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

**Code:**

package DataStructuresAndAlgorithms.EcommerceSearch;

import java.util.\*;

class Product {

  String name;

  double price;

  Product(String name, double price) {

    this.name = name;

    this.price = price;

  }

}

class ProductSearch {

  private Map<String, Product> productMap = new HashMap<>();

  public void addProduct(Product product) {

    productMap.put(product.name.toLowerCase(), product);

  }

  public Product searchByName(String name) {

    return productMap.get(name.toLowerCase());

  }

}

class Main {

  public static void main(String[] args) {

    ProductSearch search = new ProductSearch();

    search.addProduct(new Product("Laptop", 60000));

    search.addProduct(new Product("Phone", 25000));

    Product result = search.searchByName("laptop");

    if (result != null) {

      System.out.println("Found: " + result.name + " for Rs " + result.price);

    } else {

      System.out.println("Product not found.");

    }

  }

}

**Output:**



**Exercise 7: Financial Forecasting**

**Code:**

package DataStructuresAndAlgorithms.FinancialForecasting;

public class FinancialForecast {

    public static double[] forecast(int[] sales, int daysToPredict) {

        int n = sales.length;

        double sumX = 0, sumY = 0, sumXY = 0, sumX2 = 0;

        for (int i = 0; i < n; i++) {

            sumX += i;

            sumY += sales[i];

            sumXY += i \* sales[i];

            sumX2 += i \* i;

        }

        double m = (n \* sumXY - sumX \* sumY) / (n \* sumX2 - sumX \* sumX);

        double c = (sumY - m \* sumX) / n;

        double[] predictions = new double[daysToPredict];

        for (int i = 0; i < daysToPredict; i++) {

            predictions[i] = m \* (n + i) + c;

        }

        return predictions;

    }

    public static void main(String[] args) {

        int[] sales = { 100, 120, 130, 90, 150, 160, 170, 200 };

        int daysToPredict = 5;

        double[] forecast = forecast(sales, daysToPredict);

        System.out.println("Linear Regression Forecast for next 5 days:");

        for (double f : forecast) {

            System.out.printf("%.2f ", f);

        }

    }

}

**Output:**

