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

import java.util.\*;

public class EcommerceSearch {

public static int binarySearch(String[] products, String target) {

int low = 0, high = products.length - 1;

while (low <= high) {

int mid = low + (high-low)/2;

int found = products[mid].compareToIgnoreCase(target);

if (found == 0) return mid;

else if (found < 0) low = mid + 1;

else high = mid - 1;

}

return -1;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter number of items: ");

int n = sc.nextInt();

String products[] = new String[n];

Map<String,Integer> map = new HashMap<>();

for(int i=0;i<n;i++){

System.out.print("Enter name of item: ");

products[i] = sc.next();

map.put(products[i],i); //storing original index

}

Arrays.sort(products); // Ensure sorted for binary search

System.out.print("Enter product name to search: ");

String target = sc.next();

int index = binarySearch(products, target);

if (index != -1)

System.out.println(target + " found at index " + map.get(target) + ".");

else

System.out.println(target + " not found in the store.");

}

}

**Why we chose binary search logic:**

* We are searching for a product name in a **sorted list** of product strings.
* **Binary Search** is optimal for sorted data:  
  Time Complexity = **O(log n)**  
  Efficiently halves the search space each time.
* We used compareToIgnoreCase() to handle **case-insensitive lexicographic comparison**.

**This makes the search fast, efficient, and accurate** for real-world e-commerce use cases.

**OUTPUT**

