**Data Structures & Algorithms**

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

**Code:**

import java.util.\*;

public class ECommerceSearch {

public static List<String> searchProducts(List<String> products, String keyword) {

List<String> matched = new ArrayList<>();

String lowerKeyword = keyword.toLowerCase();

for (String product : products) {

if (product.toLowerCase().contains(lowerKeyword)) {

matched.add(product);

}

}

if (matched.isEmpty()) {

matched.add("No products found");

}

return matched;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

List<String> products = Arrays.asList(

"reduce","fire","result","watch","output","hello","product","produce"

);

System.out.println("Welcome to the E-Commerce Search Platform");

System.out.print("search:");

String keyword = scanner.nextLine();

List<String> results = searchProducts(products, keyword);

System.out.println("\nSearch Results:");

for (String item : results) {

System.out.println("- " + item);

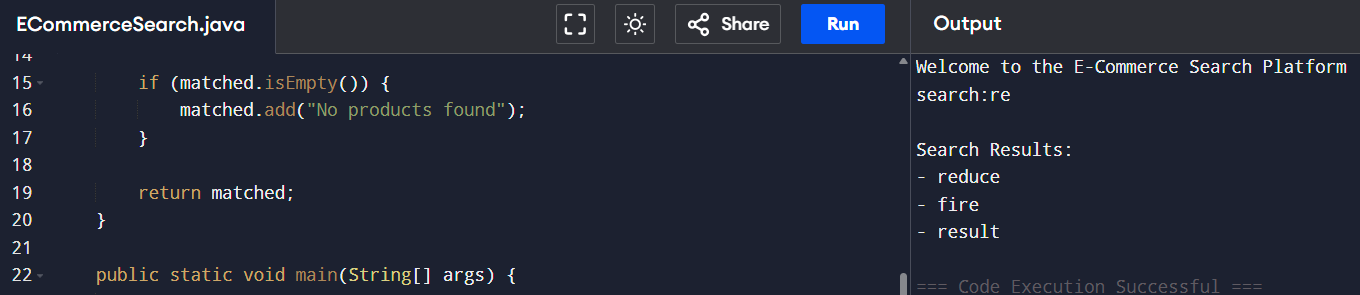
}

scanner.close();

}

}

**Output:**

****

**Exercise 7: Exercise 7: Financial Forecasting**

**Code:**

public class Main {

double calculateCompoundInterest(double principal, double rate, int years) {

return principal \* Math.pow(1 + rate / 100.0, years);

}

void run() {

double principal = 1000;

double rate = 2;

int years = 4;

double result = calculateCompoundInterest(principal, rate, years);

System.out.println("No of years: " + years);

System.out.printf("Results: Rs. %.2f\n", result);

}

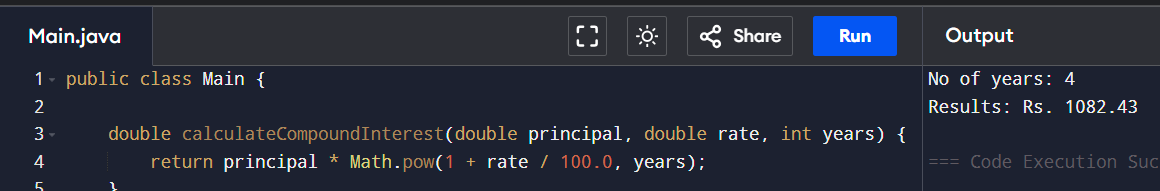
public static void main(String[] args) {

new Main().run();

}

}

**Output:**

****