**Data Structures and Algorithms**

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

import java.util.Arrays;

import java.util.Comparator;

public class Main {

public static void main(String[] args) {

Product[] products = {

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

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

new Product(110, "Shoes", "Fashion"),

new Product(103, "Watch", "Fashion")

};

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

Product result1 = linearSearch(products, 110);

if (result1 != null) {

System.out.println("Found: " + result1.productName);

} else {

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

}

Arrays.sort(products, Comparator.comparingInt(p -> p.productId));

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

Product result2 = binarySearch(products, 110);

if (result2 != null) {

System.out.println("Found: " + result2.productName);

} else {

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

}

}

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

for (Product p : products) {

if (p.productId == targetId) {

return p;

}

}

return null;

}

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

int low = 0, high = products.length - 1;

while (low <= high) {

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

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

return products[mid];

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

low = mid + 1;

} else {

high = mid - 1;

}

}

return null;

}

}

class Product {

int productId;

String productName;

String category;

Product(int id, String name, String category) {

this.productId = id;

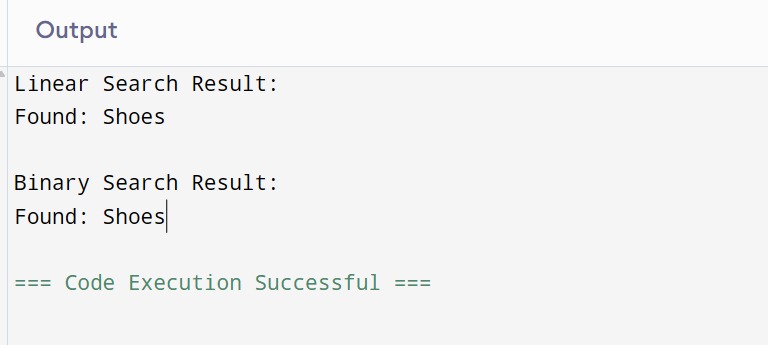
this.productName = name;

this.category = category;

}

}

**Output:**



**Exercise 7: Financial Forecasting**

public class FinancialForecast {

public static double predictFutureValue(double currentValue, double growthRate, int years) {

if (years == 0) {

return currentValue;

}

return predictFutureValue(currentValue \* (1 + growthRate), growthRate, years - 1);

}

public static void main(String[] args) {

double currentValue = 10000;

double growthRate = 0.08;

int years = 5;

double futureValue = predictFutureValue(currentValue, growthRate, years);

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

}

}

**Output :**

