**Data structures and Algorithms**

**Week 1**

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

**Code:**

**Product.java**

public 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 "[" + productId + "] " + productName + " - " + category;

    }

}

**Main.java**

import java.util.Arrays;

import java.util.Comparator;

public class Main {

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

        for (Product p : products) {

            if (p.productName.equalsIgnoreCase(target)) {

                return p;

            }

        }

        return null;

    }

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

        int left = 0;

        int right = products.length - 1;

        while (left <= right) {

            int mid = left + (right - left) / 2;

            int cmp = products[mid].productName.compareToIgnoreCase(target);

            if (cmp == 0) {

                return products[mid];

            } else if (cmp < 0) {

                left = mid + 1;

            } else {

                right = mid - 1;

            }

        }

        return null;

    }

    public static void main(String[] args) {

        Product[] products = {

            new Product(101, "Laptop", "Electronics"),

            new Product(102, "Shirt", "Clothing"),

            new Product(103, "Book", "Education"),

            new Product(104, "Smartphone", "Electronics"),

            new Product(105, "Shoes", "Footwear")

        };

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

        String target = "Book";

        Product foundLinear = linearSearch(products, target);

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

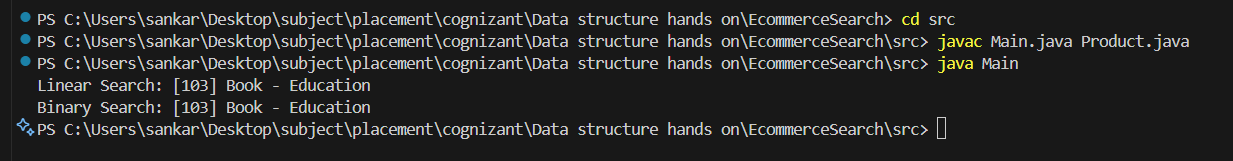
        Product foundBinary = binarySearch(products, target);

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

    }

}

**Output:**

****

**Exercise 7: Financial Forecasting**

**Code:**

**Forecast.java**

public class Forecast {

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

        if (years == 0) {

            return currentValue;

        }

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

    }

    public static void main(String[] args) {

        double currentValue = 10000; // ₹10,000

        double annualGrowthRate = 0.08; // 8% growth

        int forecastYears = 5;

        double futureValue = forecastValue(currentValue, annualGrowthRate, forecastYears);

        System.out.printf("Forecasted value after %d years: Rupees %.2f\n", forecastYears, futureValue);

    }

}

**Output:**

