**Data Structures and Algorithms**

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

import java.util.Arrays;

import java.util.Comparator;

public class ECommerceSearch {

public static Product findByLinearSearch(Product[] items, String name) {

for (Product item : items) {

if (item.productName.equalsIgnoreCase(name)) {

return item;

}

}

return null;

}

public static Product findByBinarySearch(Product[] items, String name) {

int left = 0;

int right = items.length - 1;

while (left <= right) {

int mid = (left + right) / 2;

int check = items[mid].productName.compareToIgnoreCase(name);

if (check == 0) {

return items[mid];

} else if (check < 0) {

left = mid + 1;

} else {

right = mid - 1;

}

}

return null;

}

public static void main(String[] args) {

Product[] products = {

new Product(1, "Laptop", "Electronics"),

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

new Product(3, "Shoes", "Apparel"),

new Product(4, "Book", "Books"),

new Product(5, "Watch", "Accessories")

};

String keyword = "Shoes";

System.out.println("Linear Search Result:");

Product resultA = findByLinearSearch(products, keyword);

if (resultA != null) {

System.out.println("Found: " + resultA.productName);

} else {

System.out.println("Not Found");

}

Arrays.sort(products, Comparator.comparing(p -> p.productName.toLowerCase()));

System.out.println("\nBinary Search Result:");

Product resultB = findByBinarySearch(products, keyword);

if (resultB != null) {

System.out.println("Found: " + resultB.productName);

} else {

System.out.println("Not Found");

}

}

}

class Product {

int productId;

String productName;

String category;

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

this.productId = productId;

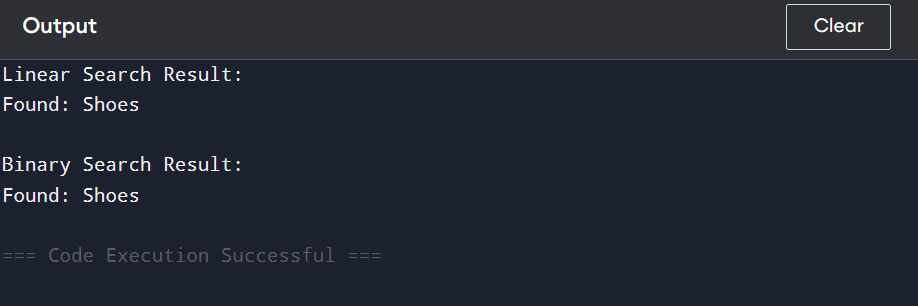
this.productName = productName;

this.category = category;

}

}

**Output:**

****

**Exercise 7: Financial Forecasting**

public class FinancialForecasting {

public static double forecastValue(double presentValue, double rate, int years) {

if (years == 0) {

return presentValue;

} else {

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

}

}

public static void main(String[] args) {

double presentValue = 1000.0;

double growthRate = 0.05;

int years = 5;

double futureValue = forecastValue(presentValue, growthRate, years);

System.out.printf("Future value after %d years: %.2f\n", years, futureValue);

}

}

**Output :**

