CMPE 252 C Programming, Spring 2023

Lab 5

In this lab, you are given a binary file on LMS named as **researcher.bin** which includes records of researchers in a university. Each researcher record will be stored using `researcher` struct as:

```
typedef struct
{
    unsigned int id;         // researcher id
    char name[20];          // researcher name
    char surname[20];      // researcher surname
    char department[20]; // department of researcher
    int citIndex;        // citation index of the researcher
} researcher;
```

researchers.bin file consists of 100 records. 94 of them are blank records. The other 6 records (not blank) are placed in specific positions of the binary file based on their ids. For example, if id of a researcher is 5, its record is the fifth record in the file. The size of each record is equal to the size of `researcher` struct. The records are sorted according to their id number.

Complete the skeleton code **lab5_v1_skeleton.c** on LMS by implementing the following 4 functions:

Part I (25 points)

`int modifyCitIndex(FILE *filePtr, unsigned int id, int increaseCit);`

Takes `FILE` pointer to the binary file. Updates citation index of the researcher whose id is provided in `id` parameter with the given `increaseCit` value. If there is a researcher record with the given `id`, its `citIndex` field is updated by summing up with the given `increaseCit` value and the function returns 1; otherwise, it returns 0.

For sample run, see `test_case_2.txt` on LMS.

Part II (25 points)

`int insertResearcher(FILE *filePtr, unsigned int id, char name[], char surname[], char department[], int citIndex);`

Takes `FILE` pointer to the binary file. Inserts a researcher record for which all the information is provided via the parameters of the function. If there is already a researcher record with the given `id`, the function returns 0; otherwise, it adds a new researcher record and returns 1.

For sample run, see `test_case_4.txt` on LMS.

Part III (25 points)

`int removeResearcher(FILE *filePtr, unsigned int id);`

Takes `FILE` pointer to the binary file. Removes the record of the researcher whose id is provided in the parameter `id` by setting its fields to `{0, "", "", "", 0}` (i.e. lazy deletion approach). If there is a researcher record with the given `id`, it is removed and the function returns 1; otherwise, it returns 0.
For sample run, see `test_case_6.txt` on LMS.

Part IV (25 points)

`int viewDepartmentCits(FILE *filePtr, char department[], int maxCit);`

Takes `FILE` pointer to the binary file. Prints researcher records whose department field is the same as the parameter `department` and also `citIndex` field is less than or equal to the given parameter `maxCit` and returns the count of printed researcher records.
For sample run, see `test_case_8.txt` on LMS.

Important Notes

As a small **hint**: `fseek()` is used to move file pointer associated with a given file to a specific position. You need to use `fseek()` function to add, update and delete any record in binary file.

Note that parts are independent from each other so solution of each part does not require solution of other parts. Please check and see all the remaining VPL test cases on LMS while submitting.

Notice that, in the skeleton code, we have provided implementation of the following functions which are needed to remain as they are:

`int main();`

Opens the binary file **`researchers.bin`** for read and update (`rb+`). Shows all records. Asks for choice of the operation to be done, calls the corresponding function, and either shows all records or prints a message based on the value returned from the function call.

`void showRecords(FILE *filePtr);`

Takes `FILE` pointer to the binary file and prints all researcher records in it.