**Problem Statement**

You are tasked with creating a **Library System** where:

1. Each library has a collection of **Books**.
2. Each book is written by one or more **Authors**.
3. Two books are considered **equal** if they have:
   * The same **title**.
   * The same set of **authors** (order doesn't matter).

Your task is to:

1. Implement a Library class that uses **composition** to manage a list of Book objects.
2. Implement a Book class that uses **composition** to manage a list of Author objects.
3. Override the equals() method in the Book class to:
   * Compare the title and the **set** of authors.
4. Write a program to:
   * Input multiple libraries, each with its own books and authors.
   * Identify and display **duplicate books** within a library based on the overridden equals() method.
   * Identify and display the **common books** between two libraries.

**Input Format**

1. The first input specifies the number of libraries, n.
2. For each library:
   * The number of books.
   * For each book: The title and the number of authors.
   * For each author: The name and email.

**Output Format**

1. List of duplicate books in each library.
2. List of common books between two libraries.

**Sample Input**

2

Library 1:

3

"Harry Potter" 2

J.K. Rowling rowling@hogwarts.com

John Doe john.doe@example.com

"The Hobbit" 1

J.R.R. Tolkien tolkien@middleearth.com

"Harry Potter" 2

J.K. Rowling rowling@hogwarts.com

John Doe john.doe@example.com

Library 2:

2

"Harry Potter" 2

J.K. Rowling rowling@hogwarts.com

John Doe john.doe@example.com

"1984" 1

George Orwell orwell@dystopia.com

**Sample Output**

Duplicate Books in Library 1:

"Harry Potter"

Common Books Between Library 1 and Library 2:

"Harry Potter"

**Assignment 2: Company and Employee Management**

**Problem Statement**

You are tasked with creating a **Company Management System** where:

1. Each company has multiple **Departments**.
2. Each department has multiple **Employees**.
3. Employees across departments in the same company are considered **equal** if they have:
   * The same **employee ID**.
   * The same **name**.

Your task is to:

1. Implement a Company class that uses **composition** to manage a list of Department objects.
2. Implement a Department class that uses **composition** to manage a list of Employee objects.
3. Override the equals() method in the Employee class to:
   * Compare the employee ID and name.
4. Write a program to:
   * Input a company with multiple departments and employees.
   * Identify and display **duplicate employees** within the company (even if they are in different departments).
   * Display the **total number of unique employees** in the company.

**Input Format**

1. The first input specifies the number of departments, d.
2. For each department:
   * The department name.
   * The number of employees.
   * For each employee: The employee ID, name, and role.

**Output Format**

1. List of duplicate employees within the company.
2. Total number of unique employees in the company.

**Sample Input**

Company: TechCorp

3

Department: HR

2

101 John Doe Manager

102 Jane Smith Recruiter

Department: Engineering

3

103 Alice Brown Developer

101 John Doe Team Lead

104 Mark Johnson Developer

Department: Sales

2

102 Jane Smith Executive

105 Emily Davis Salesperson

**Sample Output**

Duplicate Employees in TechCorp:

John Doe

Jane Smith

Total Unique Employees in TechCorp: 4