Algorithm Design and Programming II

Lab 6 (30 Points)

Objectives:

- Learn how to use structure array and file I/O

Description:

createArray function

- Use the implementation of file I/O operations, you've done in prelab to read the given file.
- The first number in the file is the number of students.
- Create an array of **N** Student structures, using malloc().
- Store an integer number of students at the beginning of the array.
- Close your file.
- Each Student structure should look like as follows:

```
typedef struct {
        int id;
        float GPA;
} Student;
```

After updating your createArray(), your array should look like this:

Number of Students	student[0]	student[1]	student[2]	…	student[N-1]
N	**id**: 0 **GPA**: 2.34	**id**: 1 **GPA**: 2.99	**id**: 2 **GPA**: 3.01	…	**id**: N-1 **GPA:** 2.98

getAverageOfGPA function

- Compute and display your average GPA for this given student array.
- Return average of all students' GPA.

getNumOfStudents function

- Return the number of students that located in front of the array.

Main function steps:

- Call ***createArray*** function.
- Call **averageOfGPA** function.
- Display the result as **Example** below.
- Free the allocated array using your ***freeArray*** function.

Every user-defined function must have a comment describing:

- What function does;
- What parameter values are;
- What value it returns.

Example from the terminal window:

```
$ gcc main.c lab6.c -Wall -Werror
$ ./a.out
The average of N students' GPA is X.xx
```

Grading Criteria:

- Main program: 6 points
- createArray function: 12 points
- getAverageOfGPA function: 6 points
- getNumOfStudents function: 4 points
- freeArray function: 2 points

Note:

- If your code does not compile with **–Wall** and **–Werror**, you will receive a **zero** for this assignment.
- You need to finish at least **three** peer reviews within three days of this lab. Otherwise, you will get a 20% penalty.
- You will lose points if you don't have enough comments.