CMPE 252 C Programming, Spring 2023

Lab 5

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

```
typedef struct
{
    unsigned int id;      // actor id
    char name[20];       // actor name
    char surname[20];   // actor surname
    char major[20];    // major of actor
    int repIndex;     // reputation index of the actor
} actor;
```

actor.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 an actor is 5, its record is the fifth record in the file. The size of each record is equal to the size of `actor` struct. The records are sorted according to their id number.

Complete the skeleton code **lab5_v2_skeleton.c** on LMS by implementing the following 4 functions:

Part I (25 points)

`int modifyRepIndex(FILE *filePtr, unsigned int id, int decreaseRep);`
Takes `FILE` pointer to the binary file. Updates reputation index of the actor whose id is provided in `id` parameter with the given `decreaseRep` value. If there is an actor record with the given `id`, its `repIndex` field is updated by subtracting with the given `decreaseRep` 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 insertActor(FILE *filePtr, unsigned int id, char name[], char surname[], char major[], int repIndex);`
Takes `FILE` pointer to the binary file. Inserts an actor record for which all the information is provided via the parameters of the function. If there is already an actor record with the given `id`, the function returns 0; otherwise, it adds a new actor record and returns 1.
For sample run, see `test_case_4.txt` on LMS.

Part III (25 points)

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

Takes `FILE` pointer to the binary file. Removes the record of the actor whose id is provided in the parameter `id` by setting its fields to `{0, "", "", "", 0}` (i.e. lazy deletion approach). If there is an actor 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 viewMajorReps(FILE *filePtr, char major[], int minRep);`

Takes `FILE` pointer to the binary file. Prints actor records whose department field is the same as the parameter `major` and also `minRep` field is greater than or equal to the given parameter `minRep` and returns the count of printed actor 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 **`actor.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 actor records in it.