**EXERCISE 1:**

***E-COMMERCE PLATFORM USING SEARCH FUNCTION***

import java.util.Arrays;

class Product {

int productId;

String productName;

String category;

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

this.productId = id;

this.productName = name;

this.category = cat;

}

void display() {

System.out.println(productId + " - " + productName + " (" + category + ")");

}

}

public class Main {

// Linear Search

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

for (Product p : products) {

if (p.productId == id) {

return p;

}

}

return null;

}

// Binary Search (Assumes sorted array by productId)

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

int left = 0, right = products.length - 1;

while (left <= right) {

int mid = (left + right) / 2;

if (products[mid].productId == id) return products[mid];

if (products[mid].productId < id) left = mid + 1;

else right = mid - 1;

}

return null;

}

public static void main(String[] args) {

Product[] products = {

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

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

new Product(104, "Book", "Stationery"),

new Product(102, "Phone", "Electronics")

};

// For Binary Search: sort by productId

Arrays.sort(products, (a, b) -> a.productId - b.productId);

// Test Linear Search

Product found1 = linearSearch(products, 104);

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

if (found1 != null) found1.display(); else System.out.println("Not Found");

// Test Binary Search

Product found2 = binarySearch(products, 102);

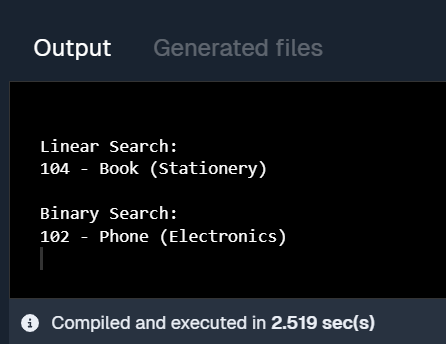
System.out.println("\nBinary Search:");

if (found2 != null) found2.display(); else System.out.println("Not Found");

}

}

**Using JDoodle compiler**

****

**EXERCISE 2:**

***FINANCIAL FORECASTING***

public class Main {

// Recursive method to calculate future value

public static double forecastFutureValue(double initialAmount, double growthRate, int years) {

// Base case: if no more years, return the initial amount

if (years == 0) return initialAmount;

// Recursive step: calculate value for previous year and apply growth

return forecastFutureValue(initialAmount, growthRate, years - 1) \* (1 + growthRate);

}

public static void main(String[] args) {

double initialAmount = 1000.0; // starting value in currency

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

int years = 5; // forecast for 5 years

double futureValue = forecastFutureValue(initialAmount, growthRate, years);

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

}

}

**Using JDoodle compiler**

****