**[Digital-Nurture-4.0-JavaFSE](https://github.com/seshadrimr/Digital-Nurture-4.0-JavaFSE/tree/main) Week 1 Exercise:**

Algorithms\_Data Structures:

**1.** **E-commerce Platform Search Function**

**Code:**

import java.util.\*;

public class ProductSearch {

public static void main(String[] args) {

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

catalog.put(101, "red dress");

catalog.put(102, "blue jeans");

catalog.put(103, "red shirt");

String keyword = "red";

System.out.println("Search results for: " + keyword);

for (Map.Entry<Integer, String> entry : catalog.entrySet()) {

if (entry.getValue().toLowerCase().contains(keyword.toLowerCase())) {

System.out.println("Product ID: " + entry.getKey() + ", Name: " + entry.getValue());

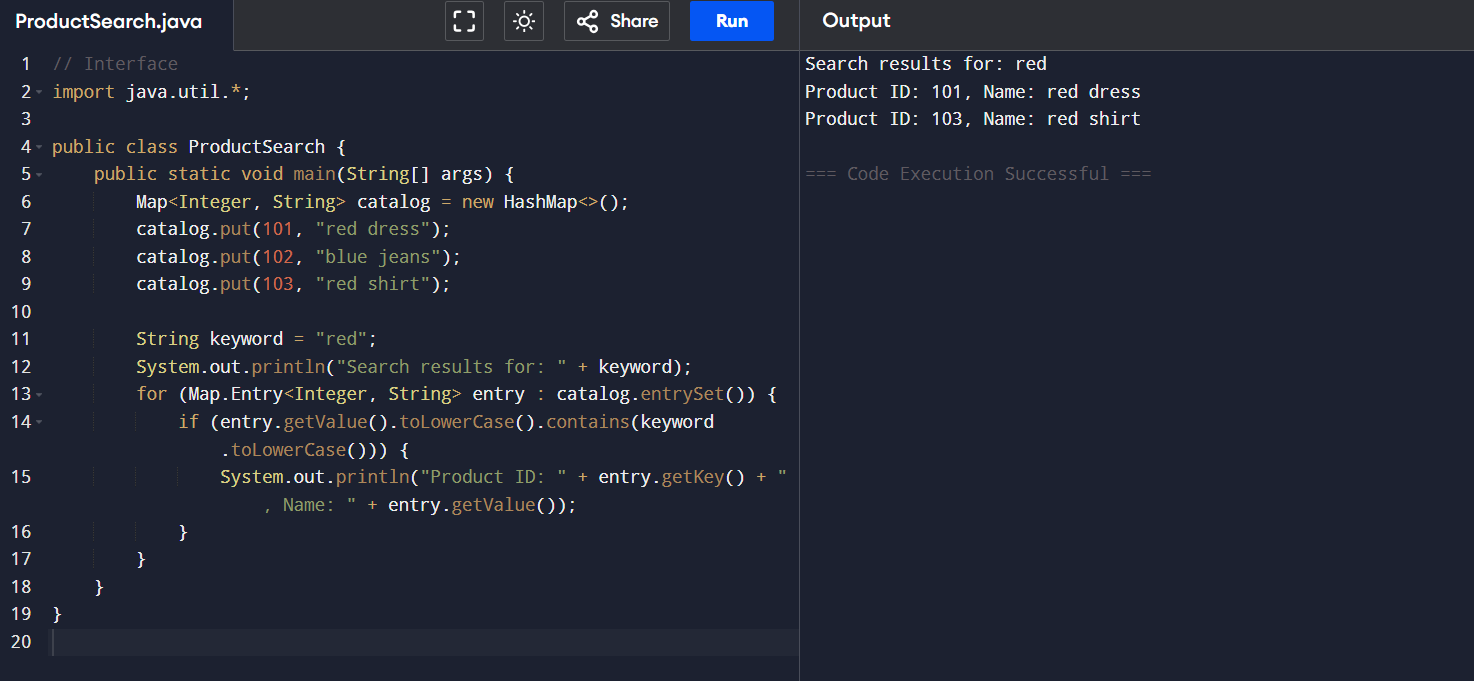
}

}

}

}

**Output:**

****

**2.Financial Forecasting:**

**Code:**

public class FinancialForecast {

public static void main(String[] args) {

int[] years = {2019, 2020, 2021, 2022};

double[] revenue = {100, 150, 200, 250}; // example values in millions

int n = years.length;

double sumX = 0, sumY = 0, sumXY = 0, sumXX = 0;

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

sumX += years[i];

sumY += revenue[i];

sumXY += years[i] \* revenue[i];

sumXX += years[i] \* years[i];

}

double slope = (n \* sumXY - sumX \* sumY) / (n \* sumXX - sumX \* sumX);

double intercept = (sumY - slope \* sumX) / n;

int predictYear = 2023;

double forecast = slope \* predictYear + intercept;

System.out.printf("Forecasted revenue for %d: %.2f million\n", predictYear, forecast);

}

}

**Output:**

