Section: A1+A2 **Time Limit: 50 minutes**

Library Management System

Problem Statement

You are tasked with building a Library Management System where:

- A library can have multiple branches.
- Each **branch** can store multiple **books**.

Your system should allow:

1. **Adding library branches**.
2. **Adding books to branches**.
3. **Displaying details of books in each branch**.
4. **Borrowing a book from a branch by its ID**.
5. **Returning a book to a branch by its ID**.

Functionality

Initial I/O

- The program should read the **number of branches** as a **command-line argument**.
- For each branch:
  - Assign a **unique branch ID** (starting from 1).
  - Ask the user how many books to add to this branch.
  - For each book, get the following inputs:
    - **Book ID** (integer)
    - **Book Title** (string)
    - **Availability** (true/false)

User Menu (Loop)

The program should then repeatedly show a menu to the user:

```
Menu:
1. Borrow a Book
2. Return a Book
3. Display All Branch Info
4. Exit
```

Option 1: Borrow a Book

- Prompt the user to enter **Branch ID** and **Book ID** to borrow.
- If the book is **available**, update its status and show a success message.
- If the book is **not available**, inform the user accordingly.

Option 2: Return a Book

- Prompt the user to enter **Branch ID** and **Book ID** to return.
- If the book was borrowed, mark it as available and confirm the return.
- If the book is already available, inform the user.

Option 3: Display All Branch Info

- Display the details of all library branches and their books:
  - For each branch, print its **Branch ID**.
  - List all books in that branch with their details (Book ID, Title, Availability).
  - If a branch has no books, print:

    ```
    No books available.
    ```

Option 4: Exit

- End the menu loop and terminate the program.

Constraints

- The system should limit:
  - A **maximum of 10 branches**.
  - A **maximum of 20 books per branch**.
- If these limits are exceeded, show an appropriate message.

Classes to Implement

You are free to design the attributes and methods of each class as you see fit, but your solution should include **only** the following classes:

1. **Book**
   Represents a book in the library.
2. **LibraryBranch**
   Represents a branch of the library that contains books.

3. **Main**
   Contains the `main` method, handles user input/output, and controls program flow. The `Main` class should **not store any information** about library branches or books.

Submission Guidelines

- Create a folder named by your ID.
- Move all `.java` files in the folder.
- Zip the folder and submit the zip file.