

CPE 100 Computer Programming for Engineers
Final Examination 1/2021
Part 2: File Processing (100 points)

Instructions

Create a program called `readwriteInfo.c`. This program should read all the data from the given binary file ***info.dat***, into an array of ***studentInfo*** structures. The program should print the information for each student on the screen. Then it receives input your id for searching in the structure array. After that, the program should write your information into another text file. The output text file must be named by your id such as `640705034xx.txt`.

Details

1. (10 points) Declare a structure called ***studentInfo***. This structure must contain *id*, *name*, *num1*, and *num2*. In the main function, You must define an array of ***studentInfo*** structure called ***studentList***. Moreover, a ***studentInfo*** structure, called ***myInfo***, must be declared.
2. (30 points) Create ***getStudent*** function for reading all data from ***info.dat*** into an array of ***studentInfo*** structures. This function must return the number of students. An example for calling this function is

n = getStudent(studentList);

where *n* is the number of students,

studentList is the array of ***studentInfo***.

Important note: If you cannot read binary file ***info.dat***, you should try to read ***info.txt*** instead. However, your score will be reduced by 10 points for using ***info.txt***.

3. (10 points) Create ***displayStudent*** function for printing all the information in ***studentInfo*** structure in some nice format. An example for calling this function is

displayStudent(studentList,n);

4. (20 points) In the main function, the program should prompt the user to enter a student id. The input student id must be stored to the variable *id* in ***myInfo*** structure. Then create ***searchID*** function for searching input id in the ***studentList*** records. After your id is found, the data in ***studentList*** must be copied to ***myInfo***. The program should return the line number that your id is found in the ***studentList*** records. An example for calling this function is

line = searchID(studentList, &myInfo, n);

where *line* is the line number.

Important note: If binary or text file are not read, you can assign all data into ***studentList*** by

```
studentInfo studentList[] = {{id_1, name_1, num1_1, num2_1},  
{id_2, name_2, num1_2, num2_2},  
{id_3, name_3, num1_3, num2_3},  
...,  
{id_n, name_n, num1_n, num2_n}}
```

Your score will not reduce if you assign **studentList** directly.

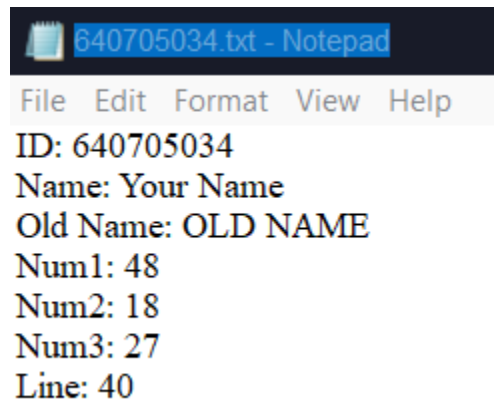
5. (30 points) Create **exportInfo** function for writing your data into a text file. The output text file must be named by your id such as **640705034xx.txt**. In this function, you must calculate the num3 by

$$num3 = 0.3*num1+0.7*num2$$

The contents of your text file must contain the following

- Id: your student id.
- Name: your name.
- Old name: old name that you found when searching your id.
- Num1: the first number that you found when searching your id.
- Num2: the second number that you found when searching your id.
- Num3: the number that this function computes.
- Line: the line number that you found when searching your id.

An example of contents in your text file is shown below.



An example for calling **exportInfo** function is

exportInfo(myInfo,line);

Important note: If **myInfo** is not retrieved from searching in **studentList** records, you can use your data in text file **dat.txt** and write your data to output text file as mention in this step. Your score will not reduce if step 4 is not completed.

6. Upload your C source file and the output text file to LEB2.