uOttawa General Engineering (GNG)

GNG1106 - Lab 9

Objectives

1. Practise the use of strings.
2. Practise programming ASCII file IO.
3. Develop proficiency with structures, pointers , and functions involving structures and pointers.
4. Write program modules (functions) that satisfy specified requirements.

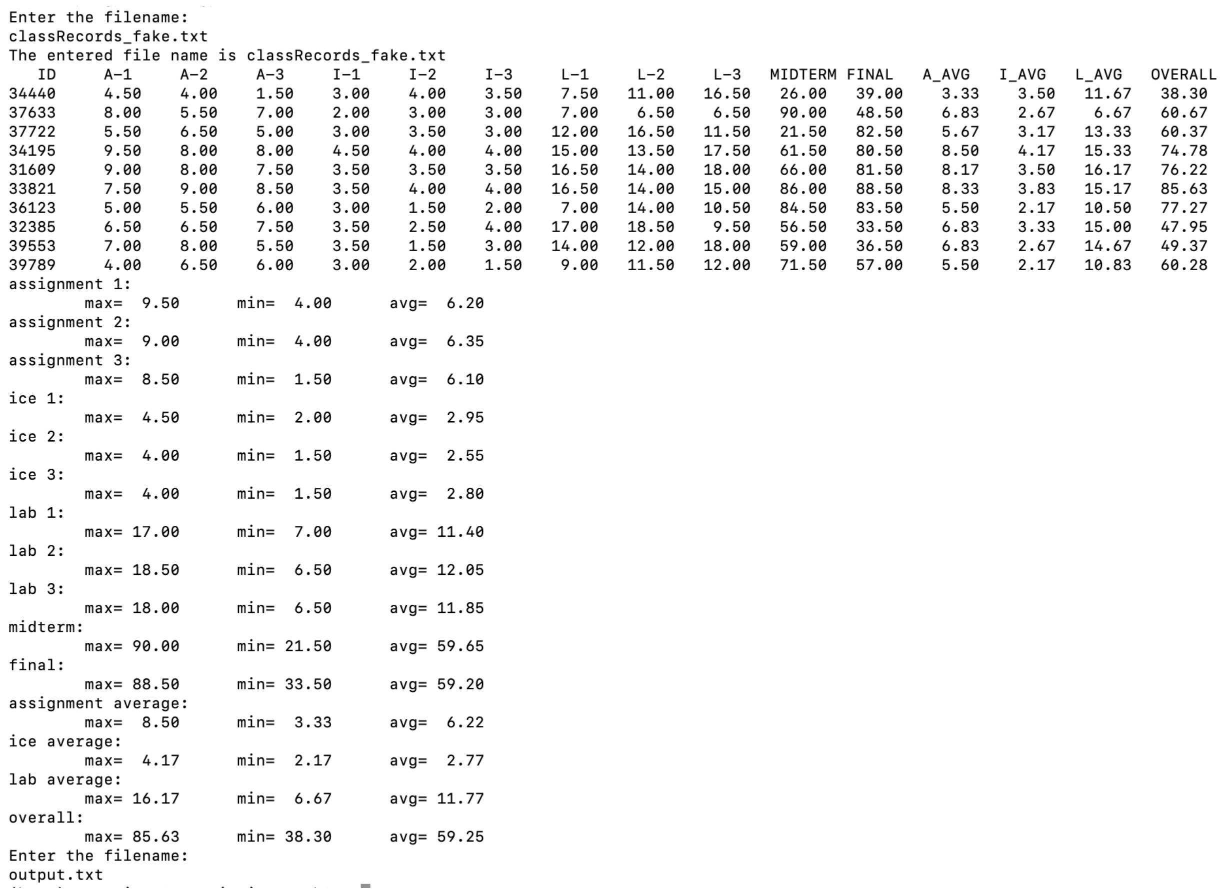
Introduction

The records of a class of students have been stored in the companion ASCII file “Lab9_classRecords.txt”. The key functions that are used to generate the file are given in the companion code “Lab9_functionsToWriteClassRecordFile.c”. Based on this information and the instruction given, you will need to complete a program which reads the class record file, processes the records, computes the statistics for various items in the student records and writes the statistics into a new ASCII file. The program to be completed is given in file “Lab9_codeTemplate.c”.

Instructions

You will need to download the class record file “Lab9_classRecords.txt” for your completed program to access. You need complete the program in the file “Lab9_codeTemplate.c” so that it performs the following tasks in order.

1. It asks the user to enter the name of class record file and immediately prints the entered file name to the screen.
2. It reads the class record file and prints its content on the screen.
3. It computes and prints the statistics of all items in the class records.
4. It asks the user to enter another file name, and save the computed class statistics to the file with this file name.



```
Enter the filename:
classRecords_fake.txt
The entered file name is classRecords_fake.txt
   ID     A-1     A-2     A-3     I-1     I-2     I-3     L-1     L-2     L-3   MIDTERM FINAL   A_AVG   I_AVG   L_AVG   OVERALL
34440     4.50    4.00    1.50    3.00    4.00    3.50    7.50   11.00   16.50   26.00   39.00    3.33    3.50   11.67   38.30
37633     8.00    5.50    7.00    2.00    3.00    3.00    7.00    6.50    6.50   90.00   48.50    6.83    2.67    6.67   60.67
37722     5.50    6.50    5.00    3.00    3.50    3.00   12.00   16.50   11.50   21.50   82.50    5.67    3.17   13.33   60.37
34195     9.50    8.00    8.00    4.50    4.00    4.00   15.00   13.50   17.50   61.50   80.50    8.50    4.17   15.33   74.78
31609     9.00    8.00    7.50    3.50    3.50    3.50   16.50   14.00   18.00   66.00   81.50    8.17    3.50   16.17   76.22
33821     7.50    9.00    8.50    3.50    4.00    4.00   16.50   14.00   15.00   86.00   88.50    8.33    3.83   15.17   85.63
36123     5.00    5.50    6.00    3.00    1.50    2.00    7.00   14.00   10.50   84.50   83.50    5.50    2.17   10.50   77.27
32385     6.50    6.50    7.50    3.50    2.50    4.00   17.00   18.50    9.50   56.50   33.50    6.83    3.33   15.00   47.95
39553     7.00    8.00    5.50    3.50    1.50    3.00   14.00   12.00   18.00   59.00   36.50    6.83    2.67   14.67   49.37
39789     4.00    6.50    6.00    3.00    2.00    1.50    9.00   11.50   12.00   71.50   57.00    5.50    2.17   10.83   60.28
assignment 1:
        max=  9.50      min=  4.00      avg=  6.20
assignment 2:
        max=  9.00      min=  4.00      avg=  6.35
assignment 3:
        max=  8.50      min=  1.50      avg=  6.10
ice 1:
        max=  4.50      min=  2.00      avg=  2.95
ice 2:
        max=  4.00      min=  1.50      avg=  2.55
ice 3:
        max=  4.00      min=  1.50      avg=  2.80
lab 1:
        max= 17.00      min=  7.00      avg= 11.40
lab 2:
        max= 18.50      min=  6.50      avg= 12.05
lab 3:
        max= 18.00      min=  6.50      avg= 11.85
midterm:
        max= 90.00      min= 21.50      avg= 59.65
final:
        max= 88.50      min= 33.50      avg= 59.20
assignment average:
        max=  8.50      min=  3.33      avg=  6.22
ice average:
        max=  4.17      min=  2.17      avg=  2.77
lab average:
        max= 16.17      min=  6.67      avg= 11.77
overall:
        max= 85.63      min= 38.30      avg= 59.25
Enter the filename:
output.txt
```

An example of the execution of the program is shown in the figure. Note that this example uses a different class record file ("classRecords_fake.txt") having a similar format as the one you will be using ("Lab9_classRecords.txt").

More detailed instructions are given in the file "Lab9_codeTemplate.c". You will need to implement various functions in the file, and add one line (and only one line) to the main function. **Feel free to design additional functions if you need.**

Deliverables

Pre-Lab Submission (20%): Ensure you have submitted your pre-lab before attending the lab session.

Deliverable 1 (10%): Define structure **`CLASS_STATS`**

Deliverable 2 (10%) Implement function

`void getFileNameFromUser(char *);`

and the missing line in the **`main`** function.

Deliverable 3 (20%): Implement functions

```
void getClassRecordsFromFile(STUDENT_REC *, int *, int *, int  *, int
*, char*);
```

and

```
void printClassRec(STUDENT_REC *, int, int, int, int);
```

Deliverable 4 (20%): Implement functions

```
void setClassStatsForItem(double *pMarks, int numOfStudents,
CLASS_STATS *s, char *description);
```

```
void getClassStatsForAllItems(STUDENT_REC *, int, int, int, int,
CLASS_STATS *, int *);
```

and

```
void printAllStats(CLASS_STATS *allStats, int numStats);
```

Deliverable 5 (20%): Implement function

```
void printAllStatsToFile(CLASS_STATS *, int, char *);
```

Submit only the completed code and the generated file in Deliverable 5. When checking out, run the code, show the TA the execution results and the content of the file generated.

Check out and Submission

You must check out with your TA before submitting the deliverables. During the check out, your TA may inspect your work and to-be-submitted deliverables, and ask you questions to further check your understanding. At the end of the check out, your TA will give you an initial mark for the in-lab component of this lab and let you know. You must then submit the deliverables before the due time of this lab. While this initial mark is likely to be the final, your TA reserves the right to reduce this initial mark after checking more carefully the deliverables you submit

Grading Criterion

- **Correctness (80%):** Correct syntax, logic and execution.
- **Style (20%):** Descriptive naming, appropriate indentation, ease of reading.