**WEEK-1 (6397712 BANDARU SOWMYA)**

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

**MAIN.JAVA:**

import java.util.Arrays;

public class Main {

public static void main(String[] args) {

// Create sample products

Product[] products = {

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

new Product(1, "Smartphone", "Electronics"),

new Product(4, "Headphones", "Accessories"),

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

};

// Create sorted array for binary search

Product[] sortedProducts = products.clone();

Arrays.sort(sortedProducts);

// Test linear search

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

int targetId = 3;

Product result = SearchAlgorithms.linearSearch(products, targetId);

System.out.println("Searching for product ID " + targetId + ": " + (result != null ? result : "Not found"));

// Test binary search

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

result = SearchAlgorithms.binarySearch(sortedProducts, targetId);

System.out.println("Searching for product ID " + targetId + ": " + (result != null ? result : "Not found"));

// Test case for product not found

targetId = 5;

System.out.println("\nLinear Search for ID " + targetId + ": " +

(SearchAlgorithms.linearSearch(products, targetId) != null ? "Found" : "Not found"));

System.out.println("Binary Search for ID " + targetId + ": " +

(SearchAlgorithms.binarySearch(sortedProducts, targetId) != null ? "Found" : "Not found"));

}

}

**PRODUCT.JAVA:**

public class Product implements Comparable<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 int compareTo(Product other) {

return Integer.compare(this.productId, other.productId);

}

@Override

public String toString() {

return "Product [ID=" + productId + ", Name=" + productName + ", Category=" + category + "]";

}

}

**SEARCHALGORITHMS:**

public class SearchAlgorithms {

// Linear Search

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

for (Product product : products) {

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

return product;

}

}

return null;

}

// Binary Search

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

int left = 0;

int right = sortedProducts.length - 1;

while (left <= right) {

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

int currentId = sortedProducts[mid].getProductId();

if (currentId == targetId) {

return sortedProducts[mid];

} else if (currentId < targetId) {

left = mid + 1;

} else {

right = mid - 1;

}

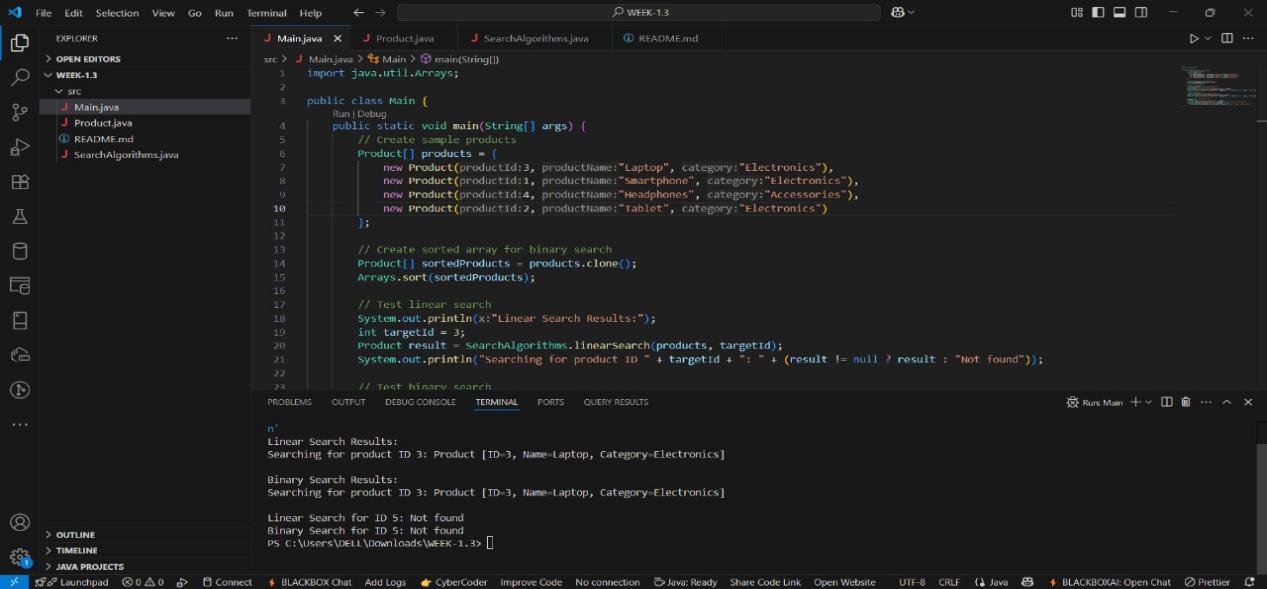
}

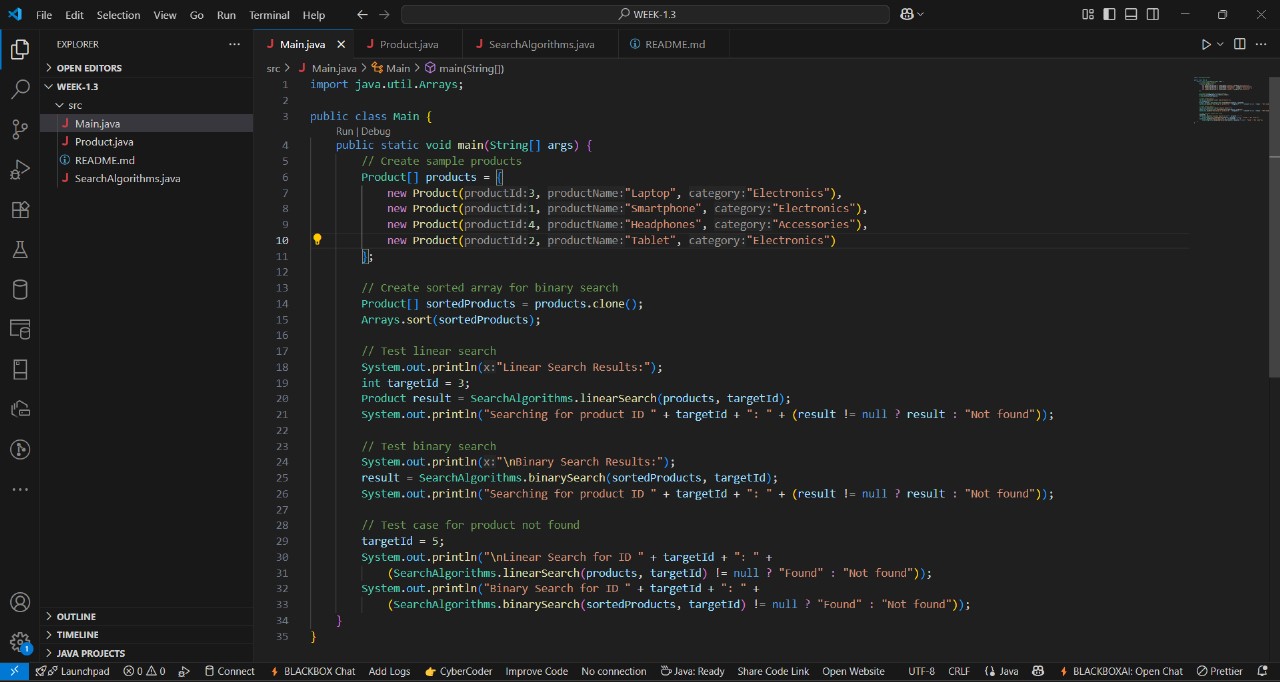
return null;

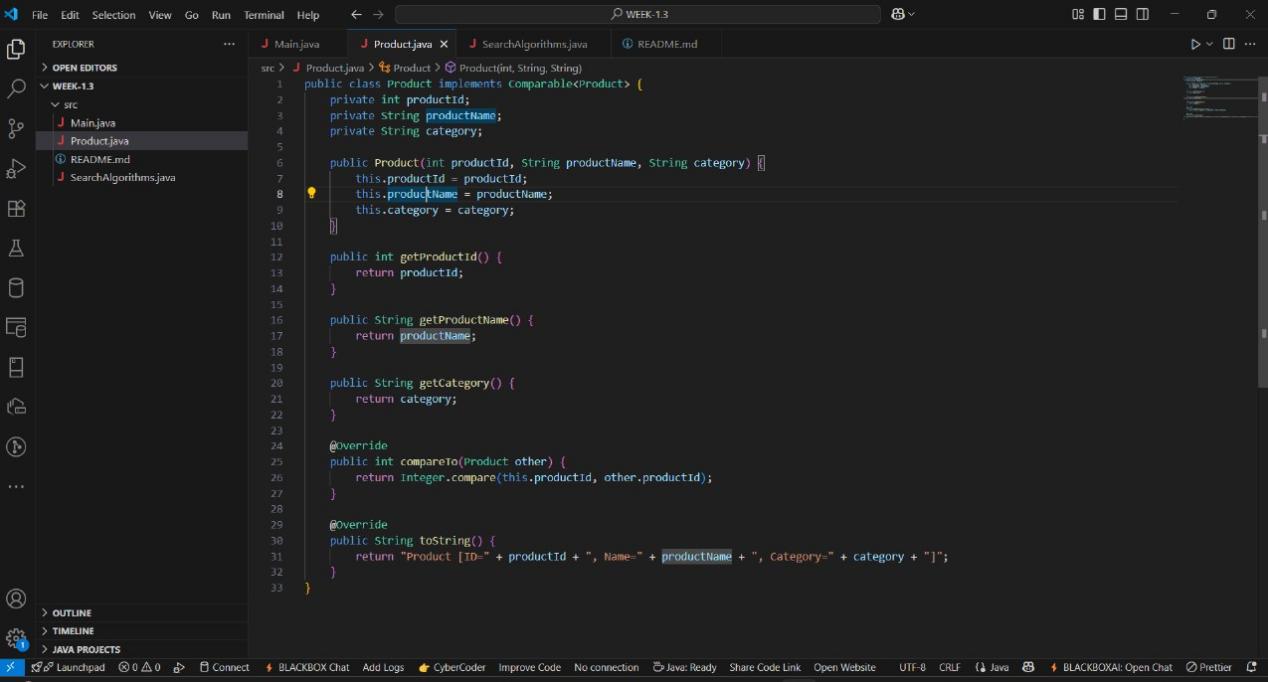
}

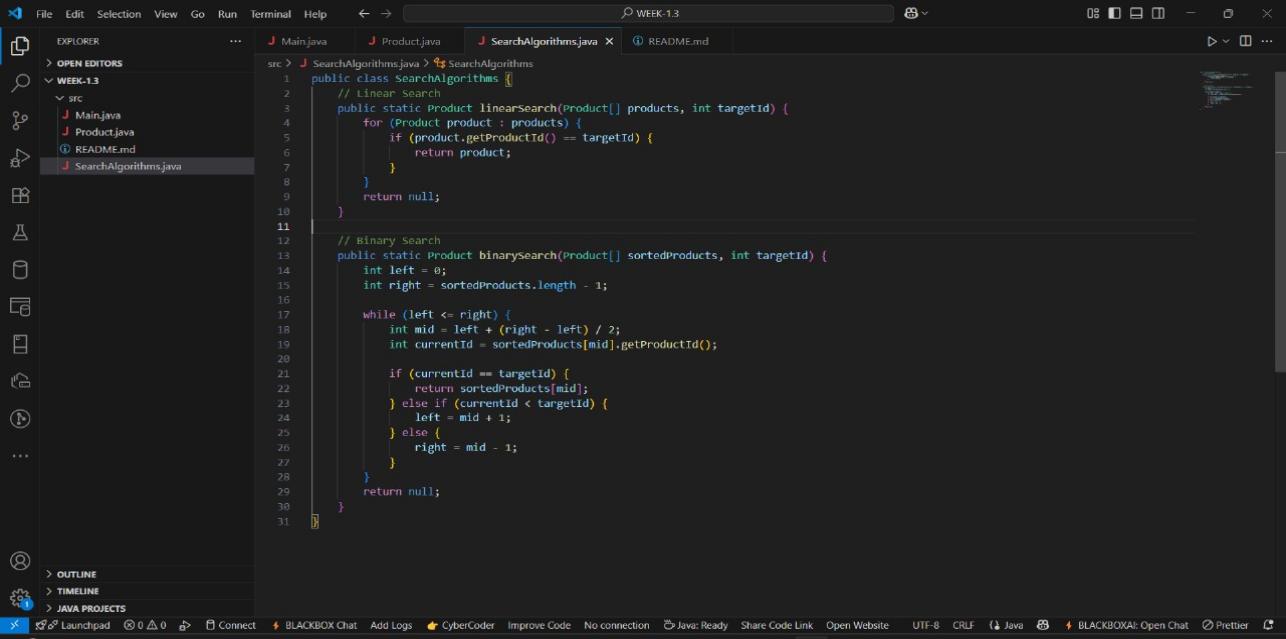
}

**OUTPUT:**

****

****

****

****