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

Products.java

public class Products {  
 int productId;  
 String ProductName;  
 String category;  
  
 public Products(int id, String Name, String category){  
 this.Productid = id;  
 this.ProductName = Name;  
 this.category = category;  
 }  
  
}

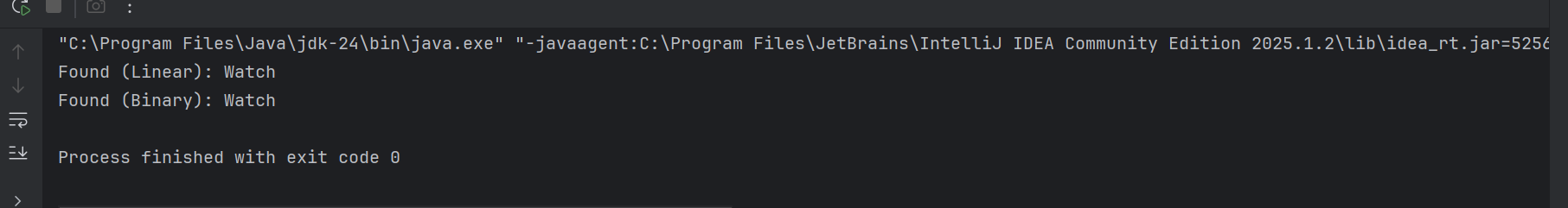
Search.java

public class Search {  
 public static Products LinearSearch(Products[] product, String Name){  
 for(Products p : product){  
 if (p.ProductName.equalsIgnoreCase(Name)) {  
 return p;  
 }  
 }  
 return null;  
 }  
 public static Products binarySearch(Products[] products, String name) {  
 int left = 0;  
 int right = products.length - 1;  
  
 while (left <= right) {  
 int mid = left + (right - left) / 2;  
 int cmp = products[mid].ProductName.compareToIgnoreCase(name);  
  
 if (cmp == 0)  
 return products[mid];  
 else if (cmp < 0)  
 left = mid + 1;  
 else  
 right = mid - 1;  
 }  
  
 return null;  
 }  
}

Main.java

import java.util.Arrays;  
import java.util.Comparator;  
  
public class Main {  
 public static void main(String[] args) {  
 Products[] products = {  
 new Products(1, "Laptop", "Electronics"),  
 new Products(2, "Shoes", "Fashion"),  
 new Products(3, "Watch", "Accessories"),  
 new Products(4, "Phone", "Electronics")  
 };  
  
  
 Products result1 = Search.*LinearSearch*(products, "Watch");  
 System.*out*.println(result1 != null ? "Found (Linear): " + result1.ProductName : "Not Found");  
  
  
 Arrays.*sort*(products, Comparator.*comparing*(p -> p.ProductName));  
  
  
 Products result2 = Search.*binarySearch*(products, "Watch");  
 System.*out*.println(result2 != null ? "Found (Binary): " + result2.ProductName : "Not Found");  
 }  
}

Output:



**Exercise 7: Financial Forecasting using Recursion**

FinancialForecast.java

public class FinancialForecast {  
  
 // Recursive method to calculate future value  
 public static double calculateFutureValue(double presentValue, double growthRate, int years) {  
 if (years == 0) {  
 return presentValue; // Base case  
 }  
  
 // Recursive case: apply one year's growth  
 return (1 + growthRate) \* *calculateFutureValue*(presentValue, growthRate, years - 1);  
 }  
}

Main.java

public class Main {  
 public static void main(String[] args) {  
 double presentValue = 10000; // Starting money ₹10,000  
 double growthRate = 0.1; // 10% annual growth  
 int years = 5;  
  
 double futureValue = FinancialForecast.*calculateFutureValue*(presentValue, growthRate, years);  
  
 System.*out*.printf(" Future Value after %d years: ₹%.2f\n", years, futureValue);  
 }  
}

output:

