Lab 1- CS1

In this lab, you will work on a c-review code that uses structures, functions, and pointers. You should not use dynamic memory allocation for this lab. Additionally, we will learn how to use shell command to compile c code, execute it, and then passing input from an input file without using file i/o operations, and generating output to an output file. Note that in this process, your code will simply use standard input output using scanf and printf. Finally, we will also see how to compare the output generated with the correct output file.

Most of the parts will be discussed by the TA. Here is a problem description for the code. **Note, that the code will NOT use any Dynamic Memory allocation for this lab.**

Write a program that reads the ID number and grades for three courses of N (N<= 500) number of students. Then it displays all the students' ID numbers, grades for three courses, and the average grade of the student. In the end, it should also display the maximum average and the student who has achieved the maximum average. The average should be displayed up to 2 decimal places. The output also should be written to out.txt file.
You must have to use a structure array in your solution.

Some restrictions:

- Your structure name has to be **Student** and it should contain at least the fields **student_ID (int)**, and average (float), and other fields as needed. You should not use typedef.
- All the data must be read from a readData function with the following prototype, however, it should not deal with finding max. This function receives an empty array and a reference to an int and fill them up with the data from the input.

```
void readData(struct Student *students, int *c);
```

- To obtain the max average, please write a function with the following prototype:

```
struct Student getMaxAverageStudent (struct Student *s, int n);
```

This function takes an array of structure and its length and then returns the structure containing the highest average. You must call this function from main function after coming back from the readData function.

Sample input (must be a standard input/output. No file i/o)

The first line of the input contains the number of students N (N<=500). Then next N lines contain 4 integers where the first integer is the id number and the next three integers are the grades for three courses.

Sample input

```
5
861022 65 72 56
851102 78 45 80
860501 55 75 90
841205 75 80 95
850630 40 50 48
```

Sample output (must be standard output. No file i/o in your code. But use command to get the output into a file

861022 65 72 56 64.33
851102 78 45 80 67.67
860501 55 75 90 73.33
841205 75 80 95 83.33
850630 40 50 48 46.00
Maximum Average is 83.33 and the student is 841205 *(note that there will be a next line after this)*

The following list of commands will be very useful for your programing assignments as you will have to deal with large input and outputs. Entering all the inputs will be time consuming. So, using the commands to pass the data from an input file will make your testing much easier.
Compiling in testing in command line (You can use replit and use the shell that you will find just beside the Console):

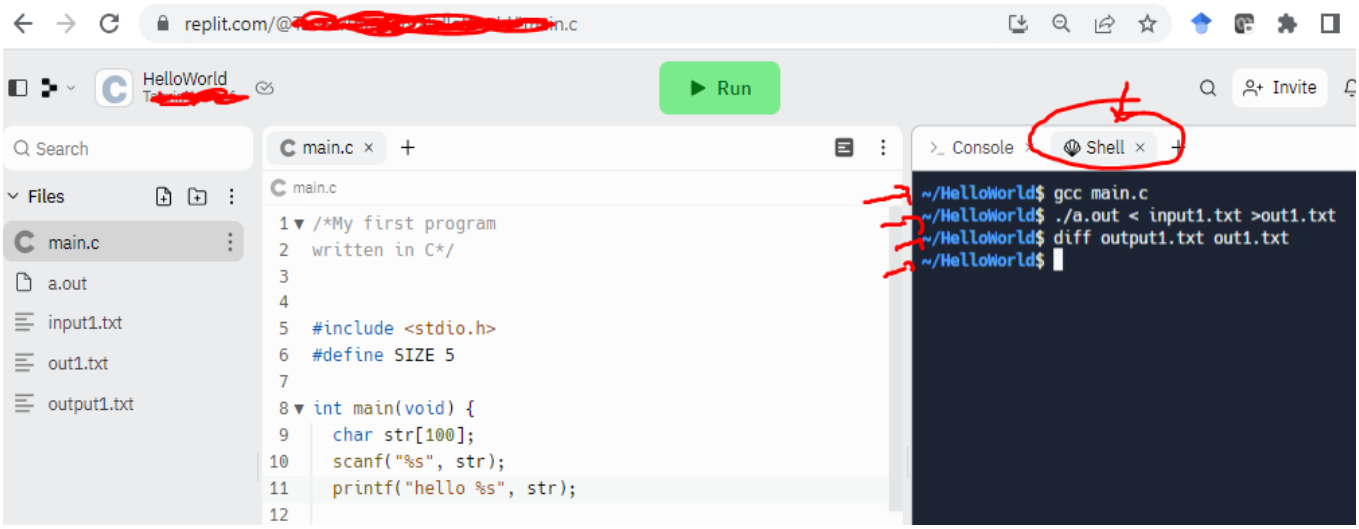
Use the following command for testing. (Don't put the $ sign. It is just an indicator that is a command)

$gcc main.c //this will compile your c file and generate a.out file as an executable file if your code compiles successfully

$./a.out < sample_in.txt >out.txt //if there is no error, run your code by this command. It will take input from sample_in.txt file and the output will be generated to out.txt file

You can use various commands to compare your output file against the given sample output file. Note that in your programming assignments you have to exactly match the output with the sample output to get credit.

$diff out.txt sample_out.txt //the result will be blank if there is no difference between the files
$ diff -c out.txt sample_out.txt //this command will show ! symbol to the unmatched lines.
$ diff -y out.txt sample_out.txt //very useful to compare the files side by side to see where exactly it is not matching



Submission:

The problem and most part of it will be discussed in the lab.

In addition, to solve the c review problem in lab1, the purpose of this submission is also to make yourself familiar with the Code grade submission system.

Please go through the video and tutorial on how to submit your assignments in code grades and how to see the feedback.

https://help.codegrade.com/for-students/getting-started/getting-started-in-canvasLinks to an external site.

You can work on your code using repl.it/code blocks/VS code/Xcode/ or any IDE that you prefer. Then save/download your file on your computer and submit or copy-paste the code from your IDE to the codegrade system.
After the submission, make sure your code passed the test cases in Codegrade.
There are some unit test cases that check some of the required functions automatically using unit testing. So, make sure you fulfill the requirements.

You need to submit three files:

- main.c
- main.h
- **commands.JPG (1 screenshot should show that you have used the command line shown above to compile and execute your code and the diff says no difference)**

The main.h file should contain the following:

```
struct Student {

//your structure
```

```
};


//function prototypes

struct Student getMaxAverageStudent(struct Student *s, int n);

void readData(struct Student *students, int *c);
```

Please include main.h file to your main.c file.

```
#include "main.h" //put this line in your main.c code right after including the header files.
```