# DATA STRUCTURES AND ALGORITHMS-HANDS ON EXERCISE:

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

**Scenario:**

You are working on the search functionality of an e-commerce platform. The search needs to be optimized for fast performance.

SearchService.java:

**import** java.util.Arrays;

**import** java.util.Comparator;

**public** **class** SearchService {

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

**for** (Product product : products) {

**if** (product.getProductId() == targetId) {

**return** product;

}

}

**return** **null**;

}

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

**int** left = 0;

**int** right = products.length - 1;

**while** (left <= right) {

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

**int** midId = products[mid].getProductId();

**if** (midId == targetId) {

**return** products[mid];

} **else** **if** (midId < targetId) {

left = mid + 1;

} **else** {

right = mid - 1;

}

}

**return** **null**;

}

**public** **static** **void** sortProductsById(Product[] products) {

Arrays.*sort*(products, Comparator.*comparingInt*(Product::getProductId));

}

}

SearchTest.java:

**public** **class** SearchTest {

**public** **static** **void** main(String[] args) {

Product[] products = {

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

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

**new** Product(101, "Book", "Stationery"),

**new** Product(103, "Phone", "Electronics"),

**new** Product(104, "Chair", "Furniture")

};

**int** targetId = 103;

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

Product foundLinear = SearchService.*linearSearch*(products, targetId);

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

System.***out***.println("\n📑 Binary Search (After Sorting):");

SearchService.*sortProductsById*(products);

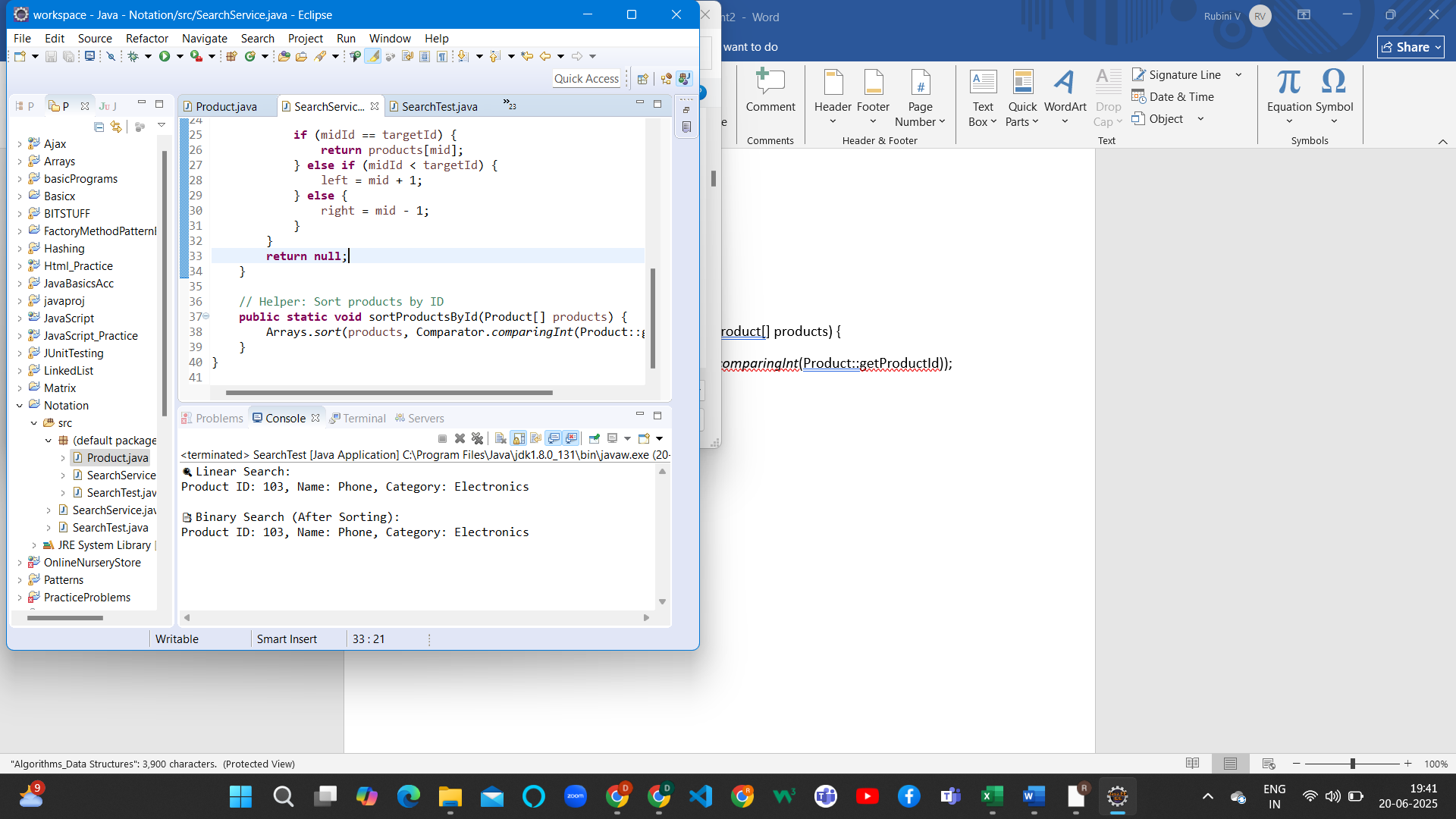
Product foundBinary = SearchService.*binarySearch*(products, targetId);

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

}

}

OUTPUT:



2. **Exercise 7: Financial Forecasting**

**Scenario:**

You are developing a financial forecasting tool that predicts future values based on past data.

**public** **class** FinancialForecast {

**public** **static** **double** futureValue(**double** initialAmount, **double** growthRate, **int** years) {

**if** (years == 0) {

**return** initialAmount;

}

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

}

**public** **static** **void** main(String[] args) {

**double** initialAmount = 1000.0; // ₹1000

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

**int** years = 5;

**double** predictedValue = *futureValue*(initialAmount, growthRate, years);

System.***out***.printf("📈 Predicted value after %d years: ₹%.2f%n", years, predictedValue);

}

}

OUTPUT:

