**Data Structures Algorithms**

**EXERCISE 6: LIBRARY MANAGEMENT SYSTEM**

**Source Code**

// Main.java

import java.util.\*;

// Step 2: Book class

class Book {

int bookId;

String title;

String author;

public Book(int bookId, String title, String author) {

this.bookId = bookId;

this.title = title.toLowerCase(); // For case-insensitive search

this.author = author;

}

public void display() {

System.out.println("ID: " + bookId + ", Title: " + title + ", Author: " + author);

}

}

public class Main {

// Step 3a: Linear Search by title

public static void linearSearch(List<Book> books, String searchTitle) {

System.out.println("Linear Search Result for: " + searchTitle);

boolean found = false;

for (Book b : books) {

if (b.title.equalsIgnoreCase(searchTitle)) {

b.display();

found = true;

}

}

if (!found) {

System.out.println("No book found with title: " + searchTitle);

}

System.out.println();

}

// Step 3b: Binary Search by title (requires sorted list)

public static void binarySearch(List<Book> books, String searchTitle) {

System.out.println("Binary Search Result for: " + searchTitle);

int left = 0, right = books.size() - 1;

boolean found = false;

searchTitle = searchTitle.toLowerCase();

while (left <= right) {

int mid = (left + right) / 2;

String midTitle = books.get(mid).title;

int cmp = searchTitle.compareTo(midTitle);

if (cmp == 0) {

books.get(mid).display();

found = true;

break;

} else if (cmp < 0) {

right = mid - 1;

} else {

left = mid + 1;

}

}

if (!found) {

System.out.println("No book found with title: " + searchTitle);

}

System.out.println();

}

public static void main(String[] args) {

List<Book> books = new ArrayList<>();

books.add(new Book(101, "The Alchemist", "Paulo Coelho"));

books.add(new Book(102, "1984", "George Orwell"));

books.add(new Book(103, "To Kill a Mockingbird", "Harper Lee"));

books.add(new Book(104, "Pride and Prejudice", "Jane Austen"));

books.add(new Book(105, "Moby Dick", "Herman Melville"));

// Linear Search (unsorted list)

linearSearch(books, "1984");

// Sort books by title for binary search

books.sort(Comparator.comparing(b -> b.title));

binarySearch(books, "Pride and Prejudice");

// Try searching for a non-existing title

linearSearch(books, "Invisible Man");

binarySearch(books, "Invisible Man");

// Step 4: Analysis

System.out.println("Analysis:");

System.out.println("- Linear Search: O(n) time complexity, no need for sorted data.");

System.out.println("- Binary Search: O(log n) time complexity, requires data to be sorted.");

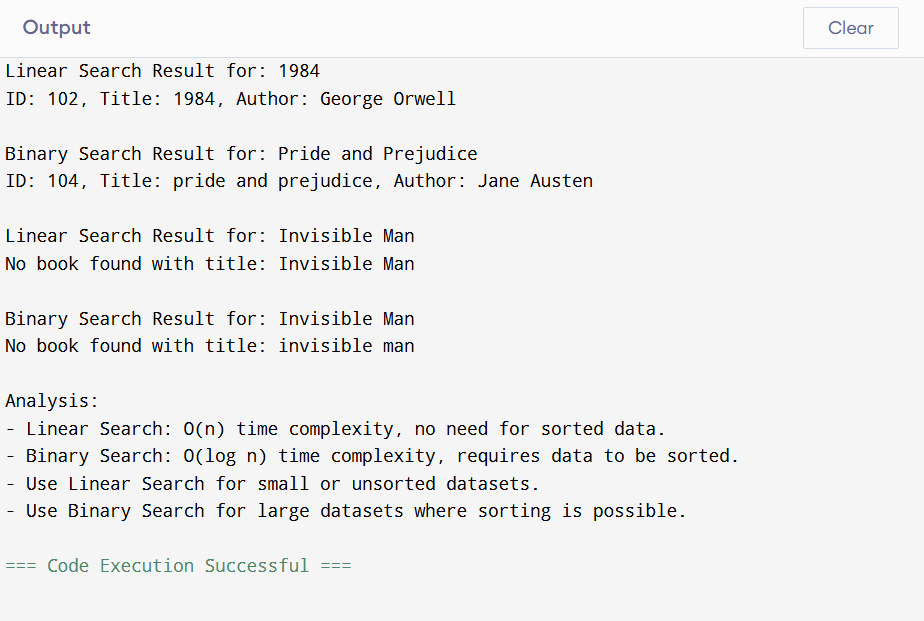
System.out.println("- Use Linear Search for small or unsorted datasets.");

System.out.println("- Use Binary Search for large datasets where sorting is possible.");

}

}

**OUTPUT**

****