**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

**Understanding Asymptotic Notation:**

**Big O Notation**

Big O notation is used to describe the performance or complexity of an algorithm. It specifically describes the worst-case scenario and provides an upper bound on the time required as the input size grows.

* **O(1)**: Constant time - The operation doesn't depend on the size of the input.
* **O(log n)**: Logarithmic time - The operation depends on the logarithm of the input size.
* **O(n)**: Linear time - The operation depends linearly on the input size.
* **O(n log n)**: Linearithmic time - The operation depends on the input size times its logarithm.
* **O(n^2)**: Quadratic time - The operation depends on the square of the input size.

**Best, Average, and Worst-case Scenarios for Search Operations**

* **Best Case**: The scenario where the search operation completes the fastest.
* **Average Case**: The scenario where the search operation completes in average time.
* **Worst Case**: The scenario where the search operation takes the longest time.

**Setup:**

I have used Visual Studio to create a new project with the name E-commerce Platform Search Function

**Implementation:**

I used both Linear Search and Binary Search algorithms. Output obtained after using the both methods is of course same.

**Analysis:**

**Time Complexity**

* **Linear Search:**
  + Best Case: O(1) - The product is the first element in the array.
  + Average Case: O(n) - The product is somewhere in the middle.
  + Worst Case: O(n) - The product is the last element or not present in the array.
* **Binary Search:**
  + Best Case: O(1) - The product is the middle element.
  + Average Case: O(log n) - The product is found in one of the divisions.
  + Worst Case: O(log n) - The product is not present and the algorithm searches through all divisions.

**Which Algorithm is More Suitable?**

* Linear Search: Suitable for small datasets where the overhead of sorting is not justified.
* Binary Search: More suitable for large datasets due to its logarithmic time complexity, but requires the dataset to be sorted first**.**

.