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

**Product.java**

**public class Product {**

**int productId;**

**String productName;**

**String category;**

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

**this.productId = productId;**

**this.productName = productName;**

**this.category = category;**

**}**

**@Override**

**public String toString() {**

**return productId + " - " + productName + " - " + category;**

**}**

**}**

**Search.java**

**public class Search {**

**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, high = products.length - 1;**

**while (low <= high) {**

**int mid = (low + high) / 2;**

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

**if (cmp == 0){**

**return products[mid];**

**} else if (cmp < 0){**

**low = mid + 1;**

**} else {**

**high = mid - 1;**

**}**

**}**

**return null;**

**}**

**}**

**Main.java**

**import java.util.Arrays;**

**import java.util.Comparator;**

**public class Main {**

**public static void main(String[] args) {**

**System.*out*.println("----------OUTPUT---------------");**

**Product[] products = {**

**new Product(103, "Notebook", "Stationery"),**

**new Product(104, "Tablet", "Electronics"),**

**new Product(101, "Computer", "Electronics"),**

**new Product(102, "Shirt", "Fashion"),**

**new Product(105, "Pencil", "Stationery")**

**};**

**Product result1 = Search.*linearSearch*(products, "Tablet");**

**System.*out*.println("Linear Search Result: " + (result1 != null ? result1 : "Not Found"));**

**Arrays.*sort*(**

**products, Comparator.*comparing*(p -> p.productName.toLowerCase())**

**);**

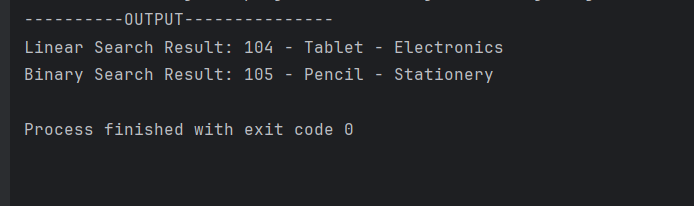
**Product result2 = Search.*binarySearch*(products, "Pencil");**

**System.*out*.println("Binary Search Result: " + (result2 != null ? result2 : "Not Found"));**

**}**

**}**

**Output**



**Exercise 7: Financial Forecasting**

**FinicialForcast.java**

**public class FinancialForecast {**

**public static double recursiveFunction(double presentValue, double rate, int years) {**

**if (years == 0) return presentValue;**

**return (1 + rate) \* recursiveFunction(presentValue, rate, years - 1);**

**}**

**}**

**Main.java**

**public class Main {**

**public static void main(String[] args) {**

**double presentValue = 20000.0;**

**double rate = 0.1;**

**int years = 6;**

**double futureValue = FinancialForecast.recursiveFunction(presentValue, rate, years);**

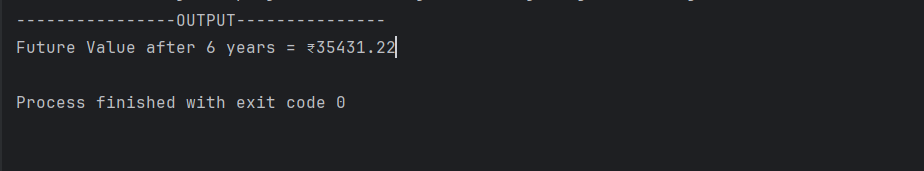
**System.*out*.println("----------------OUTPUT---------------");**

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

**}**

**}**

**Output**

****