Day-7 & 8 (Collection framework):

Collection Interface

List Interface

ArrayList Class

LinkedList Class

Vector Class

Set Interface

HashSet Class

TreeSet Class

LinkedHashSet Class

Map Interface

HashMap Class

TreeMap Class

LinkedHashMap Class

Hashtable Class

Queue Interface

Priority Queue Class

-----------------------------------------------------------------------------------------------------------------------------------------------------------

Q1.ArrayList Class: Task Scheduler

Charlie is managing his daily tasks and wants to schedule them using an ArrayList. Write a program that allows Charlie to add tasks to his schedule and

display them in the order they were added.

Test Case-1

Input:

Study Code Eat Sleep

Output:

Task schedule:

Study

Code

Eat

Sleep

Test Case-2

Input:

Workout Cook Clean

Output:

Task schedule:

Workout

Cook

Clean

Solution:

import java.util.\*;

class HelloWorld {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

String input[] = sc.nextLine().split(" ");

ArrayList<String> output = new ArrayList<>();

for(String s: input){

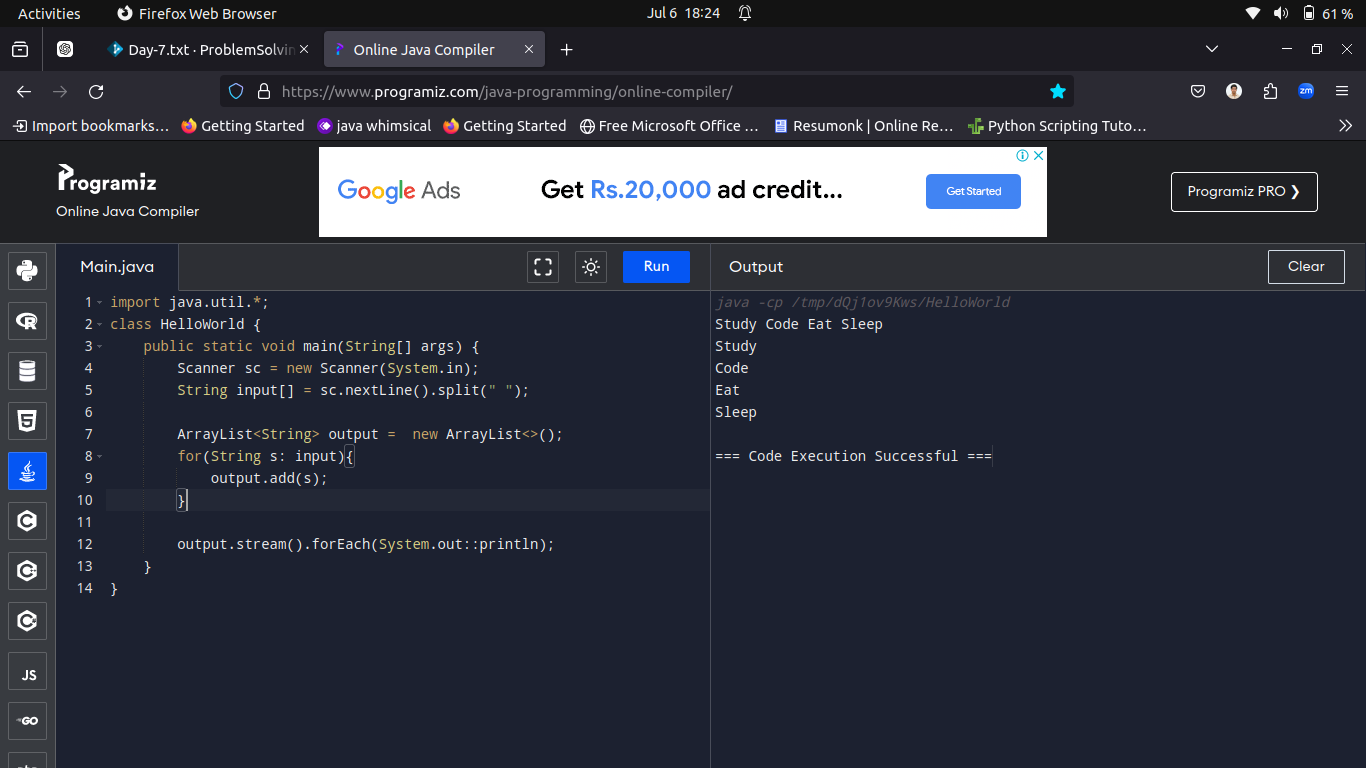
output.add(s);

}

output.stream().forEach(System.out::println);

}

}



-----------------------------------------------------------------------------------------------------------------------------------------------------------

Q2.Vector Class: Employee Payroll Calculator

Problem:

Emma is calculating employee payroll using a Vector. Write a program that allows Emma to enter employee salaries and displays the total payroll.

Test Case-1

Input:

3000 4000 3500 -1

Output:

Total payroll: 10500.0

Test Case-2

Input:

2500 2800 3200 2700 -1

Output:

Total payroll: 11200.0

Solution:

import java.util.Scanner;

import java.util.Vector;

public class PayrollCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Vector<Double> salaries = new Vector<>();

while (true) {

double salary = scanner.nextDouble();

if (salary == -1) {

break;

}

salaries.add(salary);

}

double totalPayroll = 0;

for (double salary : salaries) {

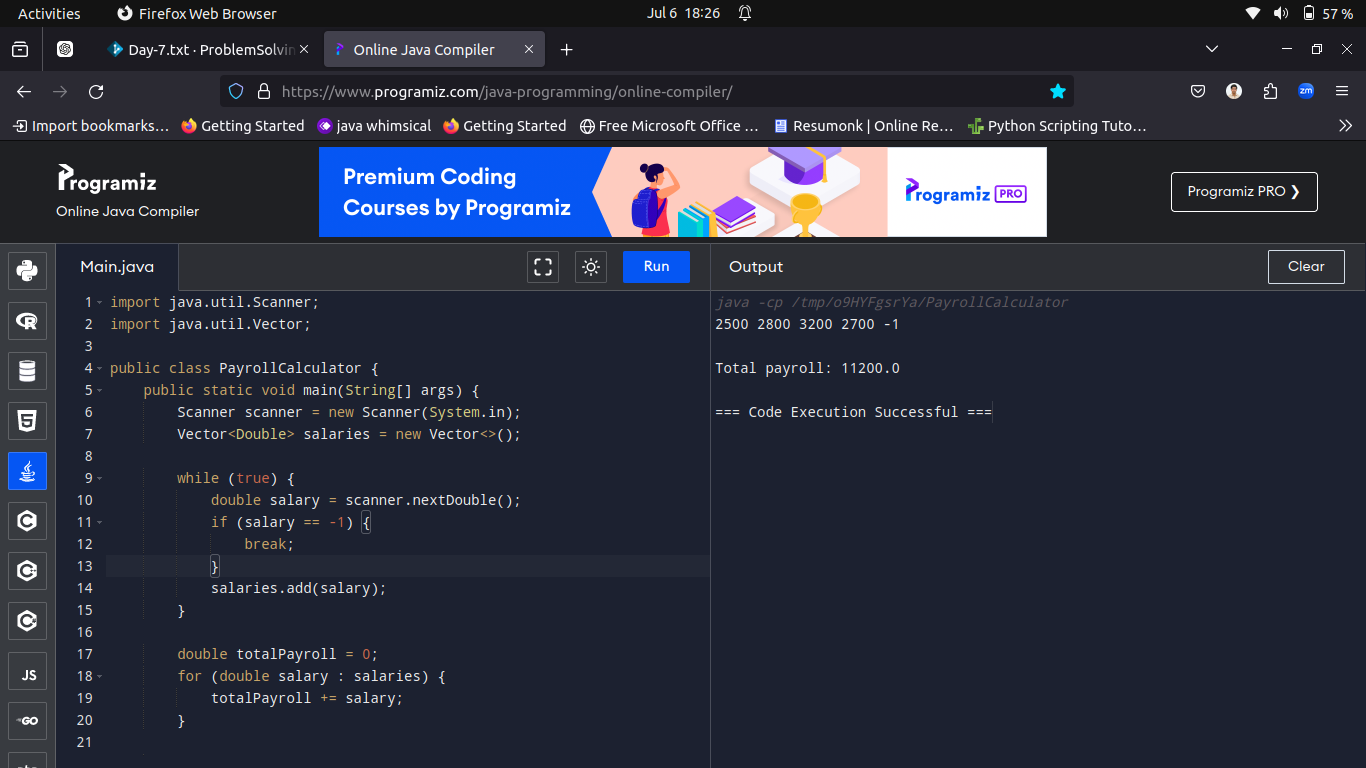
totalPayroll += salary;

}

System.out.println("\nTotal payroll: " + totalPayroll);

}

}



-----------------------------------------------------------------------------------------------------------------------------------------------------------

Q3.In a theme park, there are different zones with unique visitor counts. Write a program that takes input for the visitor count of each zone using the

Collection interface and calculates the total number of visitors.

Test Case-1

Input:

100 150 200 250

Output:

Total visitors in the theme park: 700

Test Case-2

Input:

50 75 100

Output:

Total visitors in the theme park: 225

Solution:

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

public class ThemeParkVisitorCounter {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

List<Integer> visitorCounts = new ArrayList<>();

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

String[] tokens = inputLine.split("\\s+");

for (String token : tokens) {

int count = Integer.parseInt(token);

visitorCounts.add(count);

}

int totalVisitors = 0;

for (int count : visitorCounts) {

totalVisitors += count;

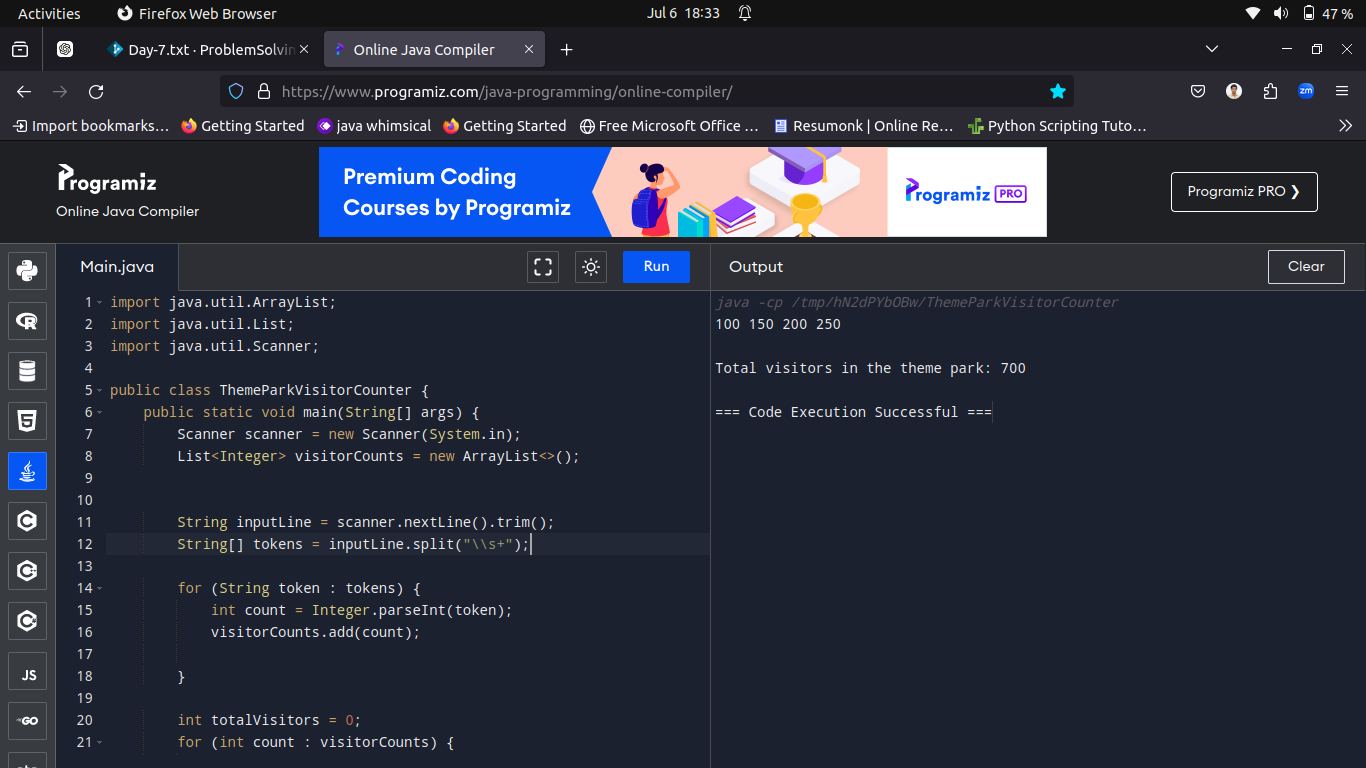
}

System.out.println("\nTotal visitors in the theme park: " + totalVisitors);

scanner.close();

}

}



-----------------------------------------------------------------------------------------------------------------------------------------------------------

Q4.Write a Java program to remove duplicates from an ArrayList.

Test Case-1

5

1 2 3 2 1

Solution:

import java.util.ArrayList;

import java.util.LinkedHashSet;

import java.util.Scanner;

public class RemoveDuplicates {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

ArrayList<Integer> list = new ArrayList<>();

int size = scanner.nextInt();

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

int num = scanner.nextInt();

list.add(num);

}

LinkedHashSet<Integer> set = new LinkedHashSet<>(list);

list.clear();

list.addAll(set);

System.out.println("List after removing duplicates:");

for (int num : list) {

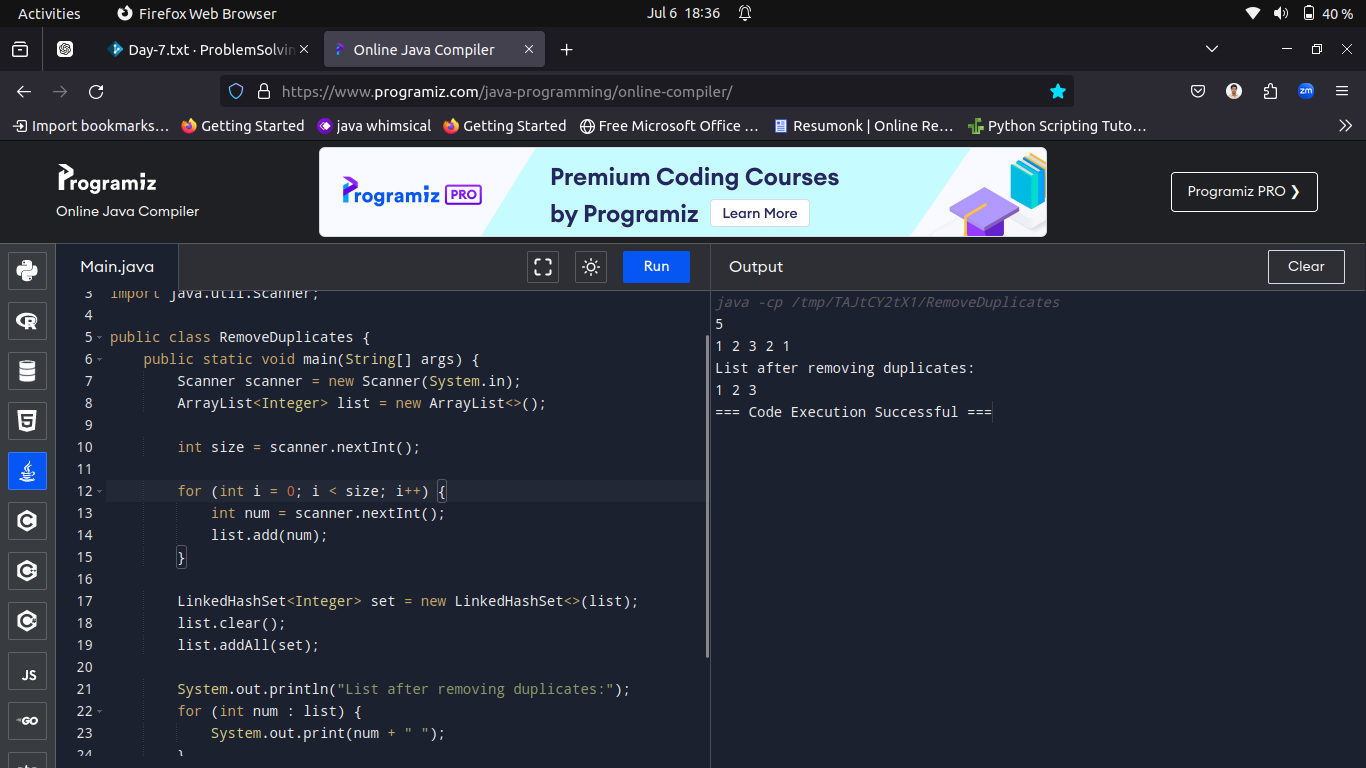
System.out.print(num + " ");

}

scanner.close();

}

}



-----------------------------------------------------------------------------------------------------------------------------------------------------------

Q5.Write a Java program to sort elements of an ArrayList in descending order.

Test Case-1

Input:

5

5 2 7 3 1

Output:

[7, 5, 3, 2, 1]

Solution:

import java.util.ArrayList;

import java.util.Collections;

import java.util.Scanner;

public class SortArrayListDescending {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

ArrayList<Integer> list = new ArrayList<>();

int size = scanner.nextInt();

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

int num = scanner.nextInt();

list.add(num);

}

Collections.sort(list, Collections.reverseOrder());

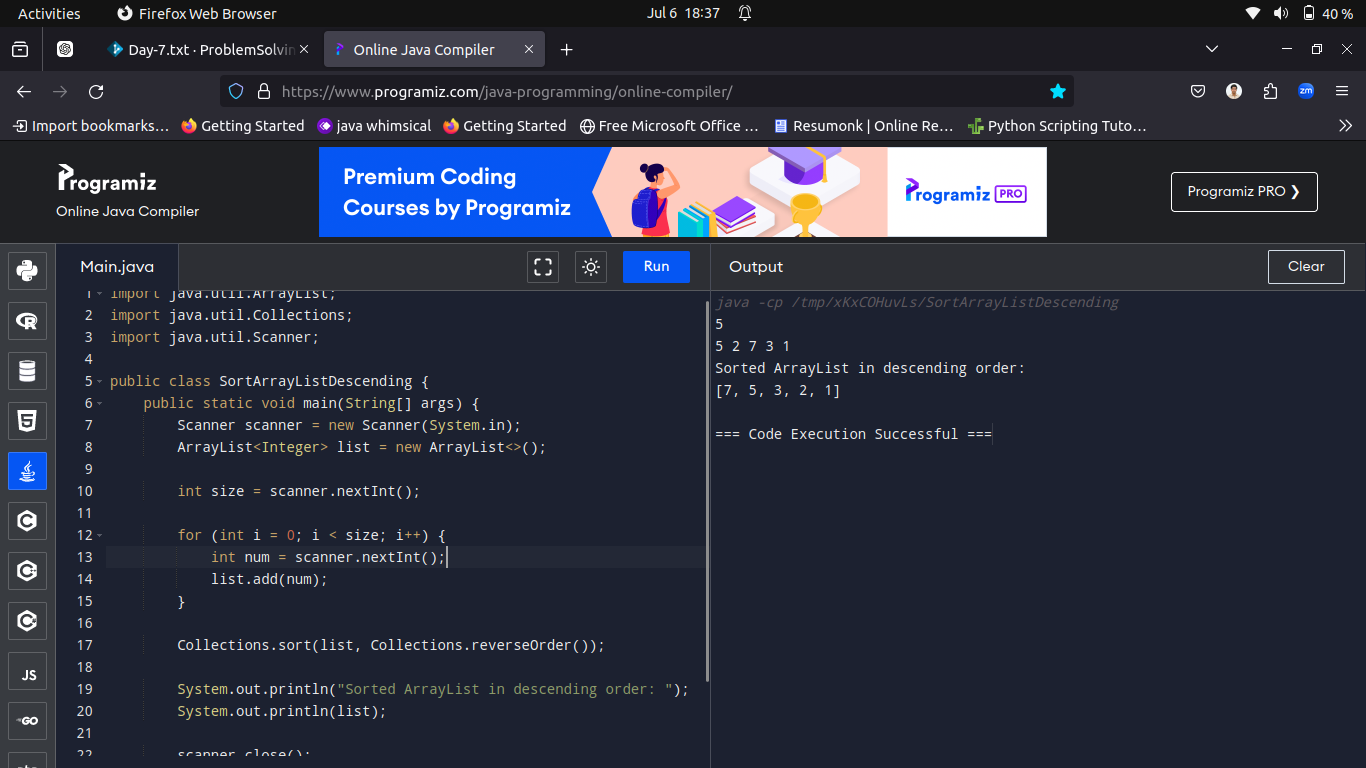
System.out.println("Sorted ArrayList in descending order: ");

System.out.println(list);

scanner.close();

}

}



-----------------------------------------------------------------------------------------------------------------------------------------------------------

Q6.Write a Java program to find the maximum and minimum element in an ArrayList.

Test Case-1

Input:

5

5 2 7 3 1

Output:

Maximum: 7

Minimum: 1

Solution:

import java.util.ArrayList;

import java.util.Scanner;

public class MaxMinArrayList {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

ArrayList<Integer> list = new ArrayList<>();

int size = scanner.nextInt();

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

int num = scanner.nextInt();

list.add(num);

}

int max = Integer.MIN\_VALUE;

int min = Integer.MAX\_VALUE;

for (int num : list) {

if (num > max) {

max = num;

}

if (num < min) {

min = num;

}

}

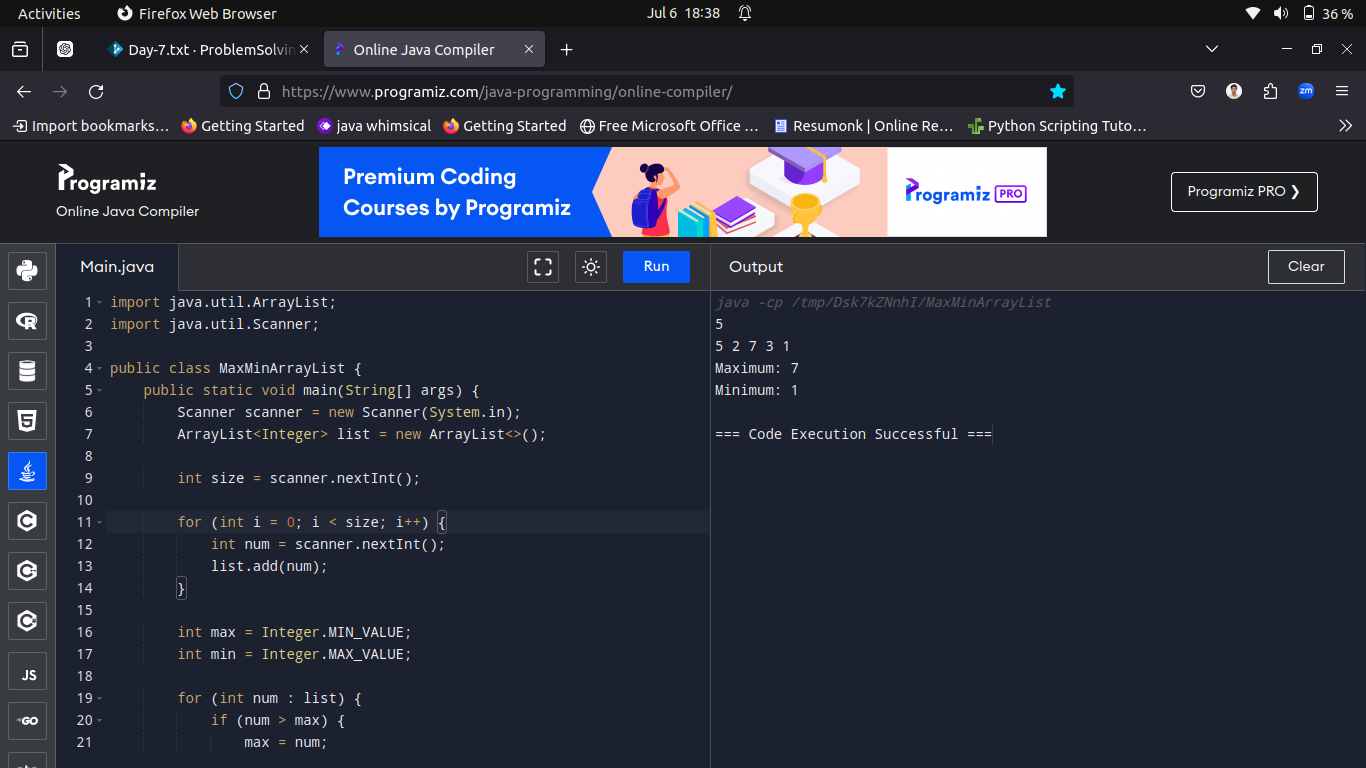
System.out.println("Maximum: " + max);

System.out.println("Minimum: " + min);

scanner.close();

}

}



-----------------------------------------------------------------------------------------------------------------------------------------------------------

Q7.Write a Java program to check if an ArrayList contains a given element.

Test Case-1

Input:

5

5 2 7 3 1

3

Output:

Element found

Solution:

import java.util.ArrayList;

import java.util.Scanner;

public class CheckElementInArrayList {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

ArrayList<Integer> list = new ArrayList<>();

int size = scanner.nextInt();

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

int num = scanner.nextInt();

list.add(num);

}

int elementToCheck = scanner.nextInt();

if (list.contains(elementToCheck)) {

System.out.println("Element found");

} else {

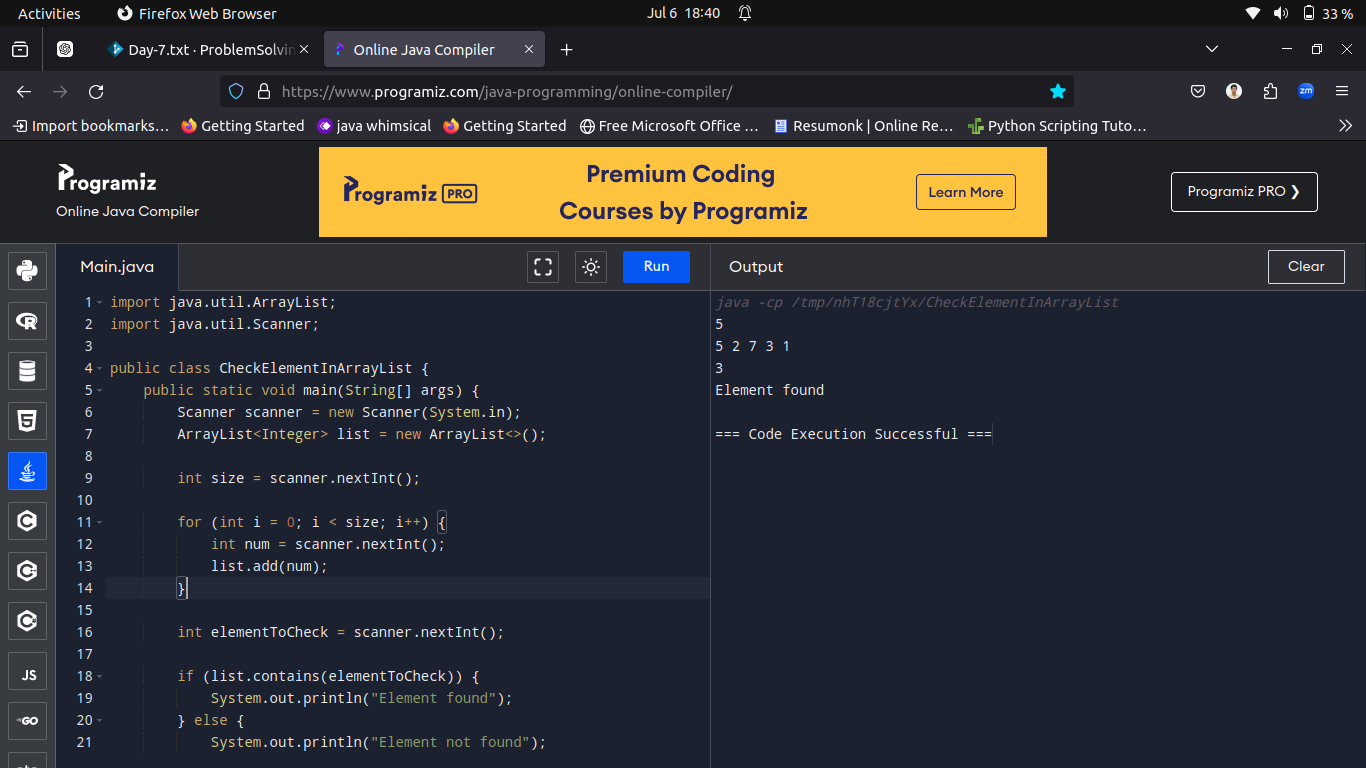
System.out.println("Element not found");

}

scanner.close();

}

}



-----------------------------------------------------------------------------------------------------------------------------------------------------------

Q8.Write a Java program to count the occurrences of each element in an ArrayList and display the elements along with their frequencies.

Input Format:

The first line contains an integer n, the size of the ArrayList.

The next line contains n space-separated integers representing the elements of the ArrayList.

Output Format:

Print each element of the ArrayList along with its frequency.

Test Case-1

7

1 2 2 3 3 3 4

Output:

Element 1 occurs 1 time

Element 2 occurs 2 times

Element 3 occurs 3 times

Element 4 occurs 1 time

Solution:

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Scanner;

public class CountOccurrencesInArrayList {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int n = scanner.nextInt();

ArrayList<Integer> list = new ArrayList<>();

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

int num = scanner.nextInt();

list.add(num);

}

HashMap<Integer, Integer> frequencyMap = new HashMap<>();

for (int num : list) {

if (frequencyMap.containsKey(num)) {

frequencyMap.put(num, frequencyMap.get(num) + 1);

} else {

frequencyMap.put(num, 1);

}

}

for (int num : frequencyMap.keySet()) {

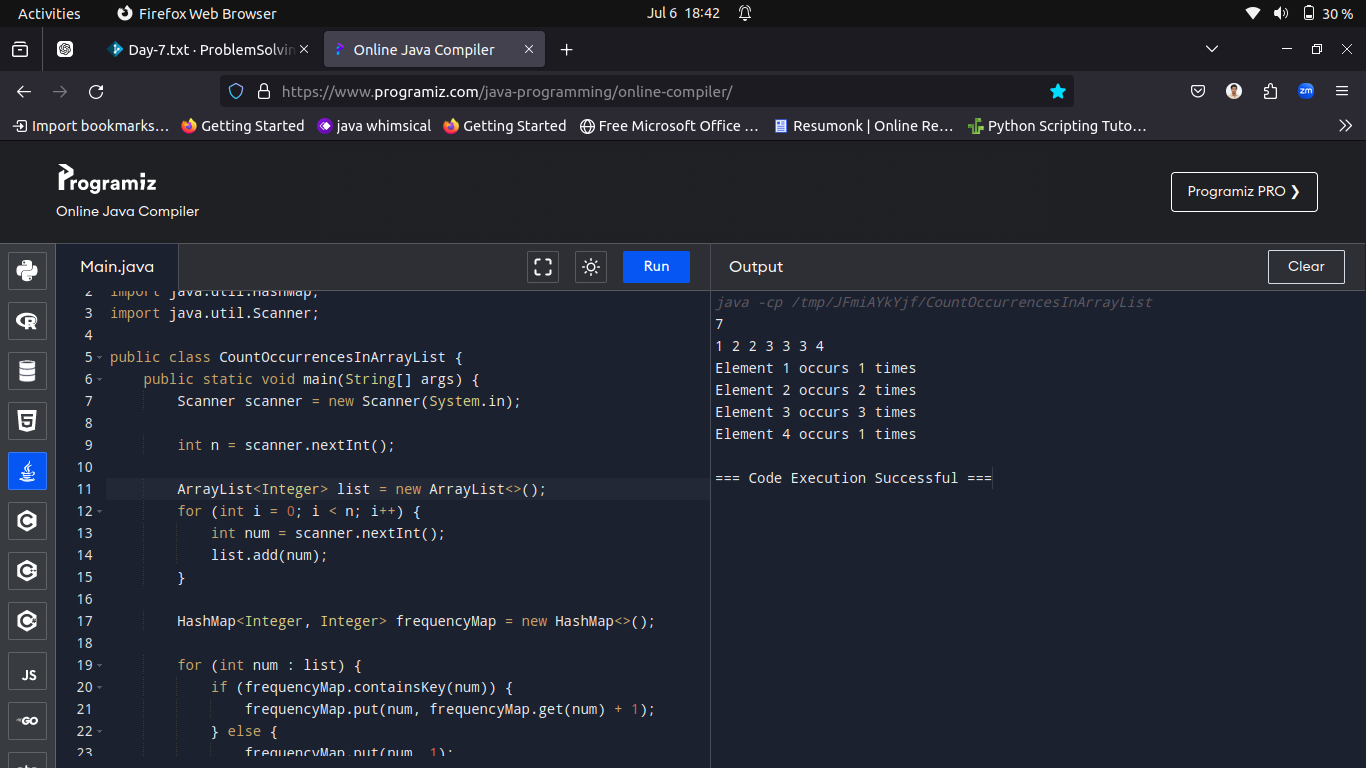
System.out.println("Element " + num + " occurs " + frequencyMap.get(num) + " times");

}

scanner.close();

}

}



-----------------------------------------------------------------------------------------------------------------------------------------------------------

Q9.Write a Java program to find the index of the first occurrence of a specific element in an ArrayList.

Input Format:

The first line contains an integer n, the size of the ArrayList.

The next line contains n space-separated integers representing the elements of the ArrayList.

The third line contains an integer x, the element whose index needs to be found.

Output Format:

Print the index of the first occurrence of the specified element.

Test Case-1

Input:

6

1 2 2 3 3 3

3

Output:

Index of first occurrence of element 3: 3

Solution:

import java.util.ArrayList;

import java.util.Scanner;

public class FindIndexOfElementInArrayList {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int n = scanner.nextInt();

ArrayList<Integer> list = new ArrayList<>();

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

int num = scanner.nextInt();

list.add(num);

}

int x = scanner.nextInt();

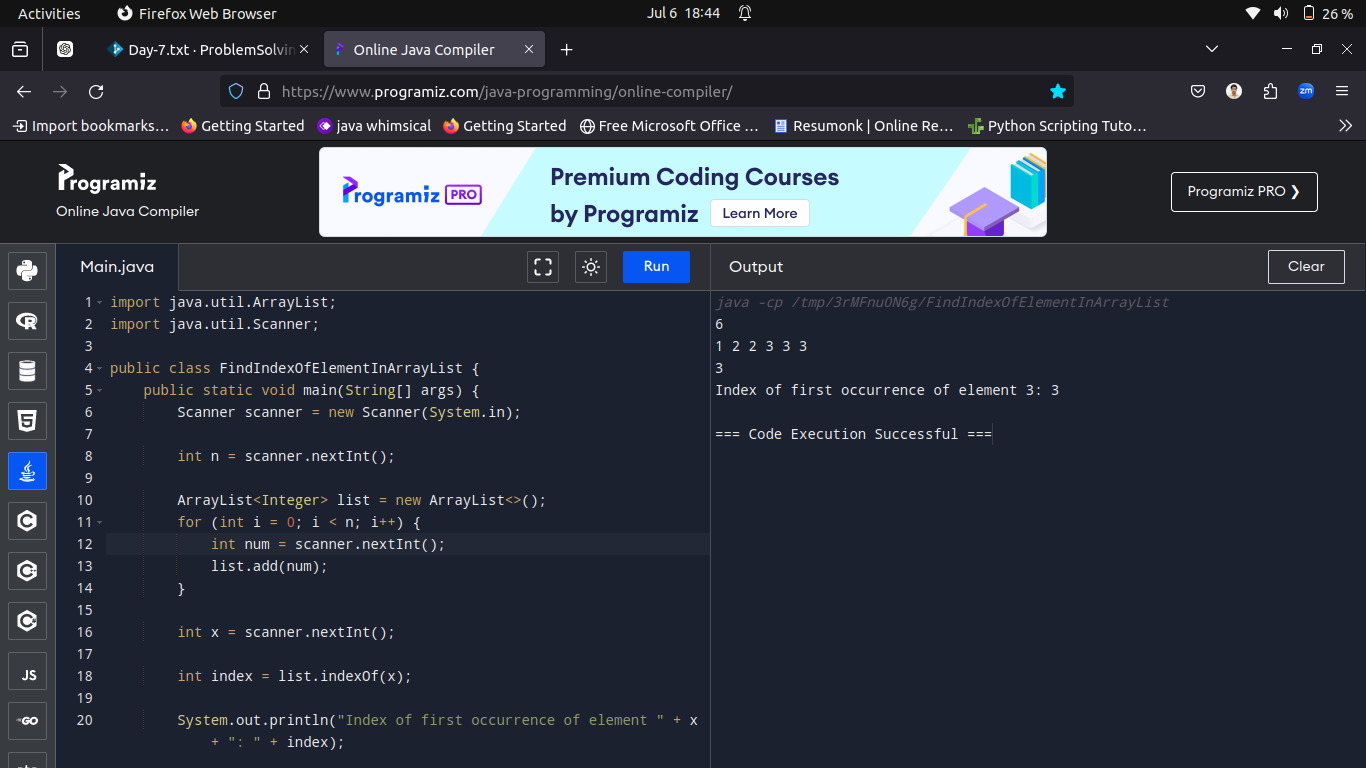
int index = list.indexOf(x);

System.out.println("Index of first occurrence of element " + x + ": " + index);

scanner.close();

}

}



-----------------------------------------------------------------------------------------------------------------------------------------------------------

Q10.Write a Java program to get a portion of an ArrayList between two specified indices.

Input Format:

The first line contains an integer n, the size of the ArrayList.

The next line contains n space-separated integers representing the elements of the ArrayList.

The third line contains an integer start, the starting index (inclusive).

The fourth line contains an integer end, the ending index (exclusive).

Output Format:

Print the portion of the ArrayList between the specified indices.

Testcase-1

Sample Input:

5

1 2 3 4 5

1

4

Sample Output:

Portion of ArrayList: [2, 3, 4]

Solution:

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

public class SublistOfArrayList {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int n = scanner.nextInt();

ArrayList<Integer> list = new ArrayList<>();

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

int num = scanner.nextInt();

list.add(num);

}

int start = scanner.nextInt();

int end = scanner.nextInt();

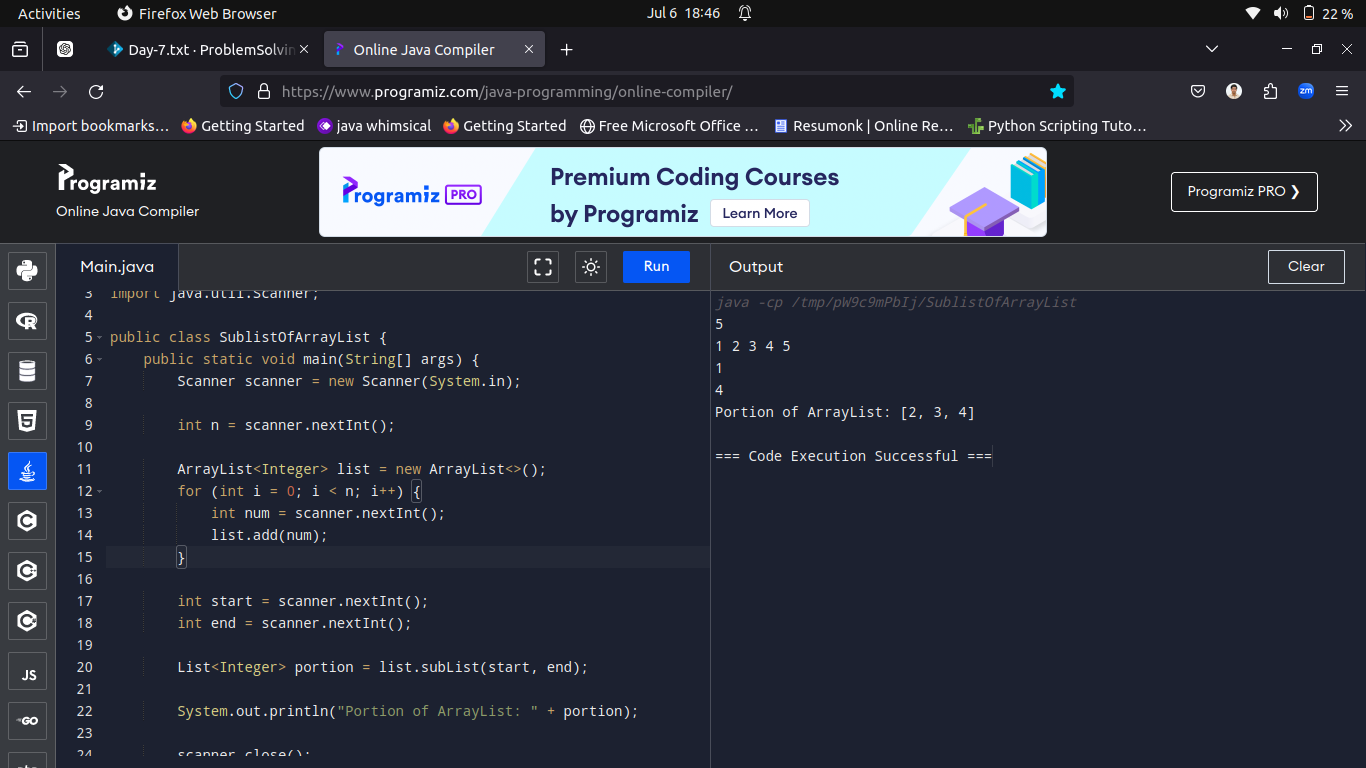
List<Integer> portion = list.subList(start, end);

System.out.println("Portion of ArrayList: " + portion);

scanner.close();

}

}



-----------------------------------------------------------------------------------------------------------------------------------------------------------