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

**Main.java**

**Code:**

package Ecommerce;

import java.util.Arrays;

class Product {

    private int productId;

    private String productName;

    private String category;

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

        this.productId = productId;

        this.productName = productName;

        this.category = category;

    }

    public int getProductId() {

        return productId;

    }

    public String toString() {

        return "ID: " + productId + ", Name: " + productName + ", Category: " + category;

    }

}

public class EcommerceSearch {

*// Linear Search*

    public static Product linearSearch(Product[] products, int productId) {

        for (Product product : products) {

            if (product.getProductId() == productId) {

                return product;

            }

        }

        return null;

    }

*// Binary Search*

    public static Product binarySearch(Product[] products, int productId) {

        int left = 0, right = products.length - 1;

        while (left <= right) {

            int mid = (left + right) / 2;

            if (products[mid].getProductId() == productId) {

                return products[mid];

            } else if (products[mid].getProductId() < productId) {

                left = mid + 1;

            } else {

                right = mid - 1;

            }

        }

        return null;

    }

    public static void main(String[] args) {

        Product[] products = {

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

            new Product(105, "Headphones", "Electronics"),

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

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

            new Product(104, "Coffee Mug", "Kitchen")

        };

*// Sort for binary search (by productId)*

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

        int searchId = 103;

        Product resultLinear = linearSearch(products, searchId);

        Product resultBinary = binarySearch(products, searchId);

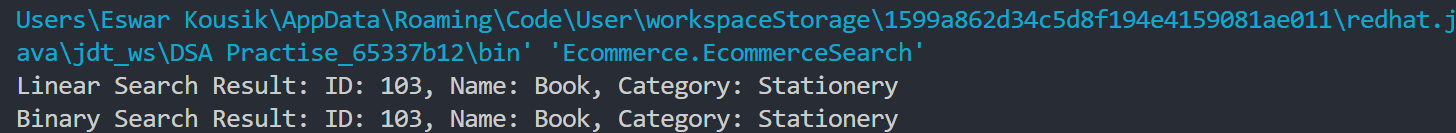
        System.out.println("Linear Search Result: " + (resultLinear != null ? resultLinear : "Product not found"));

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

    }

}

**Output:**



**Exercise 7: Financial Forecasting**

**Forecaster.java**

**Code:**

package Forecast;

public class Forecaster {

    public double futureValueRecursive(double presentValue, double rate, int years) {

        if (years == 0) {

            return presentValue;

        }

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

    }

}

**Main.java**

**Code:**

package Forecast;

public class Main {

    public static void main(String[] args) {

        Forecaster forecaster = new Forecaster();

        double presentValue = 1000.0;

        double rate = 0.05;

        int years = 10;

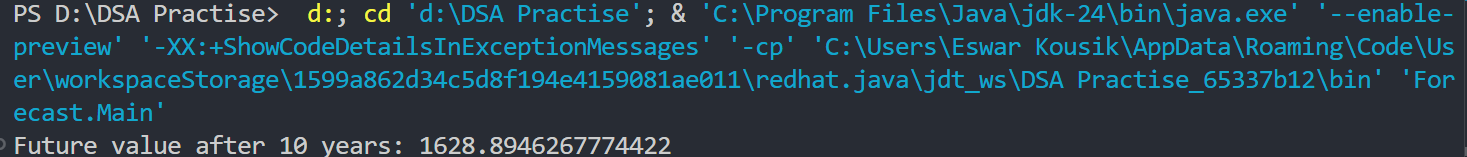
        double result = forecaster.futureValueRecursive(presentValue, rate, years);

        System.out.println("Future value after " + years + " years: " + result);

    }

}

**Output**

****