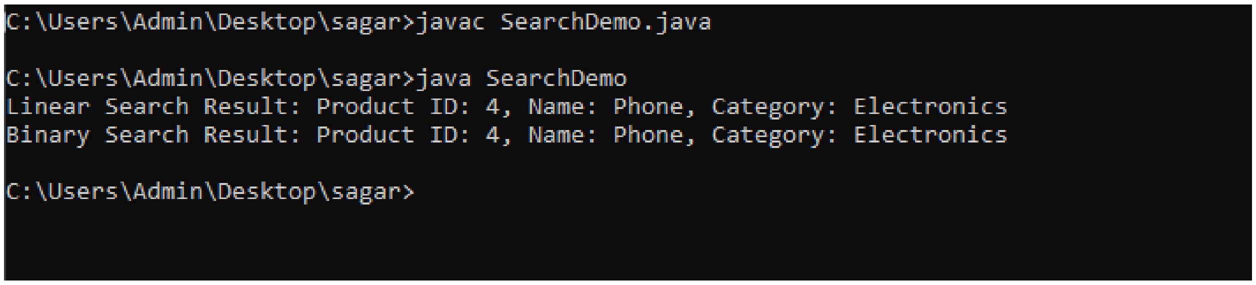
## Exercise 1: E-commerce Platform Search Function

```
import java.util.Arrays;

import java.util.Comparator;


public class SearchDemo {

    public static void main(String[] args) {

        Product[] products = {

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

            new Product(2, "Shirt", "Clothing"),

            new Product(3, "Shoes", "Footwear"),

            new Product(4, "Phone", "Electronics"),

            new Product(5, "Watch", "Accessories")

        };


        // Linear Search

        Product result1 = linearSearch(products, "Phone");

        System.out.println("Linear Search Result: " + result1);


        // Sorting for Binary Search

        Arrays.sort(products, Comparator.comparing(p -> p.productName));


        // Binary Search

        Product result2 = binarySearch(products, "Phone");

        System.out.println("Binary Search Result: " + result2);

    }


    public static Product linearSearch(Product[] products, String name) {

        for (Product p : products) {

            if (p.productName.equalsIgnoreCase(name)) {
```

```
            return p;
        }
    }
    return null;
}

public static Product binarySearch(Product[] products, String name) {
    int low = 0, high = products.length - 1;

    while (low <= high) {
        int mid = (low + high) / 2;
        int cmp = products[mid].productName.compareToIgnoreCase(name);

        if (cmp == 0) return products[mid];
        else if (cmp < 0) low = mid + 1;
        else high = mid - 1;
    }
    return null;
}
static class Product {
    int productId;
    String productName;
    String category;

    public Product(int productId, String productName, String category) {
        this.productId = productId;
        this.productName = productName;
        this.category = category;
    }
```

```
        @Override
        public String toString() {
            return "Product ID: " + productId + ", Name: " + productName + ", Category: " + category;
        }
    }
}
```

**Output**

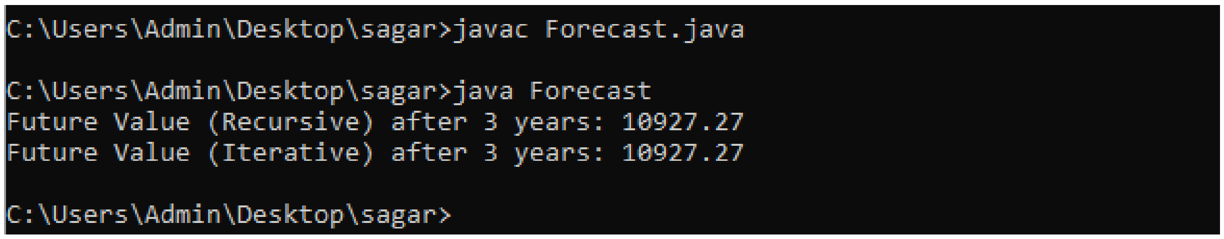


```
C:\Users\Admin\Desktop\sagar>javac SearchDemo.java

C:\Users\Admin\Desktop\sagar>java SearchDemo
Linear Search Result: Product ID: 4, Name: Phone, Category: Electronics
Binary Search Result: Product ID: 4, Name: Phone, Category: Electronics

C:\Users\Admin\Desktop\sagar>
```

## Exercise 2: Financial Forecasting

```
public class Forecast {
    public static double futureValueRecursive(double presentValue, double rate, int years) {
        if (years == 0) {
            return presentValue;
        }
        return (1 + rate) * futureValueRecursive(presentValue, rate, years - 1);
    }
    public static double futureValueIterative(double presentValue, double rate, int years) {
        double result = presentValue;
        for (int i = 0; i < years; i++) {
            result *= (1 + rate);
        }
        return result;
    }
    public static void main(String[] args) {
        double presentValue = 10000.0;

        double annualGrowthRate = 0.03;

        int years = 3;

        double futureValueRec = futureValueRecursive(presentValue, annualGrowthRate, years);

        System.out.printf("Future Value (Recursive) after %d years: %.2f\n", years, futureValueRec);

        double futureValueItr = futureValueIterative(presentValue, annualGrowthRate, years);

        System.out.printf("Future Value (Iterative) after %d years: %.2f\n", years, futureValueItr);

    }

}
```

**Output**



```
C:\Users\Admin\Desktop\sagar>javac Forecast.java

C:\Users\Admin\Desktop\sagar>java Forecast
Future Value (Recursive) after 3 years: 10927.27
Future Value (Iterative) after 3 years: 10927.27

C:\Users\Admin\Desktop\sagar>
```