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

**Code:**

import java.util.Arrays;

import java.util.Comparator;

public class BinarySearch {

public static Product search(Product[] products, String targetName) {

// Sort the array by product name

Arrays.sort(products, Comparator.comparing(Product::getProductName));

int left = 0;

int right = products.length - 1;

while (left <= right) {

int mid = left + (right - left) / 2;

String midName = products[mid].getProductName();

int cmp = midName.compareToIgnoreCase(targetName);

if (cmp == 0) return products[mid];

else if (cmp < 0) left = mid + 1;

else right = mid - 1;

}return null;

}

}

public class LinearSearch {

public static Product search(Product[] products, String targetName) {

int steps = 0;

for (Product p : products) {

steps++;

if (p.getProductName().equalsIgnoreCase(targetName)) {

System.out.println("Linear Search: O(n), Steps: " + steps);

return p;

}

}

System.out.println("Linear Search: O(n), Steps: " + steps);

return null;

}

}

public 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 getProductName() {

return productName;

}

public String getCategory() {

return category;

}

@Override

public String toString() {

return "[" + productId + "] " + productName + " (" + category + ")";

} }

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Product[] products = {

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

new Product(102, "Shoes", "Fashion"),

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

new Product(104, "Phone", "Electronics"),

new Product(105, "Watch", "Accessories")

};

Scanner scanner = new Scanner(System.in);

System.out.print("Enter product name to search: ");

String searchQuery = scanner.nextLine().trim(); // Read user input

// Linear Search

Product result1 = LinearSearch.search(products, searchQuery);

System.out.println("Found (Linear): " + (result1 != null ? result1 : "Not found"));

// Binary Search

Product result2 = BinarySearch.search(products, searchQuery);

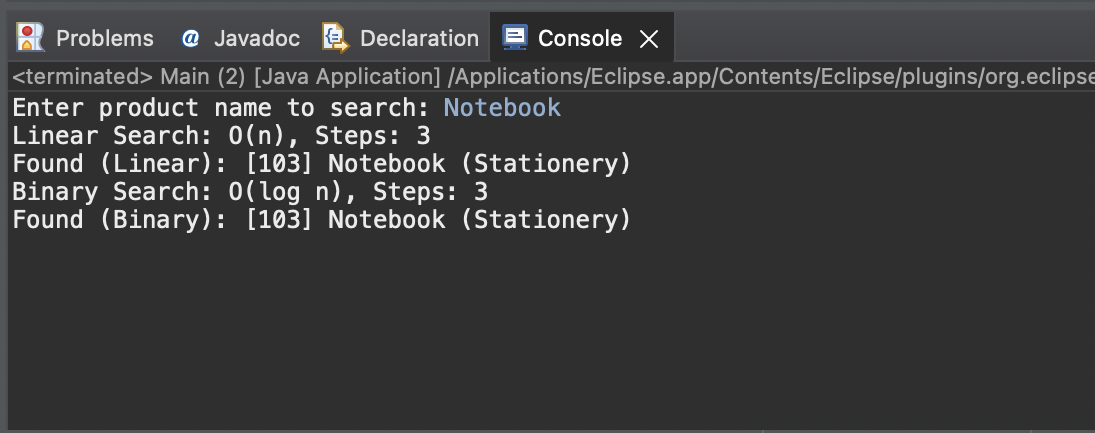
System.out.println("Found (Binary): " + (result2 != null ? result2 : "Not found"));

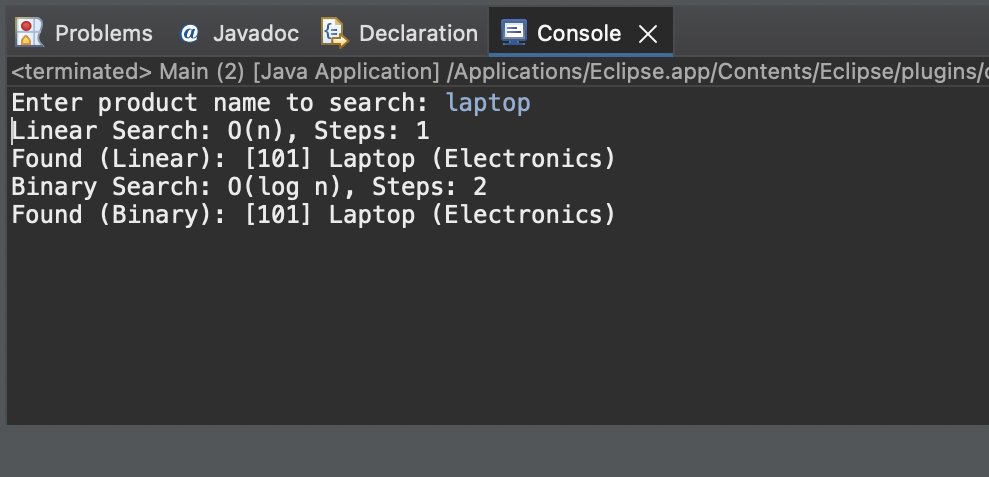
scanner.close();

}

}

**Output**:





**Exercise 7: Financial Forecasting**

**Code:**

public class ForecastCalculator {

public static double calculateFutureValue(double presentValue, double growthRate, int years) {

return presentValue \* Math.pow(1 + growthRate, years);

}

}

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter starting amount: ");

double presentValue = sc.nextDouble();

System.out.print("Enter annual growth rate (e.g. 0.1 for 10%): ");

double growthRate = sc.nextDouble();

System.out.print("Enter how many years to forecast: ");

int years = sc.nextInt();

System.out.println("Year-wise forecast:");

for (int i = 0; i <= years; i++) {

double futureValue = ForecastCalculator.calculateFutureValue(presentValue, growthRate, i);

System.out.printf("Year %d: ₹%.2f\n", i, futureValue);

}

sc.close();

}

}

**Output:**

