C program for the management of a reservation system

Objectives of the project

- Validate the skills of the course

- Learn to use structures

- Handle files

What is to be handed in :

The complete source program with extensive comments
A tested and operational executable
A report presenting the work done: (4 pages maximum)
1. Organisation of the program: division into functions, role of these functions, explanation of the program
2. Instructions for using the program
3. Qualitative assessment of the work, difficulties encountered, etc.

Evaluation of the project

It is based on the following elements:

- The source program:

  o Respect of the statement, originality of the proposed theme

  o Technical quality of the program: division into functions, instructions, algorithms, efficiency, error management, etc.

- Presentation of the program: indentation, comments and naming
- Documentation provided

  o Organisation of the program and its instructions for use

- Support of the work

Timetable and marking

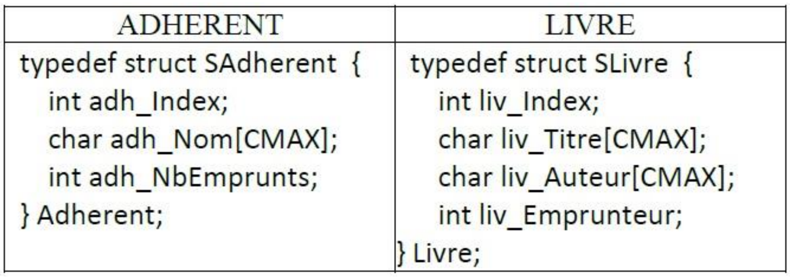
The projects are due on April 14, 2023, noon.

Presentation of the project

You will have to create a program to manage a reservation system for any object: books, video cassettes, hotel rooms, cars, school resources (classrooms, teaching...), etc. The examples given below concern the management of book loans in a library: you will have to customise your programme according to your choice.

It is up to you to choose a management system: be imaginative and creative

The model to be implemented will contain two different structures, whose fields contain at least the information given below. You will need to add fields to these structures depending on the booking system you have chosen.



ADHERENT	LIVRE

typedef struct SAdherent {
int adh_Index;
char adh_Nom[CMAX];
int adh_NbEmprunts;
} Adherent;
	typedef struct SLivre {
	int liv_Index;
	char liv_Titre[CMAX];
	char liv_Auteur[CMAX];
	int liv_Emprunteur;
	} Livre;

adh_Index (resp. liv_Index) uniquely identifies a member (resp. a book). It is up to you to manage the notion of unique number. liv_Borrower contains the value of adh_Index of the borrower of the book. The number of books borrowed by a member is stored in adh_NbBorrowed (a member can therefore borrow several books).

Programming constraints

The main() function must contain an array of Member structures and an array of Book structures (these are not global variables). You will have to save the data of these structures in two files (Look for suitable functions)

Your program should have a menu like this:

(1) Member management
- Add, modify or delete a member
- Display the list of members in alphabetical order

(2) Book management
- Add, modify or delete a book
- Display the list of books in alphabetical order (title)

(3) Borrowing management
- Borrow a book
- Return a book
- Display the list of borrowed books
- Display the list of book borrowers

(4) Exit the program

Implementation notes

- Deleting an element from the array involves shifting the "free box" to the end of the array to remove the "holes" in the array.

- Arrays must be passed by parameter to the various functions of the program.