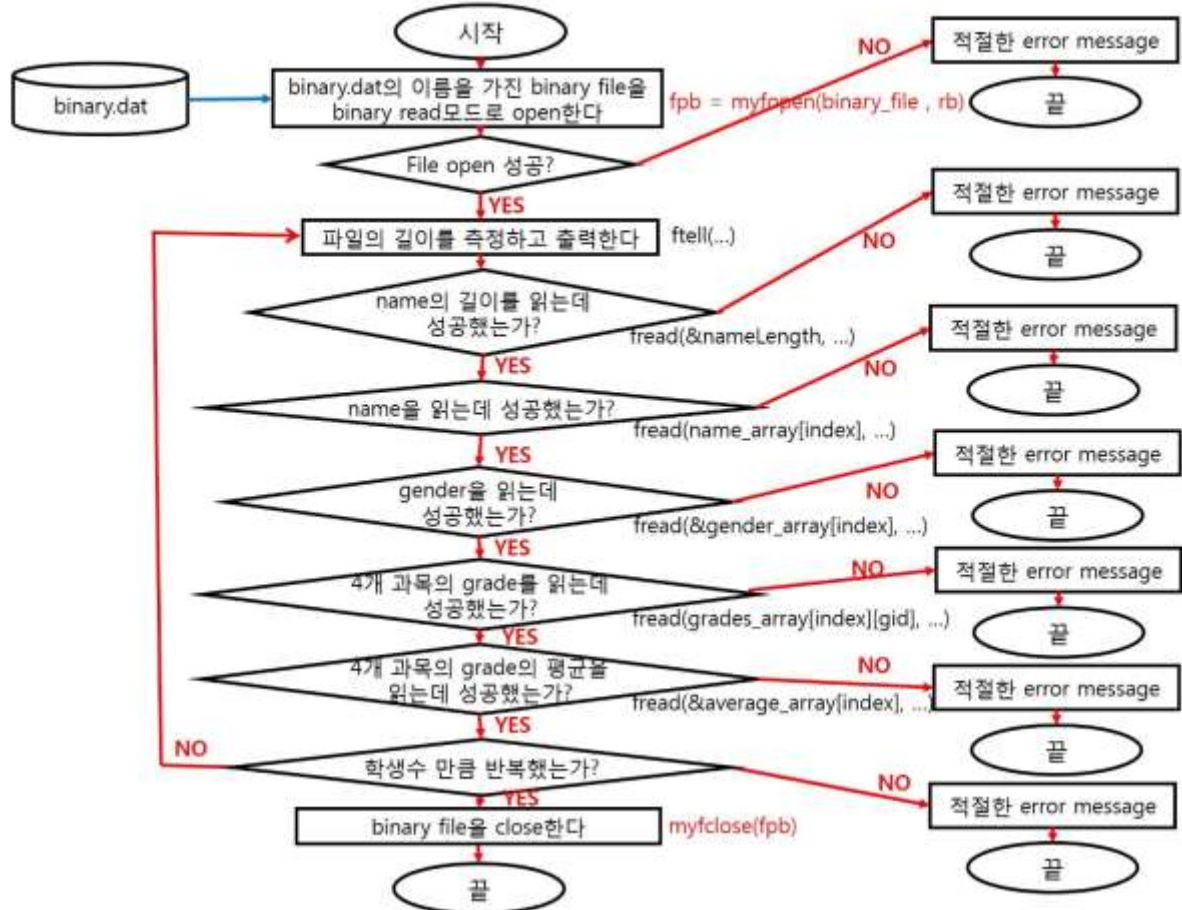
#### 4. writeBIN 함수

```
int writeBIN (char *binary_file, int num_persons, int num_grades,
             char **name_array, char *gender_array, int **grades_array)
```



## 5. readBIN 함수

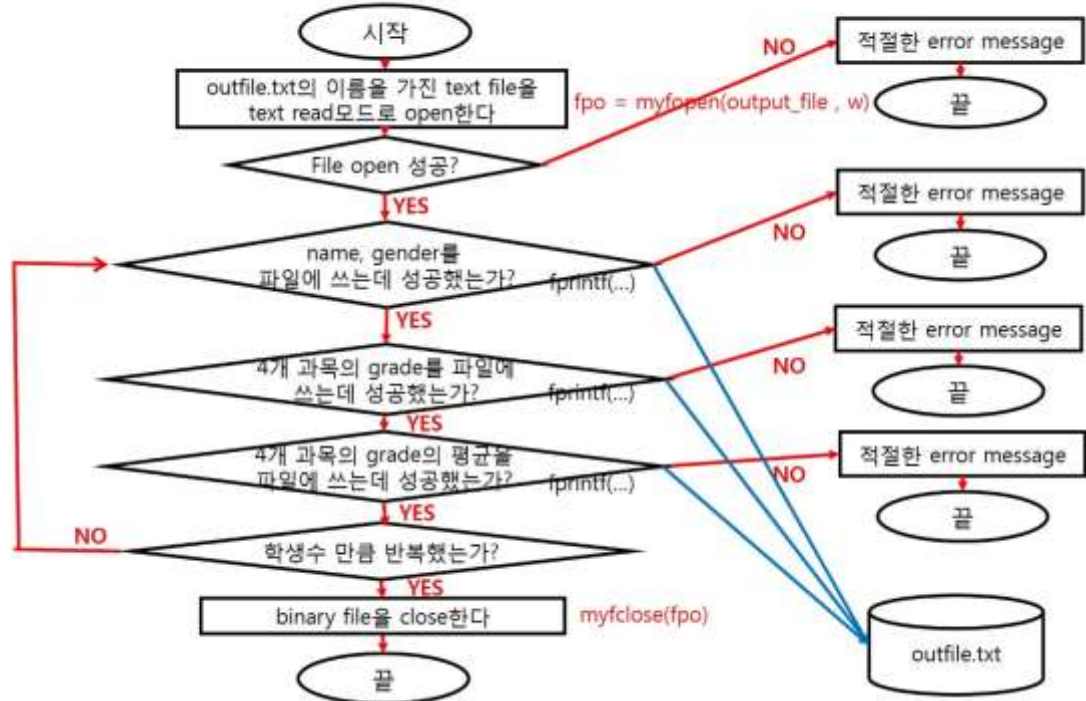
```
int readBIN (char *binary_file, int num_persons, int num_grades,
            char **name_array, char *gender_array,
            int **grades_array, double *average_array)
```





## 6. writeTXT 함수

```
int writeTXT (char *output_file, int num_persons, int num_grades,
char **name_array, char *gender_array,
int **grades_array, double *average_array)
```





### 3. Implementation

#### - Main 함수

Command line argument로부터 학생수, 과목수를 `get_positive_integer` 함수를 통해 `num_persons`, `num_grades`에 저장한다. 그 후, `buildMemory` 함수를 통해 메모리를 할당하고, `writeBIN` 함수를 통해 `infile.txt` 파일을 읽는다. 그 후 `writeBIN` 함수를 통해 이진 파일에 이진 형식으로 내용을 저장한다. 그리고 할당한 메모리를 `disposeMemory` 함수를 통해 돌려준다. 저장했던 이진 파일을 다시 `readBIN` 함수를 통해 읽어 여러 변수들에 저장한 후, `writeTXT` 함수를 통해 `outfile.txt` 파일에 내용들에 과목 평균을 더 추가하여 저장한다. 그리고 할당한 메모리를 `disposeMemory` 함수를 통해 돌려준다.

#### - disposeMemory 함수

input variable인 `char`자료형 더블포인터인 `name_array`, `char`자료형 포인터인 `gender_array`, `int`자료형 더블포인터인 `grades_array`, `double`자료형 포인터인 `average_array`를 `mymalloc`을 통해 할당했던 모든 메모리들을 돌려준다. 포인터가 가리키는 `array`가 2차원 `array`인 경우, `for` 구문을 통해 할당한 모든 메모리를 돌려준다.

#### - readTXT 함수

`infile.txt`의 내용을 읽어 `name_array`, `gender_array`, `grades_array`에 저장한다. `infile.txt`에는 이름, 성별, 4개 과목의 점수가 적혀있다. 이 텍스트 파일에서 한 줄씩 읽는데 만약 이 줄의 크기가 버퍼 크기를 벗어날 경우 적절한 `error`메시지를 출력하고 우아하게 프로그램을 종료한다. 이름 항목을 읽을 때에는 버퍼 크기만큼 읽는데, 이 버퍼의 크기는 [1]에서 200글자 정도가 적당하다는 내용을 참고하였다. 이도 마찬가지로 이름의 크기가 버퍼 크기를 벗어날 경우 적절한 `error`메시지를 출력하고 우아하게 프로그램을 종료한다. `infile.txt`의 내용이 주어진 형식에 맞춰 적혀있지 않은 경우에 적절한 `error`메시지를 출력하여 우아하게 프로그램을 종료한다.

- writeBIN 함수

readTXT를 통해 읽은 infile.txt의 내용을 토대로 이름의 길이, 4개 과목의 점수의 평균 항목을 더 추가하여 이진 형식으로 binary.dat에 저장한다. 파일을 이진 형식으로 적는 중에 실패할 시에는 적절한 error메시지를 출력한다.

- readBIN 함수

binary.dat의 크기를 ftell, fseek 함수를 통해 측정하고 콘솔창에 출력한다. 그 후, 이 파일의 내용을 이진 형식으로 읽어서 name\_array, gender\_array, grades\_array, average\_array에 저장한다. binary.dat을 읽는 도중에 파일이 잘못되어서 읽지 못한 경우에 적절한 error메시지를 출력하여 우아하게 프로그램을 종료한다.

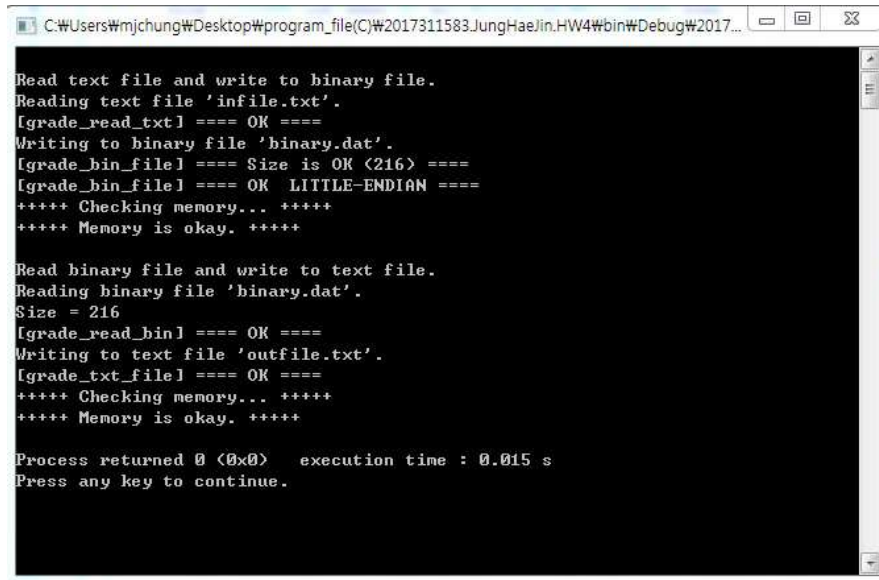
- writeTXT 함수

readBIN을 통해 읽은 binary.dat의 내용을 토대로 4개 과목의 점수의 평균 항목을 더 추가하여 outfile.txt에 저장한다.

## 4. Result

"grade.o" 파일과 링크하고, 프로그램 폴더에 "infile.txt" 파일을 넣은 후

"2017311583.정해진.HW4.c" 를 컴파일 후 실행한 결과이다.



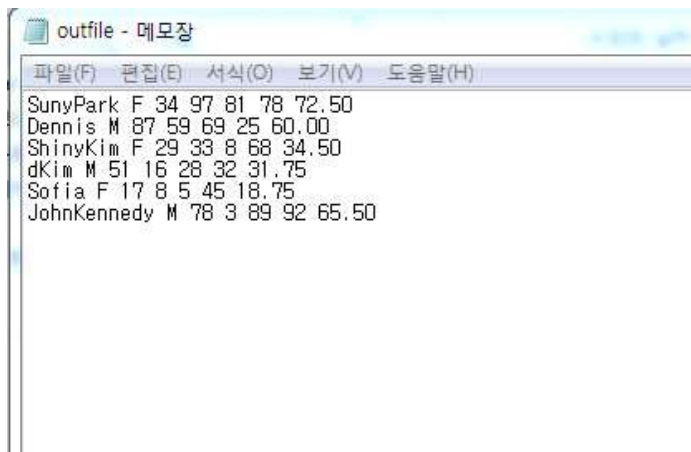
```
Read text file and write to binary file.
Reading text file 'infile.txt'.
[grade_read_txt] ==== OK ====
Writing to binary file 'binary.dat'.
[grade_bin_file] ==== Size is OK (216) ====
[grade_bin_file] ==== OK LITTLE-ENDIAN ====
+++++ Checking memory... +++++
+++++ Memory is okay. +++++

Read binary file and write to text file.
Reading binary file 'binary.dat'.
Size = 216
[grade_read_bin] ==== OK ====
Writing to text file 'outfile.txt'.
[grade_txt_file] ==== OK ====
+++++ Checking memory... +++++
+++++ Memory is okay. +++++

Process returned 0 (0x0) execution time : 0.015 s
Press any key to continue.
```

readTXT 함수가 잘 실행되었고, OK가 출력됐다. 다음으로 writeBIN 함수가 실행되어 binary.dat 파일에 text 파일의 내용이 저장되었고, 이 파일이 little-endian으로 쓰여졌음을 잘 출력하였다. 함수를 실행하면서 dynamic memory allocation을 했기 때문에, disposeMemory 함수를 통해 메모리를 반납하였고, 이 과정이 잘 이루어졌음을 출력하였다.

readBIN 함수로 binary.dat 파일을 읽어 필요한 수치들을 다 저장하고, writeTXT 함수를 통해 outfile.txt에 그 내용들을 다 저장하였다. 그 결과는 아래와 같다.



```
outfile - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
SunyPark F 34 97 81 78 72.50
Dennis M 87 59 69 25 60.00
ShinyKim F 29 33 8 68 34.50
dKim M 51 16 28 32 31.75
Sofia F 17 8 5 45 18.75
JohnKennedy M 78 3 89 92 65.50
```

## 5. Conclusion & Evaluation

실습을 통해 file을 text, binary 형식으로 다루는 방법을 잘 이해하게 된 것 같다.  
실습 2, 3에서 부족했던 부분을 다시 채워가는 과정이었다.

## 6. 참고 문헌

[1] Min, H. B. and SKKU, “Dynamic\_Memory.pdf”