**Data structures and Algorithms**

**Exercise 2: E-commerce Platform Search Function**

**Scenario:**

You are working on the search functionality of an e-commerce platform. The search needs to be optimized for fast performance.

Code

import java.util.\*;

public class ECommerceSearch {

public static class Product {

int productId;

String productName;

String category;

public Product(int productId, String productName, String category) {

this.productId = productId;

this.productName = productName;

this.category = category;

}

@Override

public String toString() {

return "Product(ID=" + productId + ", Name='" + productName + "', Category='" + category + "')";

}

}

public static Product linearSearch(Product[] products, String targetName) {

for (Product product : products) {

if (product.productName.equalsIgnoreCase(targetName)) {

return product;

}

}

return null;

}

public static Product binarySearch(Product[] products, String targetName) {

int low = 0;

int high = products.length - 1;

while (low <= high) {

int mid = (low + high) / 2;

int compare = products[mid].productName.compareToIgnoreCase(targetName);

if (compare == 0) {

return products[mid];

} else if (compare < 0) {

low = mid + 1;

} else {

high = mid - 1;

}

}

return null;

}

public static void main(String[] args) {

Product[] products = {

new Product(1, "Laptop", "Electronics"),

new Product(2, "Shampoo", "Personal Care"),

new Product(3, "T-shirt", "Clothing"),

new Product(4, "Book", "Education"),

new Product(5, "Headphones", "Electronics")

};

System.out.println("=== Linear Search ===");

Product foundLinear = linearSearch(products, "Book");

System.out.println(foundLinear != null ? foundLinear : "Product not found");

System.out.println("\n=== Binary Search ===");

Arrays.sort(products, Comparator.comparing(p -> p.productName.toLowerCase()));

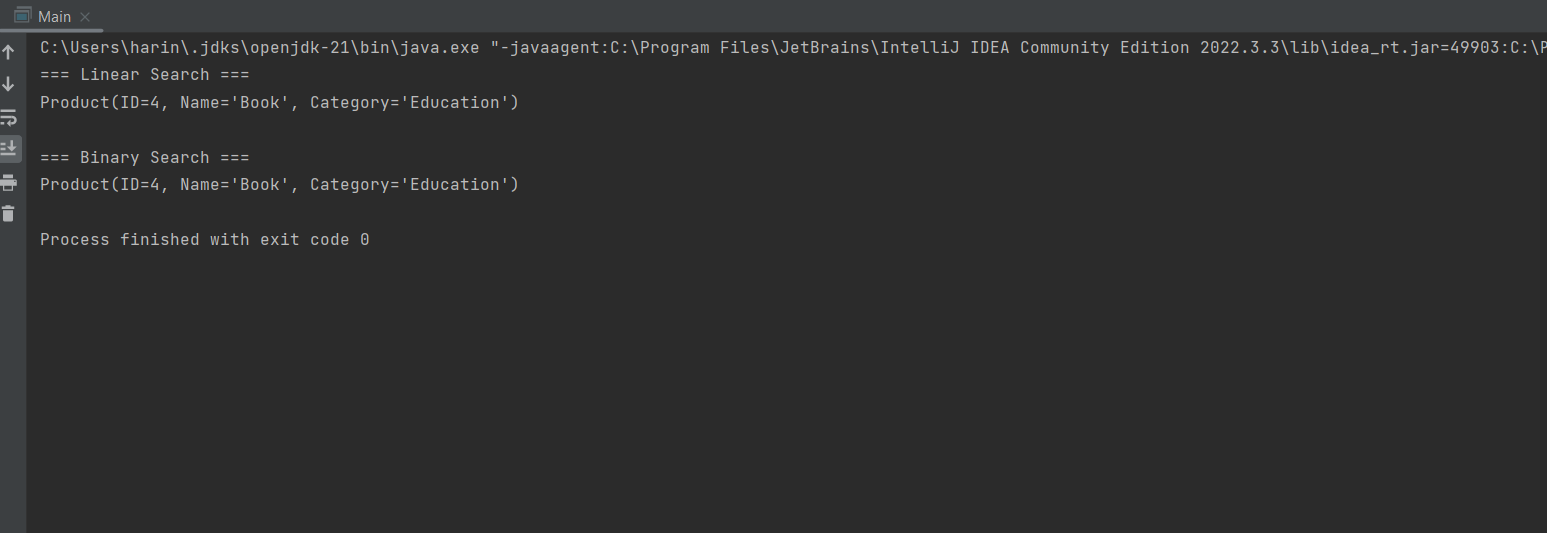
Product foundBinary = binarySearch(products, "Book");

System.out.println(foundBinary != null ? foundBinary : "Product not found");

}

}

Output:



**Exercise 7: Financial Forecasting**

**Scenario:**

You are developing a financial forecasting tool that predicts future values based on past data.

Code

public class FinancialForecast {

public static double forecastValue(double initialAmount, double growthRate, int years) {

if (years == 0) {

return initialAmount;

}

return forecastValue(initialAmount, growthRate, years - 1) \* (1 + growthRate);

}

public static void main(String[] args) {

double initialAmount = 10000;

double growthRate = 0.05;

int years = 5;

double futureValue = forecastValue(initialAmount, growthRate, years);

System.out.printf("Future Value after %d years: ₹%.2f%n", years, futureValue);

}

}

Output:

