Day-3(Arrays & Strings):

Array Manipulation & Operations

Traversing Arrays and Strings

Searching & Sorting

Array Rotation

Matrix Operation

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

Q1.You are playing the PUBG game and you entered into the Bootcamp. There you viewed the drop box which was filled with guns. you have to choosethe

biggest gun in the drop box. Find the biggest gun and you will get the chicken dinner.

Input Format:

Input consists of n+1 integers.

The first integer corresponds to ‘n’ , the size of the array.

The next ‘n’ integers correspond to the elements in the array. Assume that the maximum value of n is 15.

Output Format:

Refer sample output for details.

Sample Input 1:

5

2

3

6

8

1

Sample Output 1:

8 is the maximum element in the array

Case 1

Case 2

Input (stdin)

4

1

5

8

9

Output (stdout)

9 is the maximum element in the array

Solution:

import java.util.\*;

class HelloWorld {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Number of guns

int n = sc.nextInt();

int guns[] = new int[n];

int max = Integer.MIN\_VALUE;

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

int temp = sc.nextInt();

guns[i] = temp;

// Check the gun with max size

if(max<guns[i])

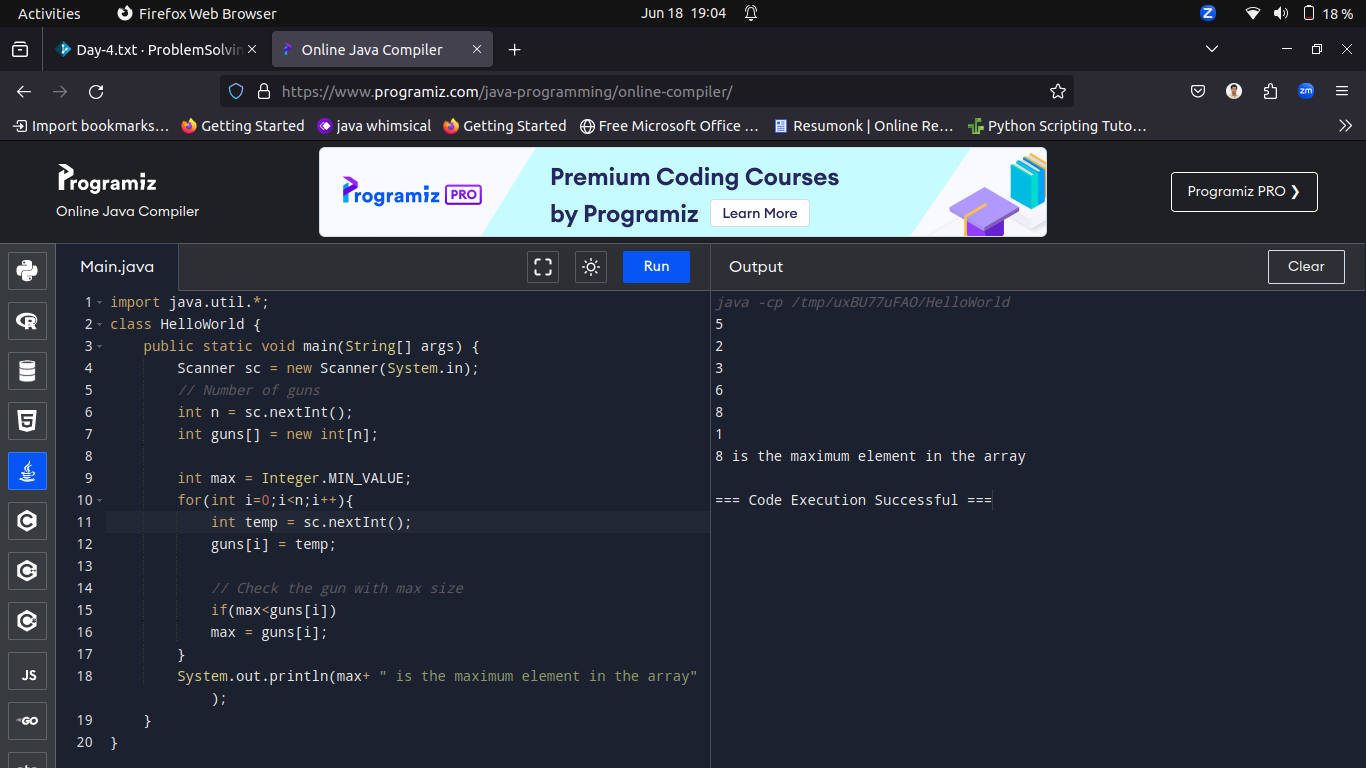
max = guns[i];

}

System.out.println(max+ " is the maximum element in the array");

}

}



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

Q2.Sum of 2 arrays

Ramu have some number of Apples and he arrange that in a matrix format. Raju have some another number of Apples. He also wants to arrange that in a matrix format. Ragul wants to calculate the total number of apples. Help him to find out the total number of apples.

Input Format:

Input consists of 2n+1 integers. The first integer corresponds to ‘n’ , the size of the array. The next ‘n’ integers correspond to the elements in the first array. The last 'n' integers correspond to the elements in the second array. Assume that the maximum value of n is 15.

Output Format:

Refer sample output for details.

Sample Input:

5

2

3

6

8

1

1

1

1

1

1

Sample Output:

3 4 7 9 2

Case 1

Input (stdin)

3

1

2

3

1

2

3

Output (stdout)

2

4

6

Case 2

Input (stdin)

2

3

6

1

2

Output (stdout)

4

8

Solution:

import java.util.Scanner;

public class SumOfArrays {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int n = scanner.nextInt();

int[] array1 = new int[n];

int[] array2 = new int[n];

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

array1[i] = scanner.nextInt();

}

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

array2[i] = scanner.nextInt();

}

int[] sumArray = new int[n];

// Compute the sum of corresponding elements

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

sumArray[i] = array1[i] + array2[i];

}

System.out.println("Output:");

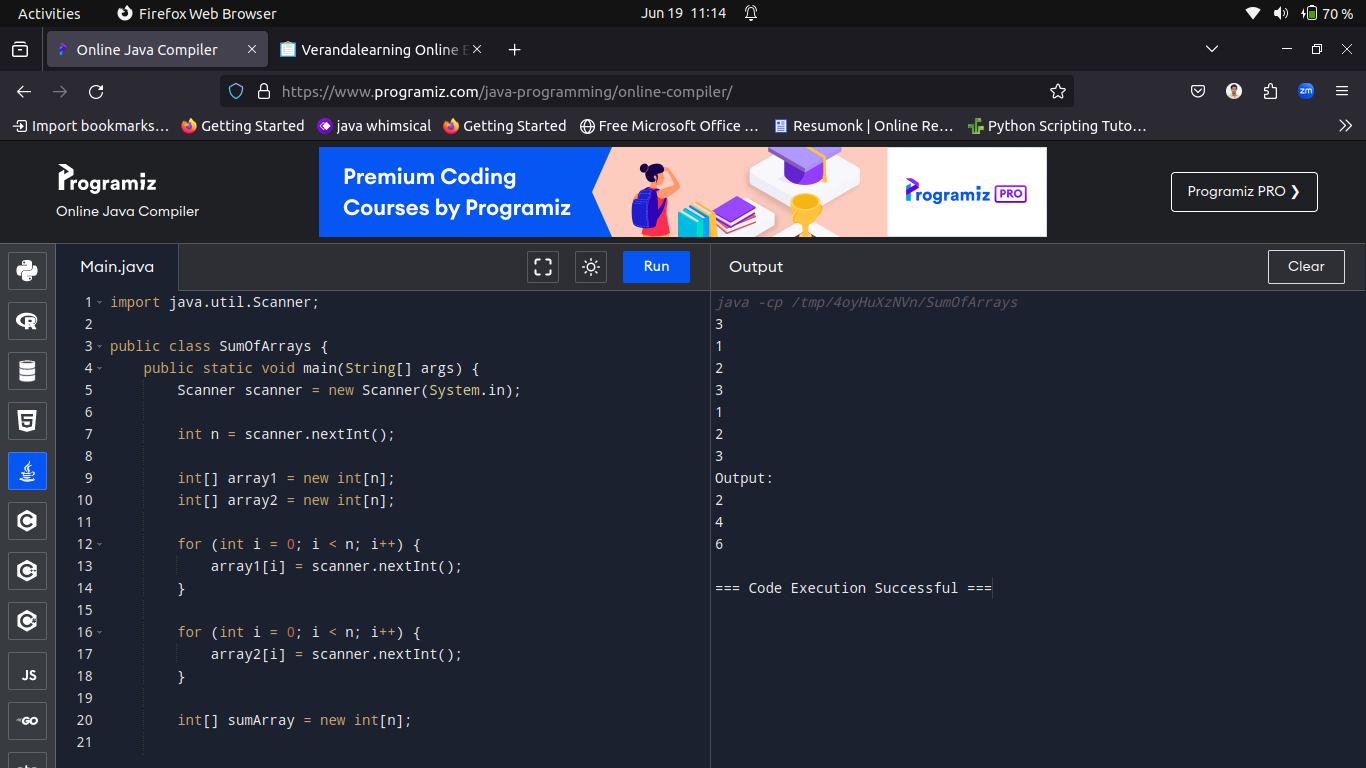
for (int i = 0; i < n; i++) {

System.out.println(sumArray[i]);

}

}

}



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

Q3.Write a program to find whether 2 arrays are compatible or not. Two arrays are said to be compatible if they are of the same size and if the ith element in the first array is greater than or equal to the ith element in the second array for all the values of i.

INPUT FORMAT:

Input consists of 2n+1 integers.

The first integer corresponds to ‘n’, the size of the array.

The next ‘n’ integers correspond to the elements in the first array.

The last 'n' integers correspond to the elements in the second array.

Assume that the maximum value of n is 15.

OUTPUT FORMAT:

Refer sample output for details.

SAMPLE INPUT:

5

2

3

6

8

1

1

1

1

1

1

SAMPLE OUTPUT:

Compatible

Case 1

Input (stdin)

5

2

3

6

8

1

1

1

1

1

2

Output (stdout)

Incompatible

Solution:

import java.util.Scanner;

public class HelloWorld {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Read input size 'n'

int n = scanner.nextInt();

// Read array1

int[] array1 = new int[n];

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

array1[i] = scanner.nextInt();

}

// Read array2

int[] array2 = new int[n];

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

array2[i] = scanner.nextInt();

}

// Check compatibility

if (areCompatible(array1, array2)) {

System.out.println("Compatible");

} else {

System.out.println("Not Compatible");

}

scanner.close();

}

public static boolean areCompatible(int[] array1, int[] array2) {

// Assume array1.length == array2.length

int n = array1.length;

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

if (array1[i] < array2[i]) {

return false;

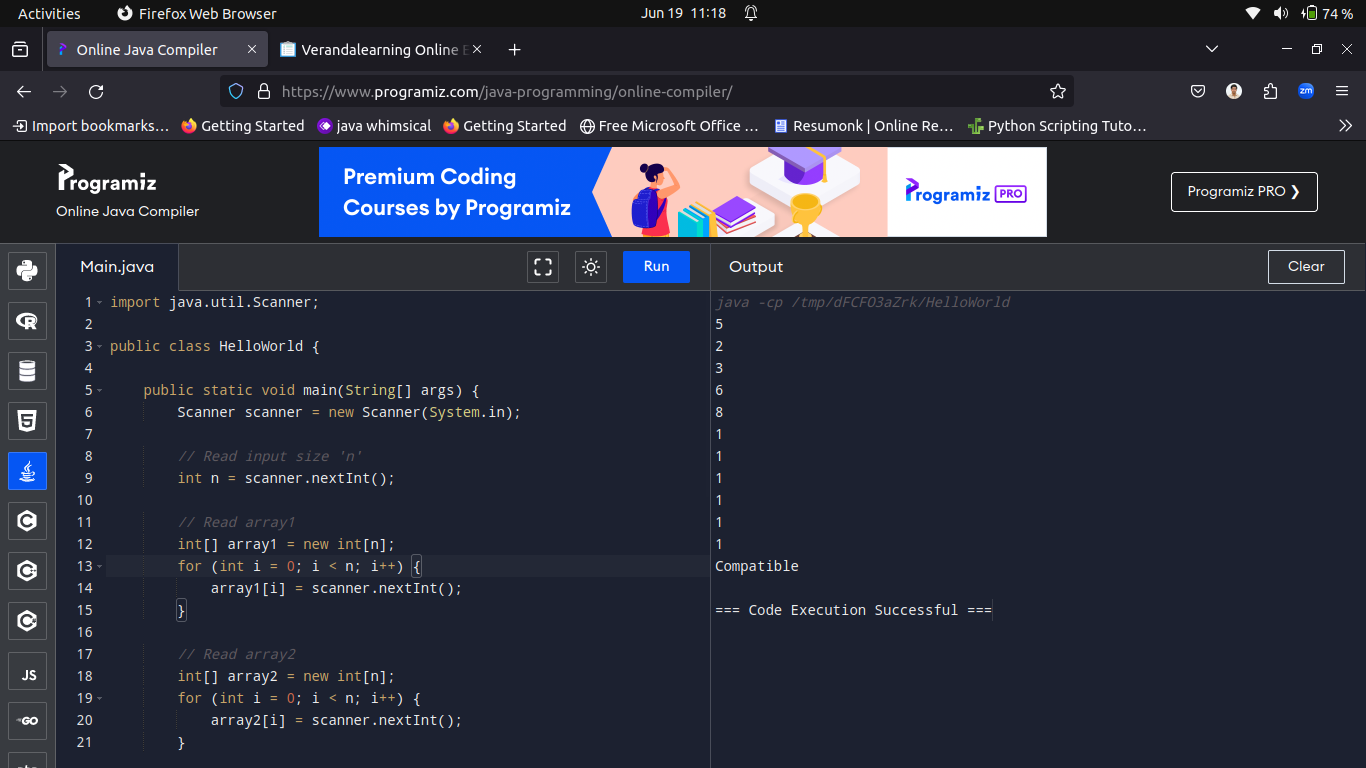
}

}

return true;

}

}



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

Q4.Kuty and Pappu are playing a coding game. Kuty gives a set of numbers to Pappu to find the even numbers in the set and sum them. Write a program to help Pappu to solve the game and to win the dairy milk.

Input Format:

Input consists of n+1 integers.

The first integer corresponds to ‘n’, the size of the array.

The next ‘n’ integers correspond to the elements in the array.

Assume that the maximum value of n is 15.

Output Format:

Refer to sample output for details.

Sample Input:

5

2

3

6

8

-1

Sample Output:

The sum of the even numbers in the array is 16

Case 1

Input (stdin)

3

4

5

6

Output (stdout)

The sum of the even numbers in the array is 10

Solution:

import java.util.Scanner;

public class HelloWorld {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

int[] arr = new int[n];

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

arr[i] = sc.nextInt();

}

int sumEven = 0;

for (int num : arr) {

if (num % 2 == 0) {

sumEven += num;

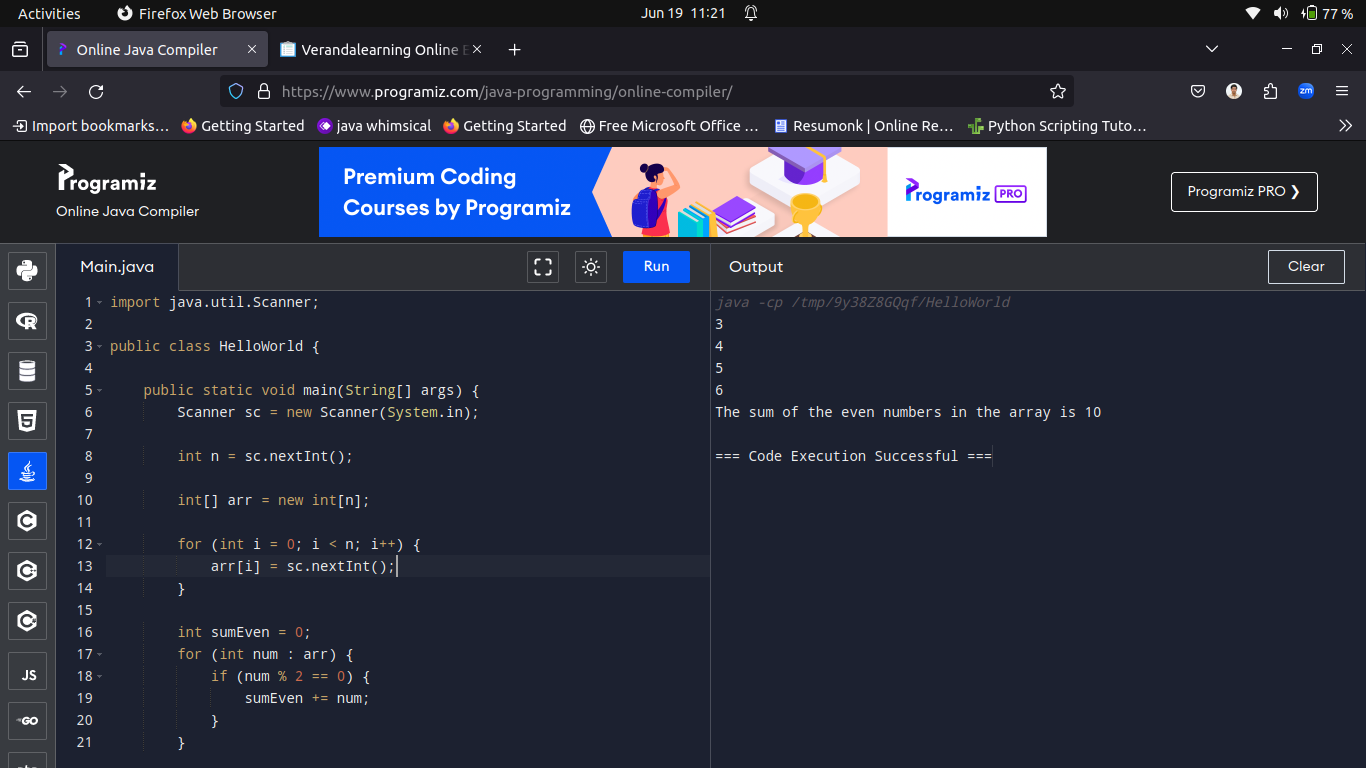
}

}

System.out.println("The sum of the even numbers in the array is " + sumEven);

}

}



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

Q5.Array Insertion

Arun and Ram were playing cards. Arun has 5 cards. Ram wants to insert a new card in between Arun's cards. Ram wants to find the position to insert the card. So help him to find the position to insert a certain card. If Ram inserts a card in a position other than the position of 5 cards, the game will be over. So play carefully. EXAMPLE: consider an array having three elements in it initially and a[0] = 1, a[1] = 2 and a[2] = 3 and you want to insert a number 45 at location 1 i.e. a[0] = 45, so we have to move elements one step below after insertion a[1] = 1 , and a[2] = 2 and a[3] = 3.

Input Format:

Assume that the maximum number of elements in the array is 20.

Output Format:

Refer to the sample input and output for formatting specifications.

Sample Input:

5

1

2

3

4

5

4

10

Sample Output:

Array after insertion is

1

2

3

10

4

5

Case 1

Input (stdin)

4

1

4

7

5

3

10

Output (stdout)

1

4

10

7

5

Solution:

import java.util.Scanner;

public class CardInsertion {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int n = scanner.nextInt();

int[] cards = new int[n+1];

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

cards[i] = scanner.nextInt();

}

// Determine the insertion point

System.out.println("Insertion point");

int insertionIndex = scanner.nextInt();

// card to be inserted

System.out.println("Card to be inserted");

int newCard = scanner.nextInt();

// Shift elements to the right from insertionIndex onwards

for (int i = n-1; i >= insertionIndex; i--) {

cards[i + 1] = cards[i];

}

// Insert the new card at insertionIndex

cards[insertionIndex] = newCard;

// Output

System.out.println("Array after insertion is");

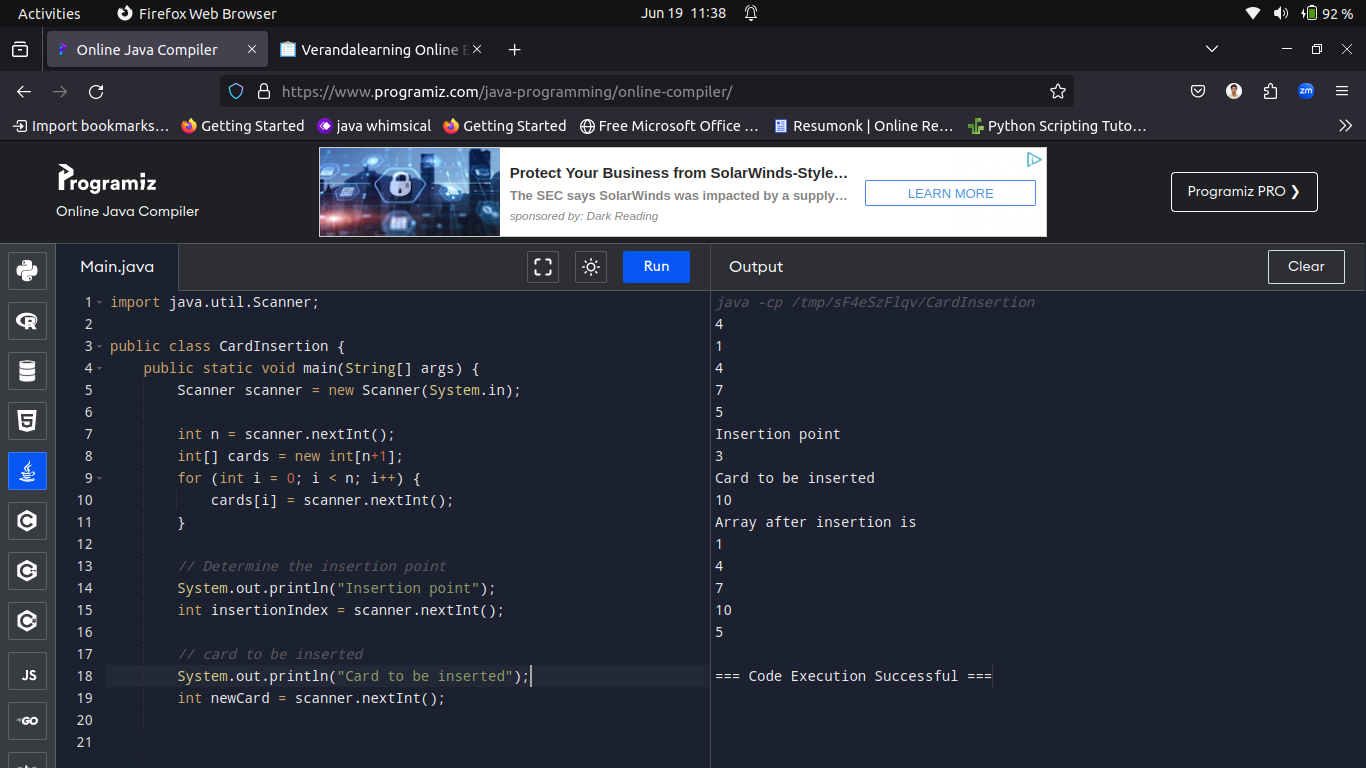
for (int i = 0; i < cards.length; i++) {

System.out.println(cards[i]);

}

}

}



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

Q6.Letter Frequency

Write a program to compute the frequency of each lowercase letter in the string.

Input Format:

Input consists of a string. Assume that all characters in the string are lowercase letters and the maximum length of the string is 200.

Output Format:

The letters are displayed sorted in ascending order.

Sample Input:

anitha

Sample Output:

a 2

h 1

i 1

n 1

t 1

Case 1

Input (stdin)

google

Output (stdout)

e1

g2

l1

o2

Solution:

import java.util.Scanner;

public class LetterFrequency {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

scanner.close();

// 26 letters in English alphabet

int[] freq = new int[26];

// Count frequency of each letter in the string

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

char c = inputString.charAt(i);

if (c >= 'a' && c <= 'z') {

freq[c - 'a']++;

}

}

// Print the sorted frequencies

for (int i = 0; i < 26; i++) {

if (freq[i] > 0) {

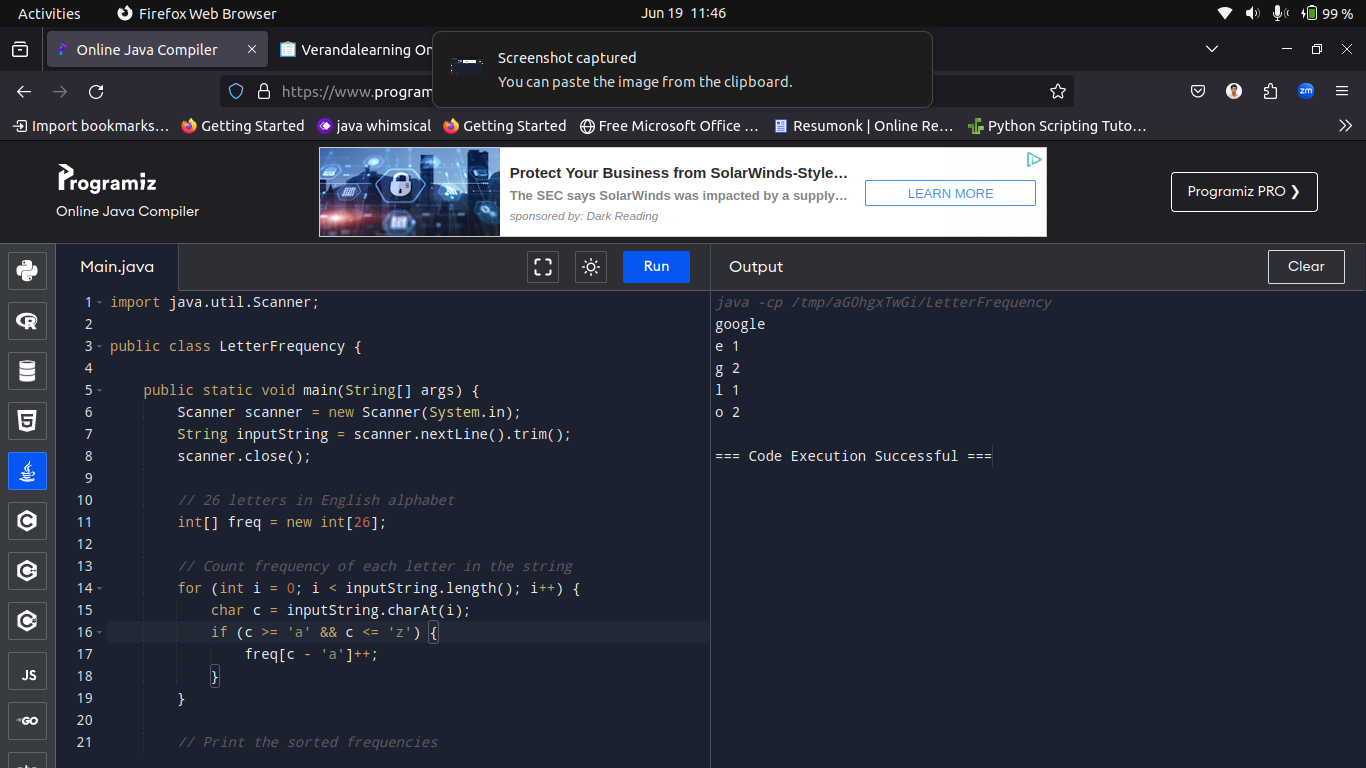
System.out.println((char) (i + 'a') + " " + freq[i]);

}

}

}

}



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

Q7.As part of a text editor application development team, you need to implement a feature that counts the number of words in a sentence. This will help in providing word count statistics to the users.

Problem Statement:

Write a Java program to count the number of words in a given sentence.

Test Cases:

Input: "The quick brown fox jumps over the lazy dog"

Output: 9

Input: "Hello World"

Output: 2

Solution:

import java.util.Scanner;

public class WordFrequency {

public static void main(String args[]){

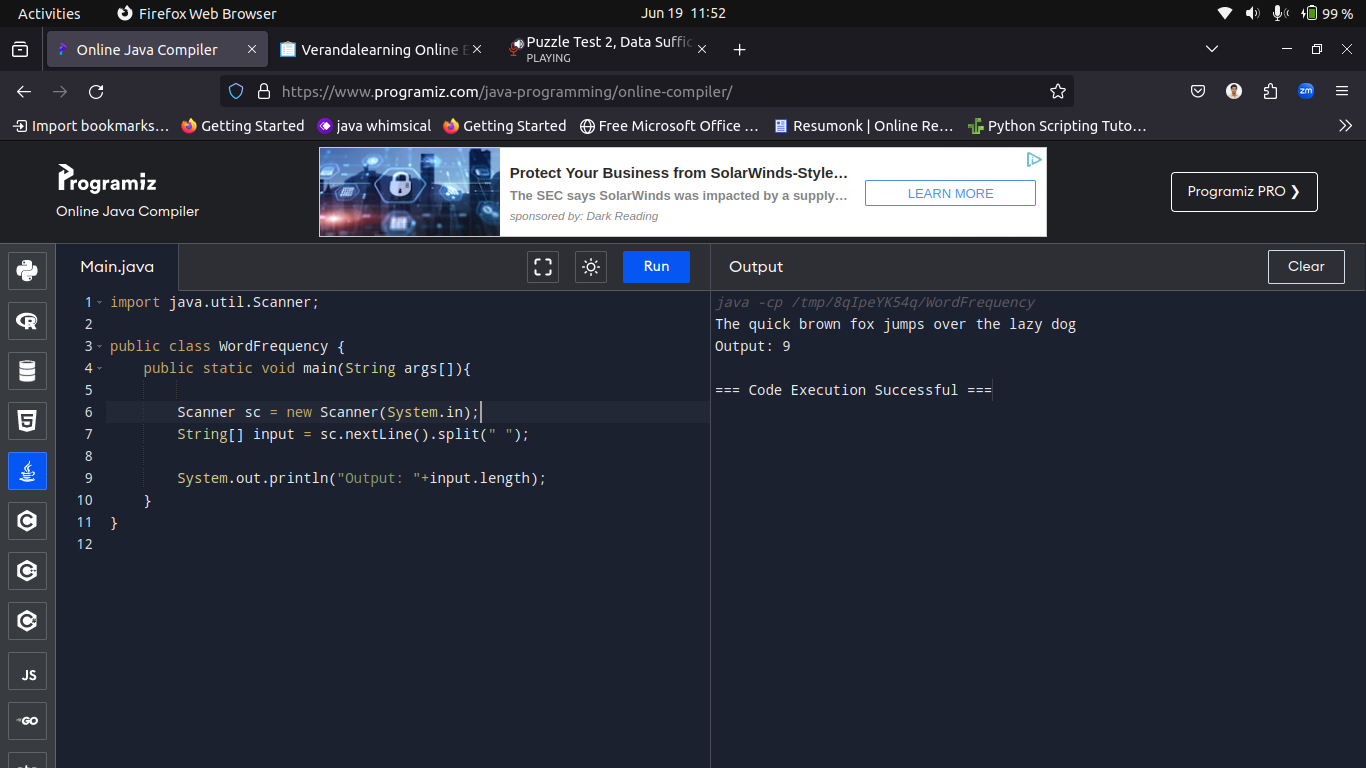
Scanner sc = new Scanner(System.in);

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

System.out.println("Output: "+input.length);

}

}



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

Q8.You are working on a billing application for a supermarket. You need to implement a feature that calculates the total bill amount after

applying discounts on certain items.

Problem Statement:

Write a Java program to calculate the total bill amount after applying discounts on items in a shopping cart. You are given an array prices

representing the prices of items in the cart and an array discounts representing the discount percentage on each item. Calculate the total bill

amount after applying discounts.

Test Cases:

Test Case 1:

Input:

prices: [100, 200, 300]

discounts: [10, 20, 30]

Output:

Total bill amount after discounts: 450.0

Test Case 2:

Input:

prices: [50, 75, 100, 125]

discounts: [5, 10, 15, 20]

Output:

Total bill amount after discounts: 312.5

Code solution:

import java.util.\*;

public class BillingApplication {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Number of Items");

int n = sc.nextInt();

double[] prices = new double[n];

int[] discounts = new int[n];

System.out.println("Enter prices");

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

prices[i] = sc.nextInt();

}

System.out.println("Enter discounts");

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

discounts[i] = sc.nextInt();

}

double totalBill1 = calculateTotalBill(prices, discounts);

System.out.println("Total bill amount after discounts: " + totalBill1);

}

public static double calculateTotalBill(double[] prices, int[] discounts) {

double totalBill = 0;

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

double price = prices[i];

int discount = discounts[i];

double discountedPrice = price - (price \* discount / 100.0);

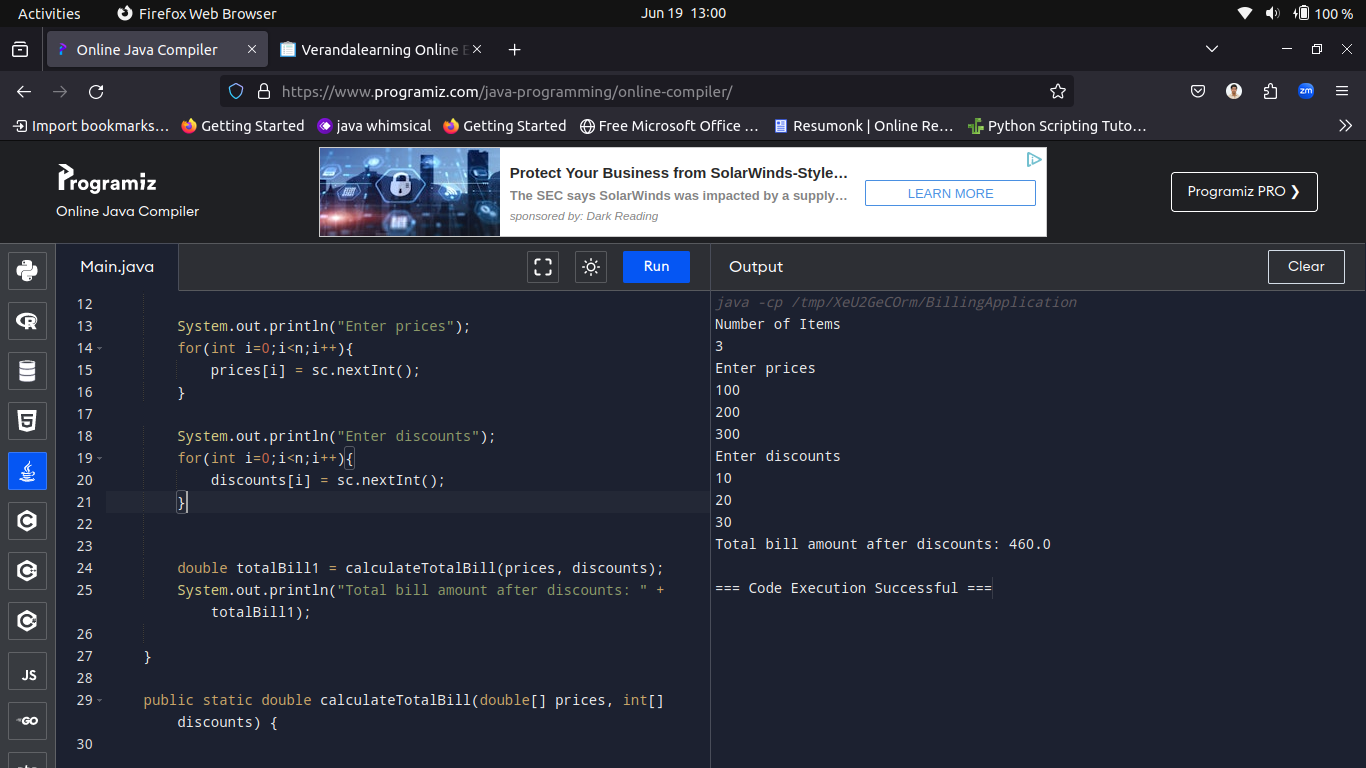
totalBill += discountedPrice;

}

return totalBill;

}

}



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

Q9.WAP a Program for Sorting first half in Ascending order and second half in Descending order

Input:

6

5 4 3 2 1 0

Output:

3, 4, 5, 2, 1, 0

code solution:

import java.util.\*;

class HelloWorld {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

sc.nextLine();

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

int[] arr = new int[n];

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

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

}

int mid = n/2;

Arrays.sort(arr,0,mid);

// System.out.println(Arrays.toString(arr));

Arrays.sort(arr, mid, n);

// System.out.println(Arrays.toString(arr));

while(mid<n){

int temp = arr[mid];

arr[mid] = arr[n-1];

arr[n-1] = temp;

mid++;

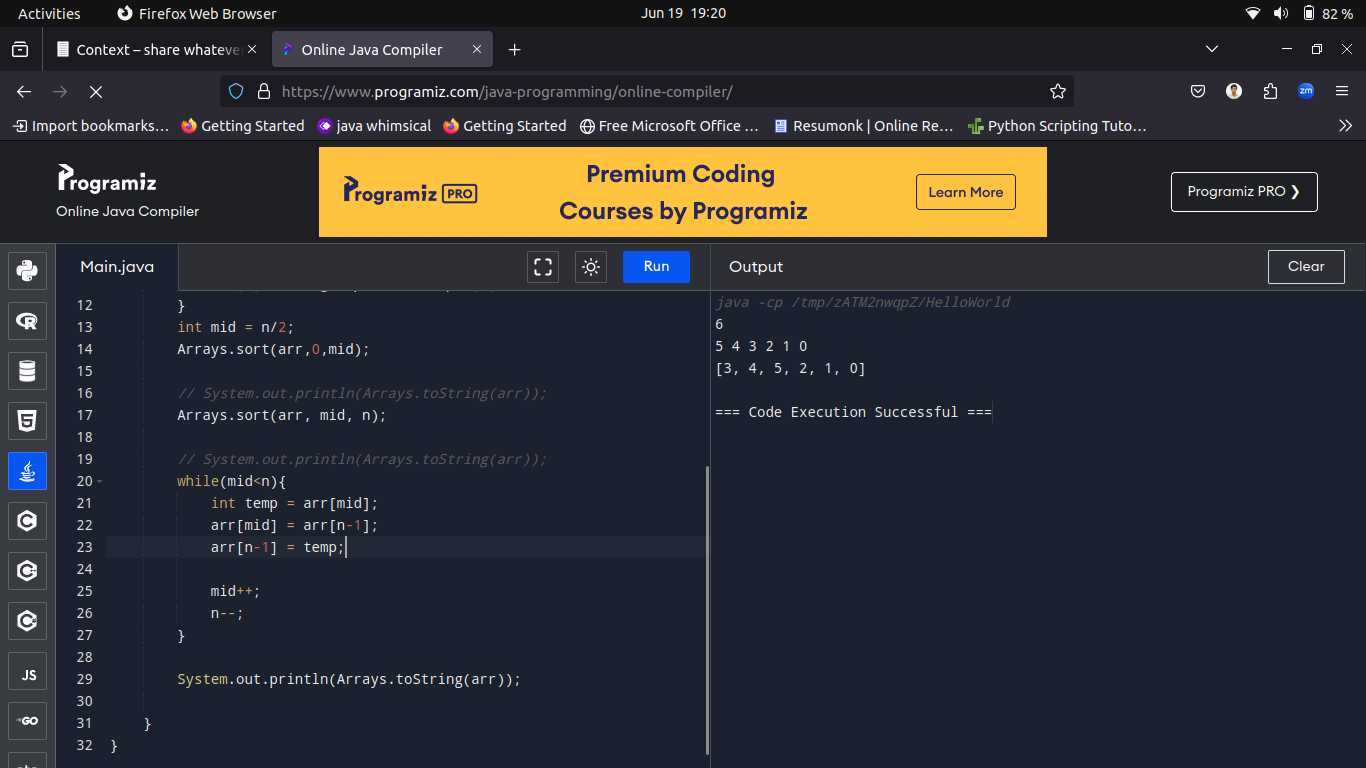
n--;

}

System.out.println(Arrays.toString(arr));

}

}



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

Q 10.Write a Java program for finding repeating element in an array.

input:

7

1 2 3 7 4 3 7

Output:

Repeating elements:

3

7

Solution:

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int n = scanner.nextInt();

int[] arr = new int[n];

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

arr[i] = scanner.nextInt();

}

Map<Integer, Integer> countMap = new HashMap<>();

for (int num : arr) {

if (countMap.containsKey(num)) {

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

} else {

countMap.put(num, 1);

}

}

System.out.println("Repeating elements:");

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

if (entry.getValue() > 1) {

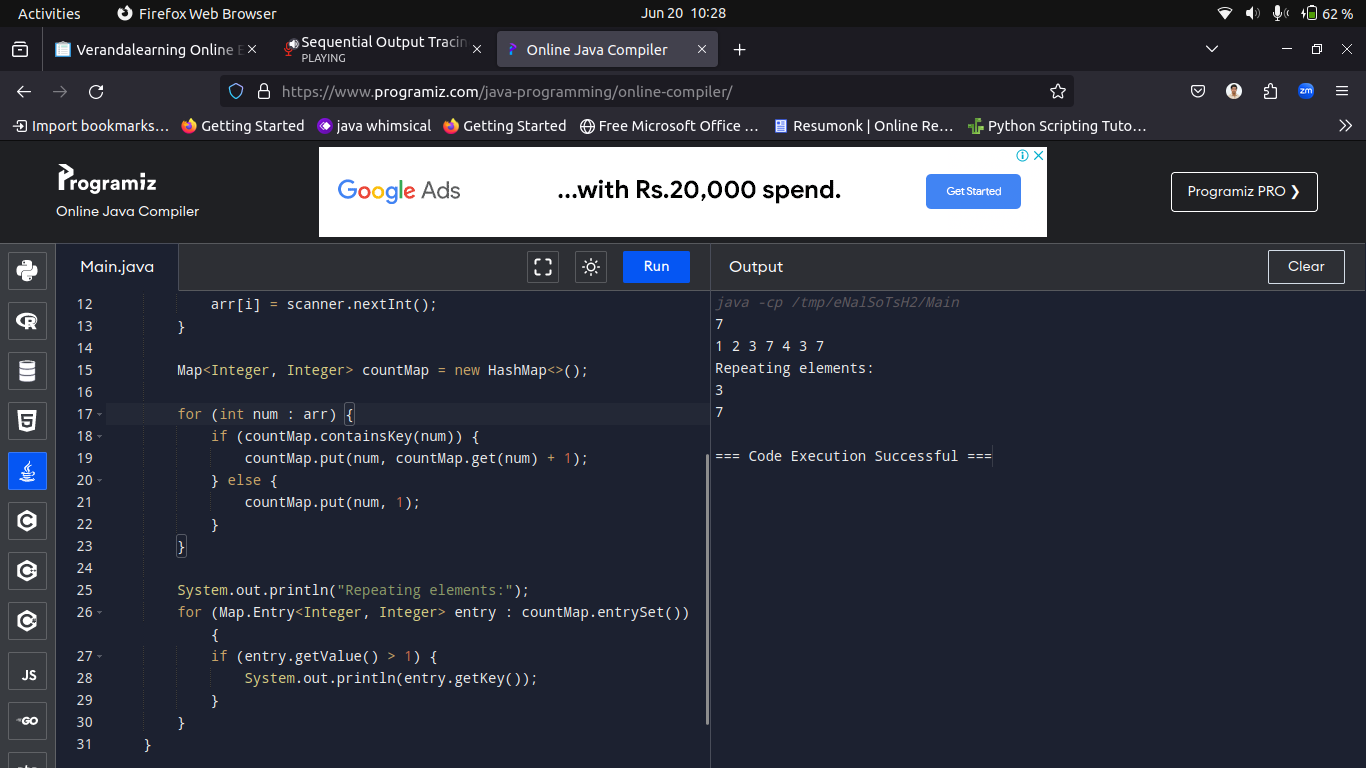
System.out.println(entry.getKey());

}

}

}

}



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