**Filename:-Algorithms\_Data Structures**

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

**Code:-**

**Product.java**

public class Product {

    int productId;

    String productName;

    String category;

    public Product(int id, String name, String cat) {

        this.productId = id;

        this.productName = name;

        this.category = cat;

    }

    @Override

    public String toString() {

        return productId + " - " + productName + " (" + category + ")";

    }

}

**SearchDemo.java**

import java.util.Arrays;

public class SearchDemo {

    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, right = products.length - 1;

        while (left <= right) {

            int mid = (left + right) / 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(1, "Mouse", "Electronics"),

            new Product(2, "Keyboard", "Electronics"),

            new Product(3, "Chair", "Furniture"),

            new Product(4, "Lamp", "Lighting")

        };

        Arrays.sort(products, (a, b) -> a.productName.compareToIgnoreCase(b.productName));

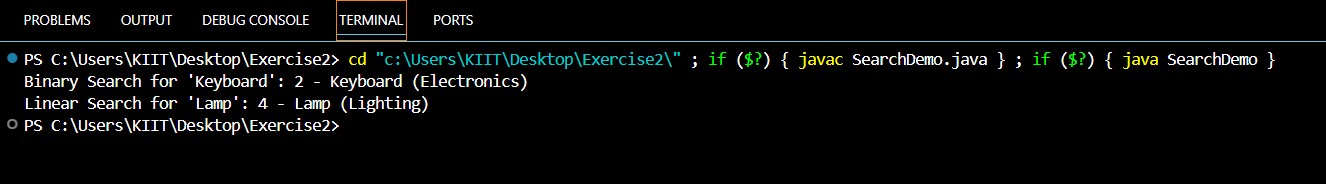
        System.out.println("Binary Search for 'Keyboard': " + binarySearch(products, "Keyboard"));

        System.out.println("Linear Search for 'Lamp': " + linearSearch(products, "Lamp"));

    }

}

**OUTPUT:-**

****

**Exercise 7: Financial Forecasting**

**Code:-**

**Forecast.java**

public class Forecast {

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

        if (years == 0) return currentValue;

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

    }

    public static void main(String[] args) {

        double initialValue = 10000;

        double growthRate = 0.08; // 8%

        int years = 5;

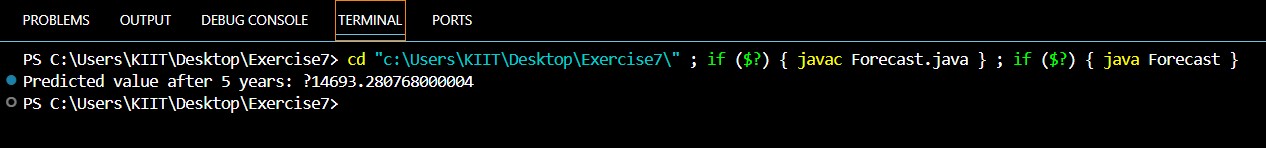
        double futureValue = predictValue(initialValue, growthRate, years);

        System.out.println("Predicted value after " + years + " years: ₹" + futureValue);

    }

}

**OUTPUT:-**

****