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

**Code:**

import java.util.Arrays;

import java.util.Scanner;

public class EcommerceSearchCLI {

    static class Product implements Comparable<Product> {

        int productId;

        String productName;

        String category;

        Product(int productId, String productName, String category) {

            this.productId = productId;

            this.productName = productName;

            this.category = category;

        }

        public String toString() {

            return "[" + productId + "] " + productName + " - " + category;

        }

        public int compareTo(Product other) {

            return this.productName.compareToIgnoreCase(other.productName);

        }

    }

    static Product[] productList = {

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

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

        new Product(3, "T-shirt", "Clothing"),

        new Product(4, "Shoes", "Footwear"),

        new Product(5, "Book", "Education"),

        new Product(6, "Watch", "Accessories")

    };

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.println("=== E-commerce Product Search (Terminal Version) ===");

        while (true) {

            System.out.print("\nEnter product name to search (or 'exit' to quit): ");

            String query = scanner.nextLine().trim().toLowerCase();

            if (query.equalsIgnoreCase("exit")) {

                System.out.println("Exiting program.");

                break;

            }

            System.out.println("Choose search type: 1. Linear Search  2. Binary Search");

            System.out.print("Enter choice (1/2): ");

            String choice = scanner.nextLine().trim();

            if (choice.equals("1")) {

                performLinearSearch(query);

            } else if (choice.equals("2")) {

                performBinarySearch(query);

            } else {

                System.out.println("Invalid choice. Please enter 1 or 2.");

            }

        }  scanner.close();

    }

    private static void performLinearSearch(String query) {

        System.out.println("Linear Search Results:");

        boolean found = false;

        for (Product p : productList) {

            if (p.productName.toLowerCase().contains(query)) {

                System.out.println(p);

                found = true;

            }

        }

        if (!found)

            System.out.println("No matching products found.");

    }

    private static void performBinarySearch(String query) {

        Product[] sortedList = Arrays.copyOf(productList, productList.length);

        Arrays.sort(sortedList);

        int index = binarySearch(sortedList, query);

        if (index != -1) {

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

            System.out.println(sortedList[index]);

        } else {

            System.out.println("Product not found.");

        }

    }

    private static int binarySearch(Product[] products, String targetName) {

        int low = 0;

        int high = products.length - 1;

        while (low <= high) {

            int mid = (low + high) / 2;

            String midName = products[mid].productName.toLowerCase();

            int cmp = midName.compareTo(targetName);

            if (cmp == 0) {

                return mid;

            } else if (cmp < 0) {

                low = mid + 1;

            } else {

                high = mid - 1;

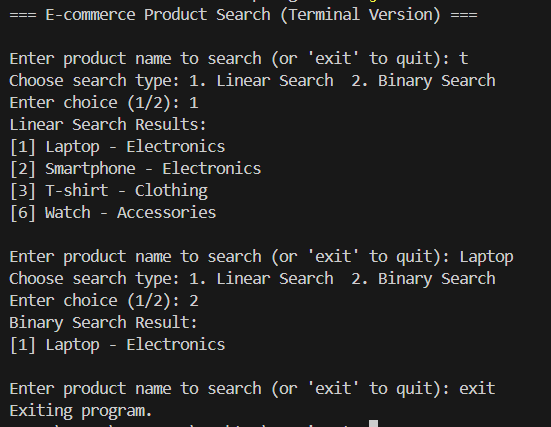
            }

        } return -1;

    }

}

**Output:**



**Exercise 7: Financial Forecasting**

**Code:**

import java.util.Scanner;

public class FinancialForecast {

    public static double futureValue(double initialInvestment, double growthRate, int years) {

        if (years == 0) {

            return initialInvestment;

        }

        return futureValue(initialInvestment, growthRate, years - 1) \* (1 + growthRate);

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the initial investment: ");

        double initialInvestment = sc.nextDouble();

        System.out.print("Enter the annual growth rate (in %): ");

        double annualGrowthRate = sc.nextDouble();

        annualGrowthRate /= 100;

        System.out.print("Enter the years: ");

        int years = sc.nextInt();

        double result = futureValue(initialInvestment, annualGrowthRate, years);

        System.out.printf("Future value after %d years: Rs. %.2f\n", years, result);

        sc.close(); //

    }

}

**Output**:

A black background with white text

AI-generated content may be incorrect.