Library

Implement a system to manage the book retrieval of a public library. The system must meet the following requirements.

R1: Book information

When a **`Book`** object must be created, the following info is provided: author and title.
These properties are accessible through corresponding *getter* methods.
The **`toString()`** conversion method provides a textual representation of the object according to the following format: "author, title" (comma plus space).

R2: Adding books to the library

The **`Library`** is organized according to the following structure:

- it has three floors
- Each floor contains 30 closets (closets are identified by codes like "C1", which are unique within the same floor only)
- Each closet contains 6 shelves
- Each shelf contains up to 10 books

The **`Library`** class provides methods to **`add()`** a book in a given position and to find out whether a given shelf (in a given closet on given floor) **`contains()`** a specified book.

R3: Getting all books in a closet

Given the name of a closet (on a given floor), the **`getBooks()`** method of class **`Library`** returns a string representing the content of the closet. The string lists all contained books, grouped by shelf (i.e., for *i-th* shelf, the returned string contains the header "`Shelf i`", which is followed by the list of books, one per line).

R4: Getting a book position

Given a **`Book`** object, it is possible to obtain its position in the library. To this aim, the **`getFloor()`**, **`getCloset()`**, and **`getShelf()`** methods are provided by class **`Book`**.

R5: Searching for a book

The **`Library`** provides a **`find()`** method to look for a specific book, given its author and title.