**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.

**Steps:**

1. **Understand Asymptotic Notation:**
   * Explain Big O notation and how it helps in analyzing algorithms.
   * Describe the best, average, and worst-case scenarios for search operations.
2. **Setup:**
   * Create a class **Product** with attributes for searching, such as **productId, productName**, and **category**.
3. **Implementation:**
   * Implement linear search and binary search algorithms.
   * Store products in an array for linear search and a sorted array for binary search.
4. **Analysis:**
   * Compare the time complexity of linear and binary search algorithms.
   * Discuss which algorithm is more suitable for your platform and why.

**CODE:**

using System;

class Product

{

public int ProductId { get; set; }

public string ProductName { get; set; }

public string Category { get; set; }

public Product(int id, string name, string category)

{

ProductId = id;

ProductName = name;

Category = category;

}

}

class Search

{

//Linear Search

public static int LinearSearch(Product[] products, string target)

{

for (int i = 0; i < products.Length; i++)

if (products[i].ProductName == target)

return i;

return -1;

}

//Binary Search

public static int BinarySearch(Product[] products, string target)

{

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

while (left <= right)

{

int mid = (left + right) / 2;

int result = string.Compare(products[mid].ProductName, target);

if (result == 0) return mid;

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

else right = mid - 1;

}

return -1;

}

}

class Program2

{

static void Main()

{

Product[] products = {

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

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

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

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

};

Array.Sort(products, (a, b) => a.ProductName.CompareTo(b.ProductName)); // For binary search

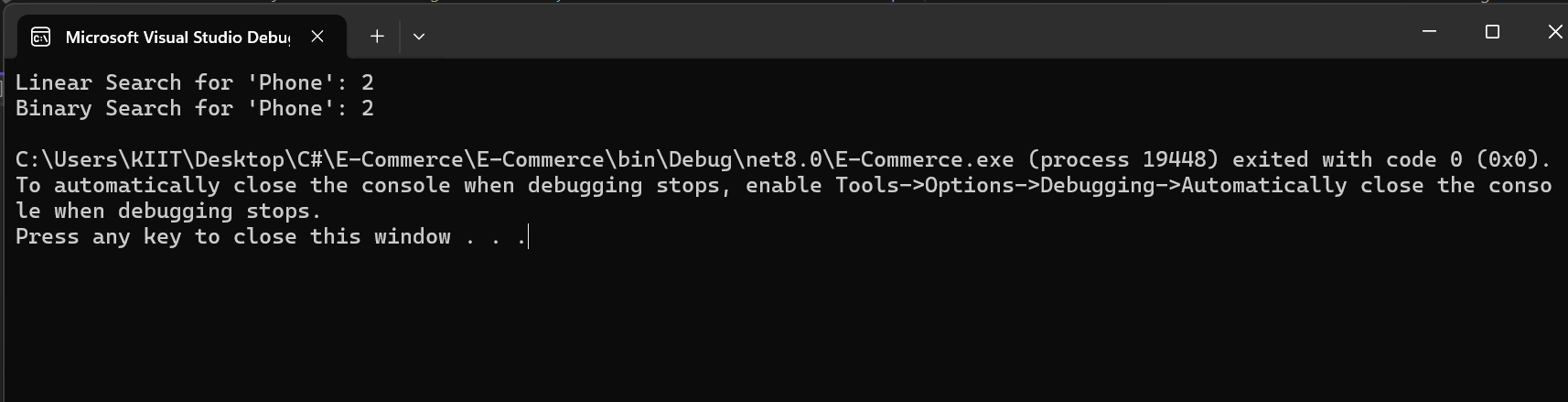
Console.WriteLine("Linear Search for 'Phone': " + Search.LinearSearch(products, "Phone"));

Console.WriteLine("Binary Search for 'Phone': " + Search.BinarySearch(products, "Phone"));

}

}

**OUTPUT:**



**ANALYSIS:**

****Time Complexity:****

* **Linear Search**: O(n)
* **Binary Search**: O(log n) (only if sorted)

**Best Algorithm:**  
Binary Search is better for large sorted datasets. Linear is simpler for small/unsorted lists.