10.06.2022

Istanbul Technical University
Faculty of Computer and Informatics Engineering
BLG 102E - Int to Scientific&Eng.Computing (C)
Spring 2021-2022

# Final Exam

- Final exam will take **75 minutes** and it will be held in ITU computer laboratories.
- The questions will be announced through Ninova. **Submit your solutions to the corresponding Ninova announcements**: The solution for Q1 must be submitted to the **Final – Q1**, whereas the solution for Q2 must be submitted to the **Final – Q2**.
- Make sure your source code is compiled and linked successfully, and passed all the tests given with the questions before your submission.
- You are **not** allowed to use any data structures or types that have not been taught in the lectures.
- Remember to use **–std=c99 –Wall –Werror** flags while compiling with gcc.

**Q1. (40 pts)** The program "fq1.c" is intended to take a number of people names as command line arguments, and print a summary of statistics regarding the number of male and female names, as well as the average lengths of names for males and females. The number of names that the program expects could be at least 1 and at most 5. Assume that female names start with "F_" and male names start with "M_" prefixes. While computing the lengths of the names, please ignore prefixes. Please modify the program so that it will work per the above specifications. Here are some sample output of the program:

```
>./fq1
Enter 1 to 5 arguments!

>./fq1 ali veli
Names should start with either M_ or F_!

>./fq1 M_ali F_ayse M_veli
Female count: 1, avg. female name length: 4.0
Male count: 2, avg. male name length: 3.5

>./fq1 M_ali M_veli
Female count: 0, avg. female name length: 0.0
Male count: 2, avg. male name length: 3.5

>./fq1 M_ali F_ayse M_veli M_ali M_veli F_ayse
Enter 1 to 5 arguments!
```

Use "test_fq1.t" file to test your program using calico. (python -m calico.cli fq1.t).
Don't forget to submit your code through Ninova.

**Q2. (60 pts)** Check the program in "fq2.c", and implement the bodies for **search** and **insert** functions in fq2.c to accomplish the following features. Do not change any other part of the provided code file.

This program will allow storing and searching student records in a file. The program will have the following features.

- The name of the file that the program will be reading and writing will be provided by the user through standard input stream. If the user-provided file name refers to an existing file name, then the program should not overwrite the existing content, and add any new records to the end of the file. If the user-provided file name refers to non-existing file, when the user wants to insert new records, the file should be created automatically. Otherwise, for searching requests, the file should not be automatically created (if it does not exist), and the program should report that the file does not exist (see the sample output).
- The user can perform 3 operations:
  Enter 1 to search for data, 2 to insert student record (0 to exit)
- The user can search for students by their id. The program should report all matching students (in their insertion order) with the given id. Assume that different students may have the same id.
- When the user chooses to insert a student record, the program should ask for student id, name, and gpa. Assume that the provided names would always contain less than 50 characters.
- The program should write and read records as instances of student struct which is already provided in the fq2.c program file.

Here are sample runs of the program:

- Run 1 (the provided file name does not exist):

```
>./fq2
Enter data file name: sample_data
Enter 1 to search for data, 2 to insert student record (0 to exit): 1
   Enter student id to search: 1
Could not open file!
>
```

- Run 2 (the provided file name does not exist):

```
>./fq2
Enter data file name: sample_data
Enter 1 to search for data, 2 to insert student record (0 to exit): 2
   Student id: 1
   Student name: ali
   Student gpa: 2.4
Enter 1 to search for data, 2 to insert student record (0 to exit): 2
   Student id: 1
   Student name: veli
   Student gpa: 3.2
Enter 1 to search for data, 2 to insert student record (0 to exit): 1
   Enter student id to search: 1
   1. Name: ali, GPA: 2.40
   2. Name: veli, GPA: 3.20
Found 2 occurrences!
Enter 1 to search for data, 2 to insert student record (0 to exit): 0
>
```

- Run 3 (the provided file name exists from previous runs):

```
>./fq2
Enter data file name: sample_data
Enter 1 to search for data, 2 to insert student record (0 to exit): 1
   Enter student id to search: 1
   1. Name: ali, GPA: 2.40
   2. Name: veli, GPA: 3.20
Found 2 occurrences!
Enter 1 to search for data, 2 to insert student record (0 to exit): 2
   Student id: 1
   Student name: ayse
   Student gpa: 3.5
Enter 1 to search for data, 2 to insert student record (0 to exit): 1
   Enter student id to search: 1
   1. Name: ali, GPA: 2.40
   2. Name: veli, GPA: 3.20
   3. Name: ayse, GPA: 3.50
Found 3 occurrences!
Enter 1 to search for data, 2 to insert student record (0 to exit): 0
>
```