Data Structures and Algorithms

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

**Program:**

public class SearchExample {

static class Product {

int productId;

String productName;

String category;

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

productId = id;

productName = name;

category = cat;

}

}

// Linear Search

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

for (int i = 0; i < products.length; i++) {

if (products[i].productName.equalsIgnoreCase(target)) {

return i;

}

}

return -1;

}

// Binary Search (requires sorted array)

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

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

while (low <= high) {

int mid = (low + high) / 2;

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

if (cmp == 0) return mid;

if (cmp < 0) low = mid + 1;

else high = mid - 1;

}

return -1;

}

public static void main(String[] args) {

Product[] products = {

new Product(1, "Book", "Education"),

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

new Product(3, "Laptop", "Electronics")

};

// Linear Search

int linearIndex = linearSearch(products, "Laptop");

if (linearIndex != -1)

System.out.println("Linear Search: Found at index " + linearIndex);

else

System.out.println("Linear Search: Not found");

// Sort for Binary Search

java.util.Arrays.sort(products, (a, b) -> a.productName.compareToIgnoreCase(b.productName));

// Binary Search

int binaryIndex = binarySearch(products, "Laptop");

if (binaryIndex != -1)

System.out.println("Binary Search: Found at index " + binaryIndex);

else

System.out.println("Binary Search: Not found");

}

}

**Output:**

****

**Exercise 7: Financial Forecasting**

**Program:**

public class FinancialForecast {

// Recursive method to calculate future value

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 current = 10000;

double rate = 0.05; // 5% growth per year

int years = 3;

double futureValue = *predictFutureValue*(current, rate, years);

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

}

}

**Output:**

