ASSIGNMENT – 2

Write a program to enter the marks of a student in four subjects. Then calculate the total and aggregate, display the grade obtained by the student. If the student scores an aggregate greater than 75%, then the grade is Distinction. If aggregate is 60>= and <75, then the grade is First Division. If aggregate is 50 >= and <60, then the grade is Second Division. If aggregate is 40>= and <50, then the grade is Third Division. Else the grade is Fail.

Sample Input & Output:

Enter the marks in python: 90

Enter the marks in c programming: 91

Enter the marks in Mathematics: 92

Enter the marks in Physics: 93

Total= 366

Aggregate = 91.5

DISTINCTION

Test cases:

a) 18, 76,93,65

b) 73,78,79,75

c) 98,106,120,95

d) 96,73, -85,95

e) 78,59.8,76,79

PROGRAM:

import java.util.Scanner;

public class StudentGrades {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the marks in Python: ");

double pythonMarks = scanner.nextDouble();

System.out.print("Enter the marks in C Programming: ");

double cMarks = scanner.nextDouble();

System.out.print("Enter the marks in Mathematics: ");

double mathMarks = scanner.nextDouble();

System.out.print("Enter the marks in Physics: ");

double physicsMarks = scanner.nextDouble();

double total = pythonMarks + cMarks + mathMarks + physicsMarks;

double aggregate = (total / 4) \* 100;

System.out.println("Total: " + total);

System.out.println("Aggregate: " + aggregate);

if (aggregate > 75) {

System.out.println("DISTINCTION");

} else if (aggregate >= 60 && aggregate < 75) {

System.out.println("FIRST DIVISION");

} else if (aggregate >= 50 && aggregate < 60) {

System.out.println("SECOND DIVISION");

} else if (aggregate >= 40 && aggregate < 50) {

System.out.println("THIRD DIVISION");

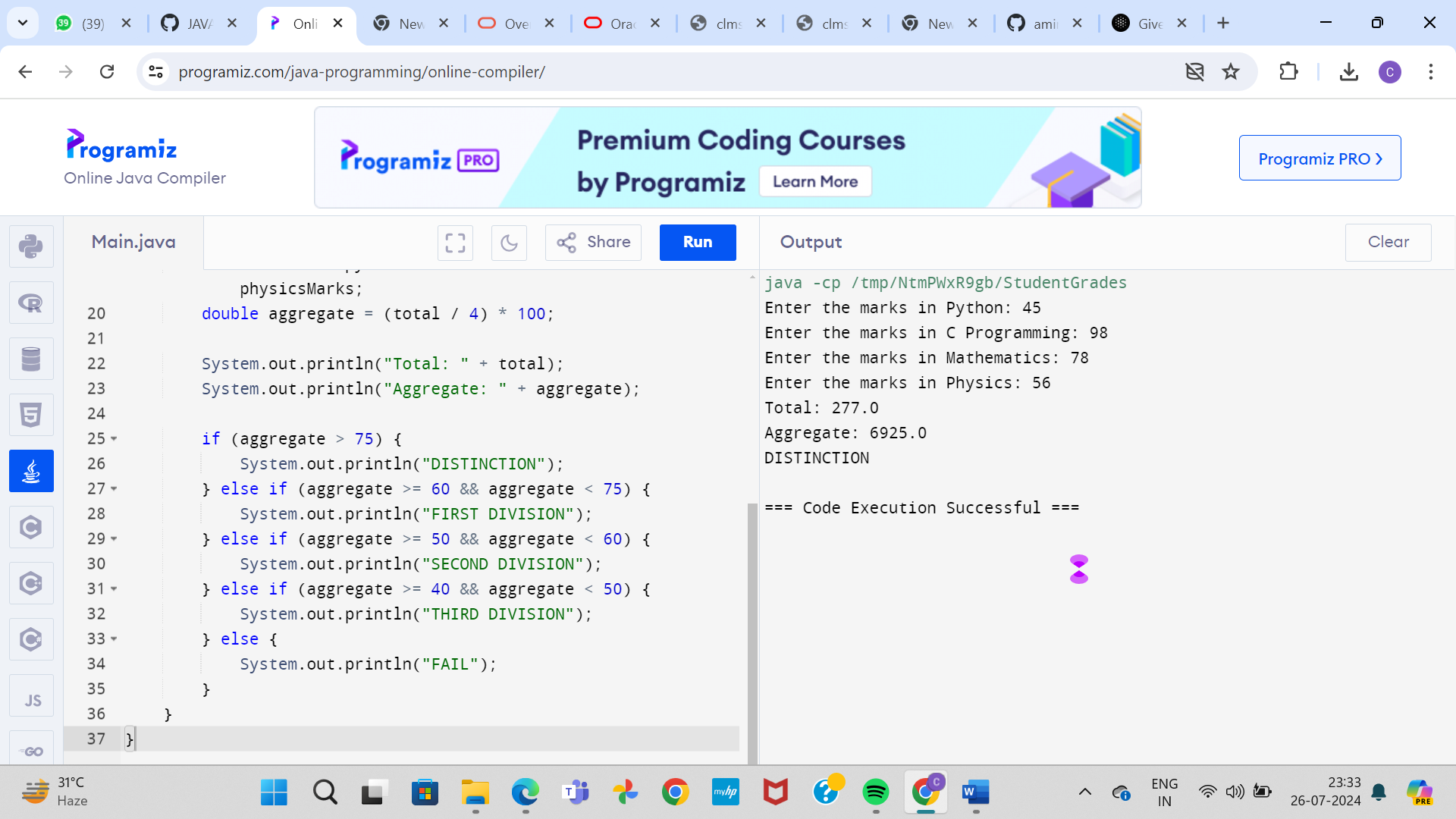
} else {

System.out.println("FAIL");

}

}

}



. Write a program to calculate tax given the following conditions:

a. If income is less than or equal to 1,50,000 then no tax

b. If taxable income is 1,50,001 – 3,00,000 the charge 10% tax

c. If taxable income is 3,00,001 – 5,00,000 the charge 20% tax

d. If taxable income is above 5,00,001 then charge 30% tax

Sample Input:

Enter the income:200000

Sample Output:

Tax= 20000

Test cases:

1. 400700

2. 2789239

3. 150000

4. 00000

5. -125486

PROGRAM:

import java.util.Scanner;

public class TaxCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

double income = scanner.nextDouble();

double tax = 0;

if (income <= 150000) {

tax = 0;

} else if (income > 150000 && income <= 300000) {

tax = (income - 150000) \* 0.10;

} else if (income > 300000 && income <= 500000) {

tax = (300000 - 150000) \* 0.10 + (income - 300000) \* 0.20;

} else if (income > 500000) {

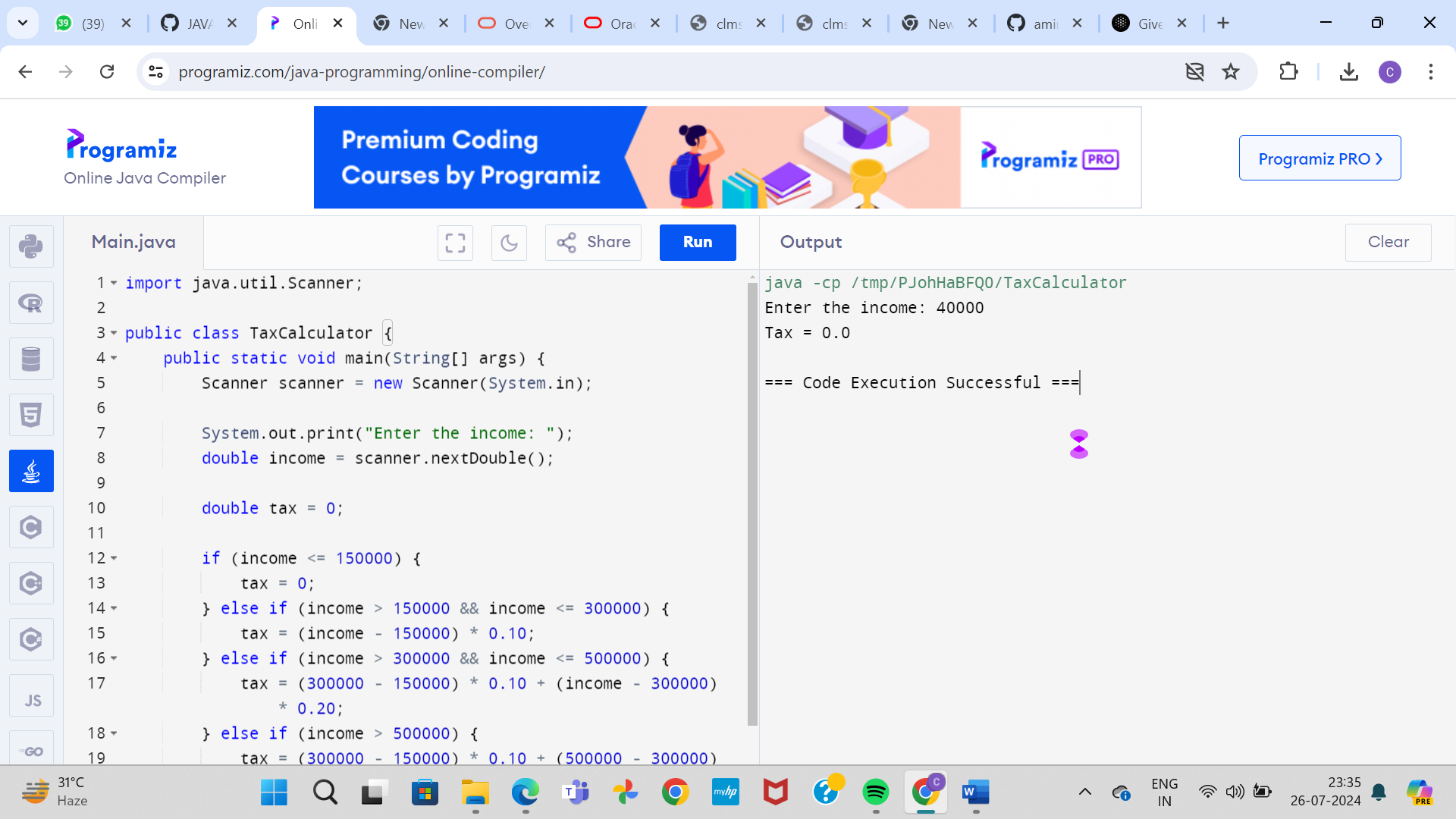
tax = (300000 - 150000) \* 0.10 + (500000 - 300000) \* 0.20 + (income - 500000) \* 0.30;

}

System.out.println("Tax = " + tax);

}

}



3. Write a program to print the first n perfect numbers. (Hint Perfect number means a positive integer that is equal to the sum of its proper divisors)

Sample Input:

N = 3

Sample Output:

First 3 perfect numbers are: 6 , 28 , 496

Test Cases:

1. N = 0

2. N = 5

3. N = -2

4. N = -5

5. N = 0.2

PROGRAM:

import java.util.Scanner;

public class PerfectNumbers {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the value of N: ");

int n = scanner.nextInt();

if (n <= 0) {

System.out.println("Invalid input. N should be a positive integer.");

return;

}

int count = 0;

int num = 1;

System.out.print("First " + n + " perfect numbers are: ");

while (count < n) {

if (isPerfect(num)) {

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

count++;

}

num++;

}

}

public static boolean isPerfect(int num) {

int sum = 0;

for (int i = 1; i < num; i++) {

if (num % i == 0) {

sum += i;

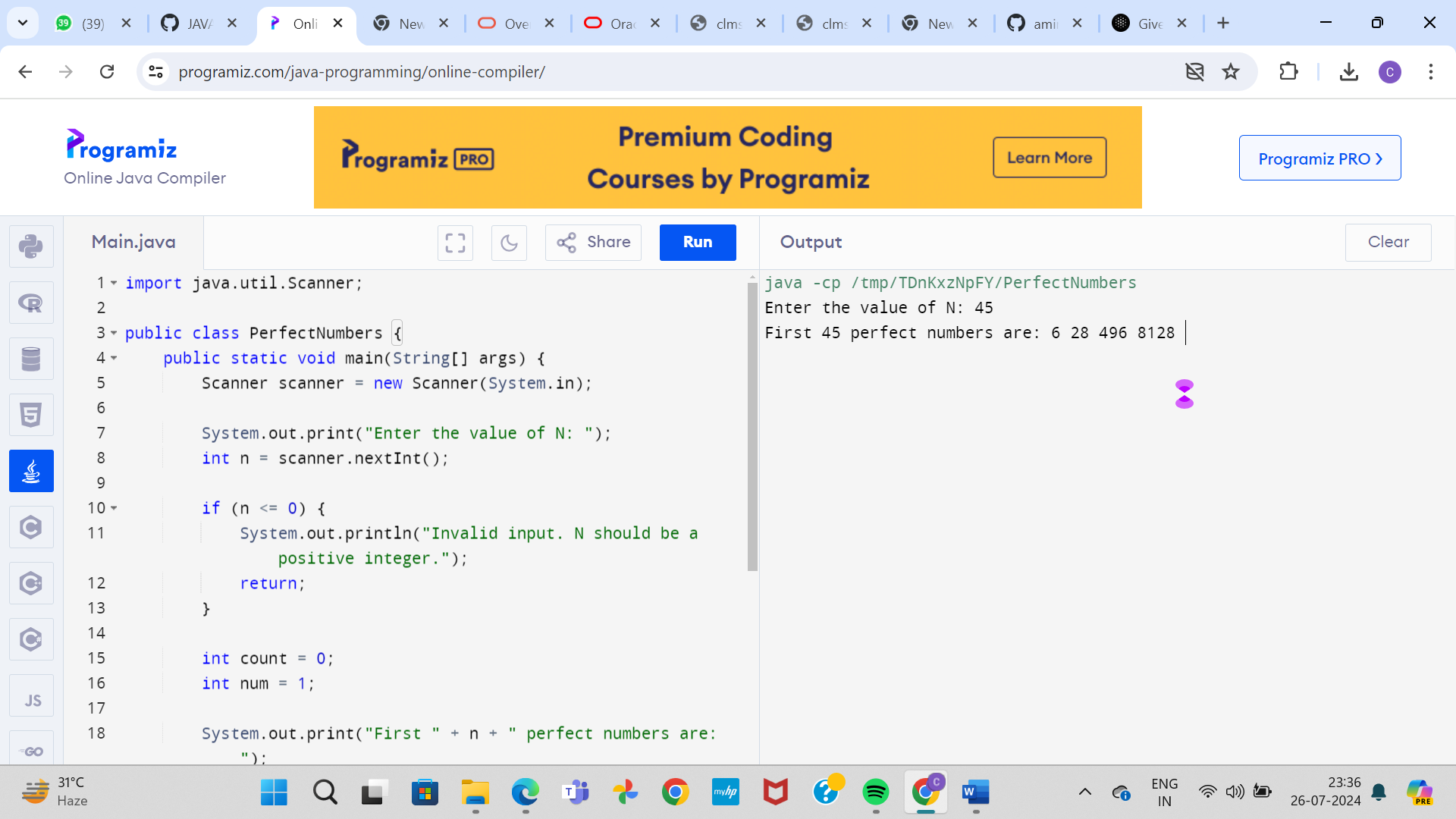
}

}

return sum == num;

}

}



16. Write a Program to Find the Nth Largest Number in a array.

Sample Input:

List : {14, 67, 48, 23, 5, 62}

N = 4

Sample Output:

4th Largest number: 23

Test cases:

1. N = 0

2. N = -5

3. N = 1

4. N = M

5. N = %

PROGRAM:

import java.util.Arrays;

import java.util.Scanner;

public class NthLargestNumber {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the list of numbers (separated by space): ");

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

int[] arr = new int[input.length];

for (int i = 0; i < input.length; i++) {

arr[i] = Integer.parseInt(input[i]);

}

System.out.print("Enter the value of N: ");

int n = scanner.nextInt();

if (n <= 0) {

System.out.println("Invalid input. N should be a positive integer.");

return;

}

if (n > arr.length) {

System.out.println("N is larger than the size of the array.");

return;

}

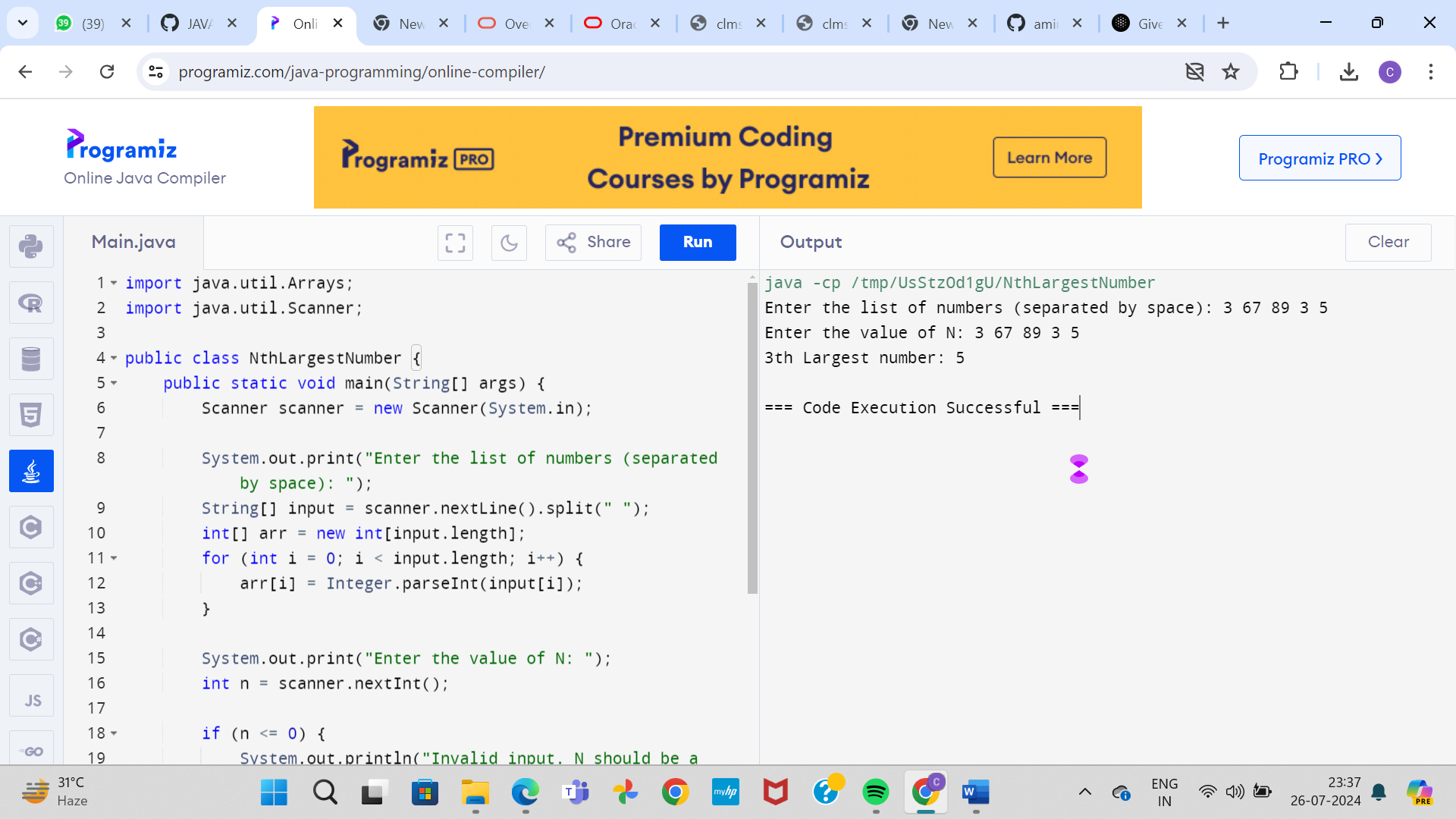
Arrays.sort(arr);

int nthLargest = arr[arr.length - n];

System.out.println(n + "th Largest number: " + nthLargest);

}

}



18. Write a program to find the number of special characters in the given statement

Sample Input:

Given statement: Modi Birthday @ September 17, #&$% is the wishes code for him.

Sample Output:

Number of special Characters: 5

PROGRAM:

import java.util.Arrays;

import java.util.Scanner;

public class NthLargestNumber {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the list of numbers (separated by space): ");

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

int[] arr = new int[input.length];

for (int i = 0; i < input.length; i++) {

arr[i] = Integer.parseInt(input[i]);

}

System.out.print("Enter the value of N: ");

int n = scanner.nextInt();

if (n <= 0) {

System.out.println("Invalid input. N should be a positive integer.");

return;

}

if (n > arr.length) {

System.out.println("N is larger than the size of the array.");

return;

}

Arrays.sort(arr);

int[] sortedArr = new int[arr.length];

System.arraycopy(arr, 0, sortedArr, 0, arr.length);

for (int i = 0; i < n - 1; i++) {

sortedArr = removeSmallest(sortedArr);

}

int nthLargest = sortedArr[0];

System.out.println(n + "th Largest number: " + nthLargest);

}

public static int[] removeSmallest(int[] arr) {

int minIndex = 0;

for (int i = 1; i < arr.length; i++) {

if (arr[i] < arr[minIndex]) {

minIndex = i;

}

}

int[] newArr = new int[arr.length - 1];

System.arraycopy(arr, 0, newArr, 0, minIndex);

System.arraycopy(arr, minIndex + 1, newArr, minIndex, arr.length - minIndex - 1);

return newArr;

}

}

