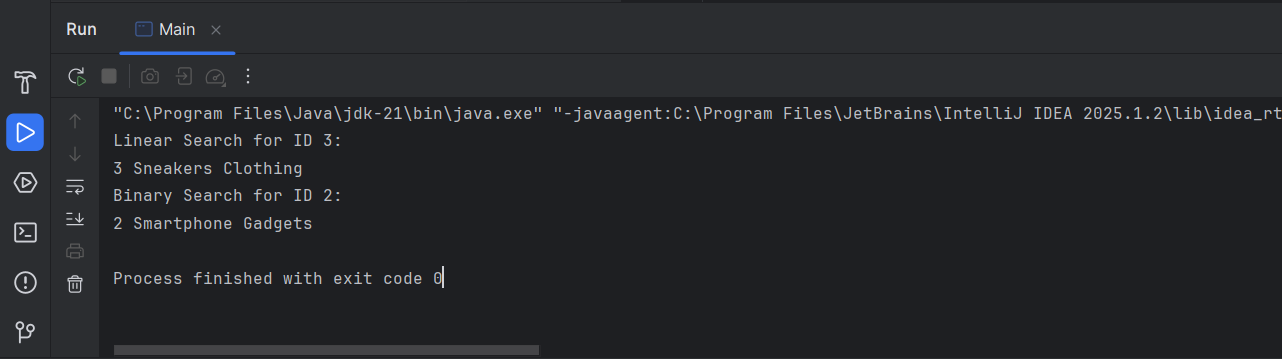
**Exercise 2: E-commerce Platform Search Function**

**Scenario:**

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



**1. UNDERSTANDING ASYMPTOTIC NOTATION**

**Big O Notation:**

Big O notation is used to express the time or space complexity of an algorithm in relation to the size of the input, denoted as n. It helps us estimate how the algorithm behaves as the input grows. Instead of focusing on exact execution time, Big O describes the *growth rate*, allowing us to analyze and compare algorithms in terms of efficiency and scalability.

For example, an algorithm with a complexity of O(n) will take twice as long if the input doubles, whereas one with O(log n) will grow much slower, making it more efficient for large datasets.

**Best, Average, and Worst Cases in Search:**

* **Best Case:** The target element is located in the first comparison. For linear search, this means it's the first item; for binary search, it could be found in the first middle split. This is the most efficient outcome.
* **Average Case:** The element is assumed to be in a random position, so the number of comparisons reflects an average across many runs. It's useful for understanding general performance.
* **Worst Case:** The element is either not in the list or at the last possible position. In linear search, this means checking every element. In binary search, it's the maximum number of divisions needed. This case reflects the maximum possible work the algorithm may need to do.

**2. SETUP**

A Product class is defined with key attributes that are relevant for search operations: productId, productName, and category. These allow for both numeric and string-based lookups in the search algorithms.

**3. IMPLEMENTATION**

Both **Linear Search** and **Binary Search** algorithms are implemented:

* **Linear Search** goes through the product list one item at a time. It works on unsorted arrays and is simple to implement.
* **Binary Search** requires the product list to be sorted by productId (or other searchable attributes). It repeatedly splits the array into halves to narrow down the target efficiently.

Products are stored in an array for linear search and a sorted array for binary search to match the algorithm's requirements.

**4. ANALYSIS**

**Time Complexity Comparison:**

* **Linear Search:** O(n) – It checks each item until the target is found or the list ends. The time increases linearly with the number of products.
* **Binary Search:** O(log n) – It divides the search range by half each time. This logarithmic growth is much faster for large inputs.

**Choosing the Right Algorithm:**

For an e-commerce platform, where fast and accurate search is essential, **Binary Search** is more appropriate—*but only when the product data is sorted*. It ensures quick results even with thousands of products. Linear search is useful for smaller or unsorted datasets but becomes inefficient as the dataset grows.

Therefore, **binary search is more suitable** for optimizing the search functionality of an e-commerce system where performance matters and the dataset is large and can be kept sorted.