**2)Algorithms \_Data Structures**

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

1. Create a file **Product.java**

public 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;

}

@Override

public String toString() {

return productId + " - " + productName + " [" + category + "]";

}

}

1. Implement Linear and Binary Search

Create a file **SearchDemo.java:**

import java.util.Arrays;

import java.util.Comparator;

public class SearchDemo {

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

for (Product p : products) {

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

return p;

}

}

return null;

}

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

int left = 0;

int right = products.length - 1;

while (left <= right) {

int mid = (left + right) / 2;

int result = name.compareToIgnoreCase(products[mid].productName);

if (result == 0) {

return products[mid];

} else if (result < 0) {

right = mid - 1;

} else {

left = mid + 1;

}

}

return null;

}

public static void main(String[] args) {

Product[] products = {

new Product(101, "Shoes", "Footwear"),

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

new Product(103, "Laptop", "Electronics"),

new Product(104, "Watch", "Accessories"),

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

};

// Linear Search

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

Product found1 = linearSearch(products, "Laptop");

System.out.println(found1 != null ? found1 : "Product not found");

// Binary Search - first sort by name

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

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

Product found2 = binarySearch(products, "Laptop");

System.out.println(found2 != null ? found2 : "Product not found");

}

}

**Output:**

A screenshot of a computer program

AI-generated content may be incorrect.

**Exercise 7: Financial Forecasting**

1)Create a Java file named **FinancialForecast.java**:

public class FinancialForecast {

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

// Base case: 0 years left

if (years == 0) {

return currentValue;

}

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

}

public static void main(String[] args) {

double presentValue = 10000; // ₹10,000

double annualGrowthRate = 0.08; // 8% growth rate

int numberOfYears = 5;

double futureValue = predictFutureValue(presentValue, annualGrowthRate, numberOfYears);

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

}

}

**Output:**

A screenshot of a computer

AI-generated content may be incorrect.