LDBMS
[Library Database Management System]

In this project you will build and manage partial database of FAST library. The software will manage

- Inventory of all issuable and reference books (course books as well as general books). If a book is issued, its status will be updated immediately. Assume that the following information about books is stored in a text file **books.dat.** Each record is on the new line and each field of one record is separated by , (comma). The file books.dat has records stored in ascending order of Accession number.
    - Author
    - Co-author. Will be **Noname** if no 2nd Author.
    - Title
    - Publisher
    - Year of publication
    - Subject (CS, Management, Mathematics, Social Sciences etc).
    - Accession Number
    - Book Issuable /Reference Book
    - Book Issued/Not Issued (if issuable)
    - Date of Issuance of Book (if issued). Will be **00/00/0000** (if not issued).
    - Issued to Employee-ID/Student-ID (if issued). Will be **0** (if not issued).
    - Return date (if issued). 15 days for students, 4 months for faculty/staff. Will be **00/00/0000** (if not issued).

- Complete updated record of all the users of the library (includes faculty, staff, students). Stored in a file with the name **users.dat**
    - Name
    - Father's Name
    - Rank (Faculty OR Staff OR Student)
    - Department
    - ID (Student-ID OR Employee-ID)
    - Date of Birth
    - Address
    - Cell number

- If a student completes his/her graduation, his/her account will be deleted and for new admitted students, the account will be created (in the file user.dat). A similar scenario for new staff/faculty etc.
- Issuance of all the issuable books (different duration time and fine for students/faculty/staff).

Develop a menu-driven database which is capable of doing the following:

1. Insert a book (the file books.dat will be updated). The record of new book will be placed so that the ascending order (based on Accession number) is not disturbed.
2. Delete a book (the file books.dat will be updated).
3. Issue a book (the file books.dat will be updated).
4. Book is returned (the file books.dat will be updated). Fine will be calculated and displayed if valid.
5. Search a book (and display its complete record).
    a. Search through Author's name.
    b. Search through co-author's name.
    c. Search through Accession number.
6. Ascending Sort the book data
    a. Sort based on Author's names, and store the records in a new file.
    b. Sort based on Book Title, and store the data in a new file.
7. Display all books that
    a. Written by a specific author.
    b. Belong to a particular subject (e.g., mathematics)

8. Edit an existing contact (the file users.dat will be updated). e.g., the address of student etc. is changed.
9. Search a user (and display complete record on screen including the book titles issued to him). Age of the user must be displayed as well.
    a. Search through name
    b. Search through ID
    c. Search through cell number
10. Exit

Your program should be menu driven. The program should exit if the user choice is 10.

Directions:

- **a)** All the books are to be stored in a file, **books.dat** (this file contains all the records sorted with respected to the first name).
- **b)** You will have to update the existing file if a record is inserted, deleted or edited.
- **c)** If searching is applied, the record(s) should be displayed on the output screen only.
- **d)** In case of sorting, new file should be created (the name of the new file should be entered by the program user).
- **e)** All the fields of a single record are comma separated (no commas otherwise except in address).
- **f)** You are not allowed to use data type **string** in any case.
- **g)** Make separate functions to perform each task.
- **h)** Inputs are to be taken in the main function only; `cin` cannot be used in any function other than `main()`.

Note:

1. An address may include comma too, e.g. 123-H1, Johar Town, Lahore.
2. A cell phone number will only start with a 0.
3. Middle name cannot be entered in the record (Author name or user name).
4. None of the entry can be skipped.
5. Maximum number of records can be 1000.