**// ARRAY BASED TASK //**

Q1. Given an array of integers (in a series) and we have to find its

missing elements (there will be a missing element)

Input/Output:

Input array: 1, 2, 3, 4, 6, 7

Output:

Missing element is: 5

// SOURCE CODE

import java.util.Scanner;

public class MissingElementFinder {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int[] array = {1, 2, 3, 4, 6, 7};

int n = array.length + 1;

int expectedSum = n \* (n + 1) / 2;

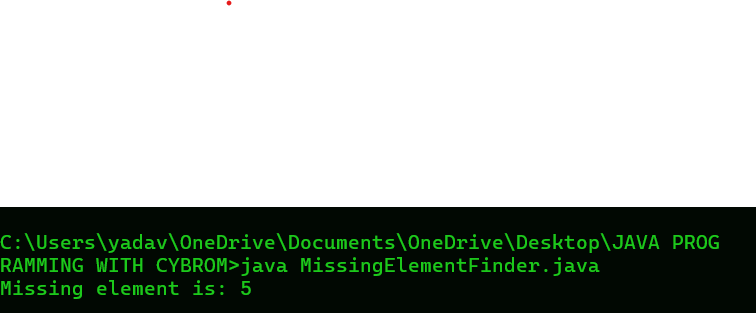
int actualSum = 0;

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

actualSum += array[i];

}

System.out.println("Missing element is: " + (expectedSum - actualSum));

// OUTPUT 

Q2.Given an array of integers and we have to find their average .

Input/Output:

Enter number of elements you want in array : 10

Enter all the elements :

65

45

25

65

84

74

74

15

36

Sum of the array elements is : 579

Average of the array elements is : 57.9

// SOURCE CODE

import java.util.Scanner;

public class AverageArray {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter number of elements you want in array: ");

int n = sc.nextInt();

int[] array = new int[n];

System.out.println("Enter all the elements:");

int sum = 0;

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

array[i] = sc.nextInt();

sum += array[i];

}

double average = (double) sum / n;

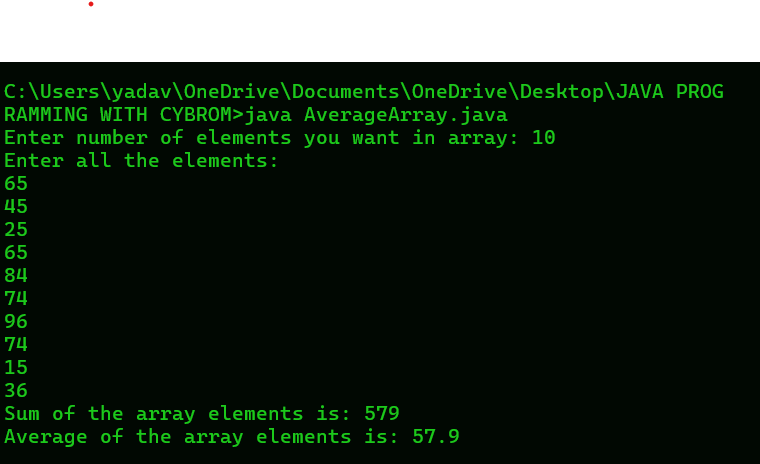
System.out.println("Sum of the array elements is: " + sum);

System.out.println("Average of the array elements is: " + average);

}

}

// OUTPUT



Q3. Move all zero at the end of the array

Given an integer array with zeros (0’s) and we have to move all zeros at

the end of the array

Input/Output:

Input array: 5, 1, 6, 0, 0, 3, 9, 0, 6, 7, 8, 12, 10, 0, 2

After moving 0 at the end

Output array: 5, 1, 6, 3, 9, 6, 7, 8, 12, 10, 2, 0, 0, 0, 0

// SOURCE CODE

public class MoveZeroes {

public static void main(String[] args) {

int[] array = {5, 1, 6, 0, 0, 3, 9, 0, 6, 7, 8, 12, 10, 0, 2};

int n = array.length;

int count = 0;

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

if (array[i] != 0) {

array[count++] = array[i];

}

}

while (count < n) {

array[count++] = 0;

}

System.out.print("After moving 0 at the end\nOutput array: ");

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

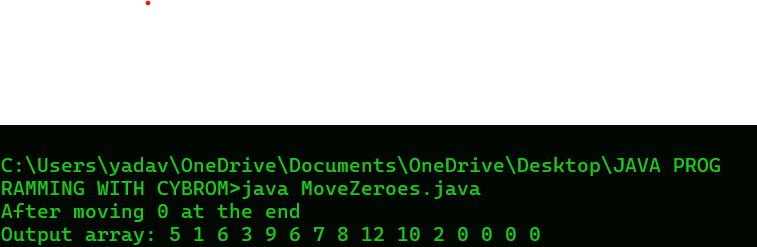
System.out.print(array[i] + " ");

}

}

}

// OUTPUT:



Q4. Delete a specific element from a one dimensional array

Given an array and an element to delete and we have to delete it from

array

Input/Output:

Input:

Given array (elements will be read in program): 10 20 30 40 50

Enter element to delete: 40

Output:

Array elements after deleting the element: 10 20 30 50

// SOURCE CODE

import java.util.Scanner;

public class DeleteElementArray {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int[] arr = {10, 20, 30, 40, 50};

int size = arr.length;

System.out.print("Enter element to delete: ");

int element = sc.nextInt();

int[] newArr = new int[size - 1];

int index = 0;

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

if (arr[i] != element) {

newArr[index++] = arr[i];

}

}

System.out.print("Array elements after deleting the element: ");

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

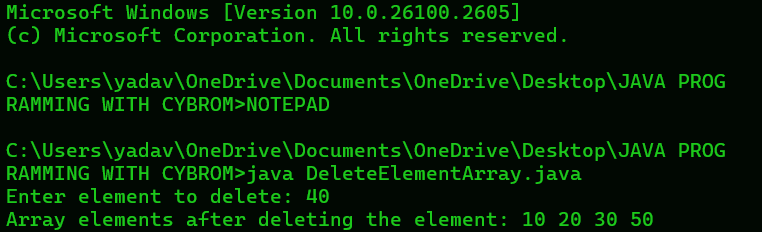
System.out.print(newArr[i] + " ");

}

}

}

// OUTPUT:



Q5. Print EVEN and ODD elements from an array

Given a one dimensional array and we have to print its EVEN and ODD

elements separately.

Input/Output:

Input:

Given array (elements will be read in program): 10 11 12 13 14

Output:

Odd numbers in the array are : 10 12 14

Even numbers in the array are : 11 13

// SOURCE CODE

public class EvenOddArray {

public static void main(String[] args) {

int[] arr = {10, 11, 12, 13, 14};

System.out.print("Even numbers in the array are: ");

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

if (arr[i] % 2 == 0) {

System.out.print(arr[i] + " ");

}

}

System.out.println();

System.out.print("Odd numbers in the array are: ");

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

if (arr[i] % 2 != 0) {

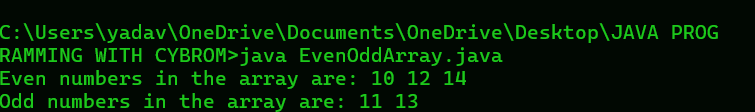
System.out.print(arr[i] + " ");

}

}

}

}



Q6. Merge two one dimensional arrays

Given two one-dimensional arrays and we have to merge them using

java program.

Input/Output:

Input:

Array 1 (elements will be read in program): 1 2 3 4 5 6 7 8 9 10

Array 2 (elements will be read in program): 11 12 13 14 15

Output:

New array (After merging elements)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

// SOURCE CODE

public class MergeArrays {

public static void main(String[] args) {

int[] array1 = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

int[] array2 = {11, 12, 13, 14, 15};

int length1 = array1.length;

int length2 = array2.length;

int[] mergedArray = new int[length1 + length2];

for (int i = 0; i < length1; i++) {

mergedArray[i] = array1[i];

}

for (int i = 0; i < length2; i++) {

mergedArray[length1 + i] = array2[i];

}

System.out.print("New array (After merging elements): ");

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

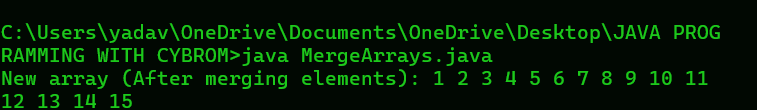
System.out.print(mergedArray[i] + " ");

}

}

}

// OUTPUT:



Q7. Read and print a two dimensional array

Read number of rows and columns, array elements for two dimensional

array and print in matrix format using java program.

Input/Output:

Enter number of rows: 3

Enter number of columns: 3

Enter elements

1

2

3

4

5

6

7

8

9

Output:

Matrix is:

1 2 3

4 5 6

7 8 9

// SOURCE CODE:

import java.util.Scanner;

public class TwoDArray {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter number of rows: ");

int rows = sc.nextInt();

System.out.print("Enter number of columns: ");

int cols = sc.nextInt();

int[][] matrix = new int[rows][cols];

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

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

for (int j = 0; j < cols; j++) {

matrix[i][j] = sc.nextInt();

}

}

System.out.println("Matrix is:");

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

for (int j = 0; j < cols; j++) {

System.out.print(matrix[i][j] + " ");

}

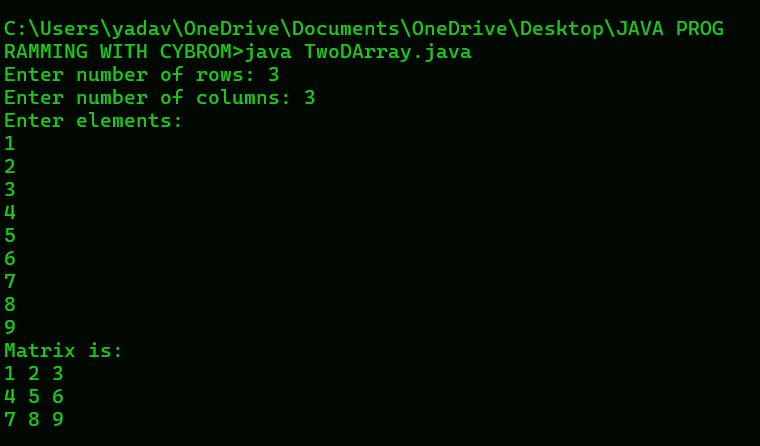
System.out.println();

}

}

}

// OUTPUT:



Q8. Count strings and integers from an array

Given an array with strings and integers and we have to count strings

and integers using java program.

Input/Output:

Input:

Array = {&quot;Raj&quot;, &quot;77&quot;, &quot;101&quot;, &quot;99&quot;, &quot;Jio&quot;}

Output:

Numeric:3

Strings:2

//SOURCE CODE

public class CountStringsIntegers {

public static void main(String[] args) {

String[] array = {"Raj", "77", "101", "99", "Jio"};

int numericCount = 0;

int stringCount = 0;

for (String element : array) {

try {

Integer.parseInt(element);

numericCount++;

} catch (NumberFormatException e) {

stringCount++;

}

}

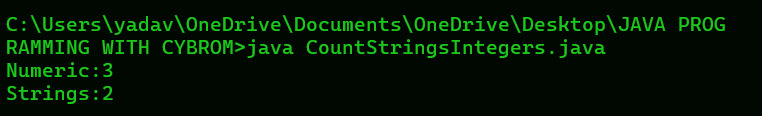
System.out.println("Numeric:" + numericCount);

System.out.println("Strings:" + stringCount);

}

}

// OUTPUT:



Q9. Remove duplicate elements from an array

Given an array of integers and we have to remove duplicate elements

using java program.

Input/Output:

Input array elements:

1, 2, 3, 1, 2, 3, 4

Output:

Elements after removing duplicates

1, 2, 3, 4

// SOURCE CODE

import java.util.LinkedHashSet;

public class RemoveDuplicates {

public static void main(String[] args) {

int[] array = {1, 2, 3, 1, 2, 3, 4};

LinkedHashSet<Integer> uniqueElements = new LinkedHashSet<>();

for (int element : array) {

uniqueElements.add(element);

}

System.out.print("Elements after removing duplicates: ");

for (int element : uniqueElements) {

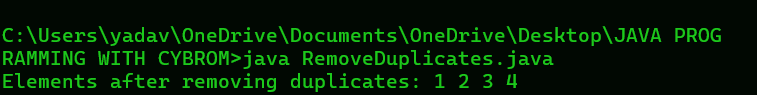
System.out.print(element + " ");

}

}

}

// OUTPUT:



Q10. Find second largest element in an array

Given an array of N integers and we have to find its second largest

element using Java program.

Input/Output:

Input:

Enter number of elements: 4

Input elements: 45, 25, 69, 40

Output:

Second largest element in: 45

// SOURCE CODE

import java.util.Scanner;

public class SecondLargestElement {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter number of elements: ");

int n = sc.nextInt();

int[] array = new int[n];

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

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

array[i] = sc.nextInt();

}

int largest = Integer.MIN\_VALUE;

int secondLargest = Integer.MIN\_VALUE;

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

if (array[i] > largest) {

secondLargest = largest;

largest = array[i];

} else if (array[i] > secondLargest && array[i] != largest) {

secondLargest = array[i];

}

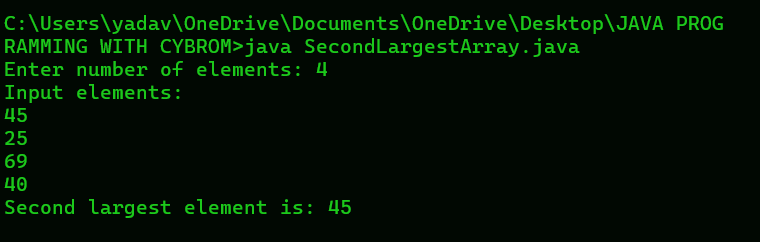
}

System.out.println("Second largest element is: " + secondLargest);

}

}

// OUTPUT:



Q11. Find second smallest element in an array

Given an array of N integers and we have to find its second

minimum/smallest element using Java program.

Input/Output:

Input:

Enter number of elements: 4

Input elements: 45, 25, 69, 40

// SOURCE CODE

import java.util.Scanner;

public class SecondSmallestArray {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter number of elements: ");

int n = sc.nextInt();

int[] array = new int[n];

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

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

array[i] = sc.nextInt();

}

int largest = Integer.MIN\_VALUE;

int secondLargest = Integer.MIN\_VALUE;

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

if (array[i] > largest) {

secondLargest = largest;

largest = array[i];

} else if (array[i] > secondLargest && array[i] != largest) {

secondLargest = array[i];

}

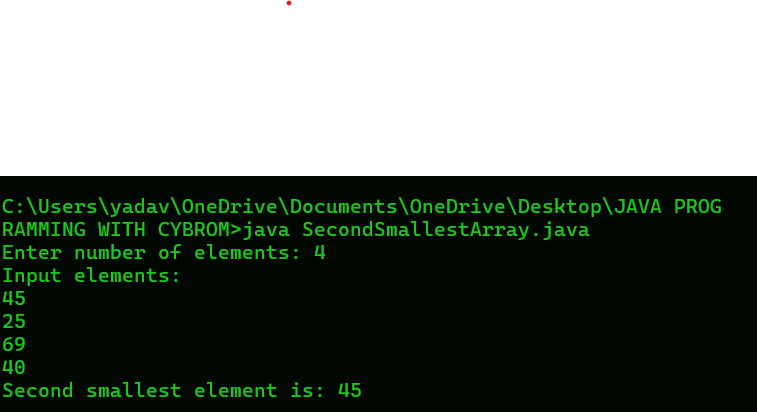
}

System.out.println("Second smallest element is: " + secondLargest);

}

}

// OUTPUT



Q12. Find smallest element in an array

Given an array of N integers and we have to find its minimum/smallest

element using Java program.

Input/Output:

Input:

Enter number of elements: 4

Input elements: 45, 25, 69, 40

Output:

Smallest element in: 25

// SOURCE CODE

import java.util.Scanner;

public class SmallestElement {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter number of elements: ");

int n = input.nextInt();

int[] array = new int[n];

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

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

array[i] = input.nextInt();

}

int smallest = array[0];

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

if (array[i] < smallest) {

smallest = array[i];

}

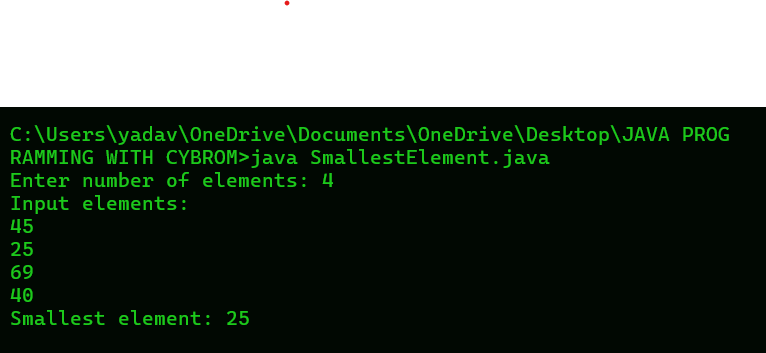
}

System.out.println("Smallest element: " + smallest);

}

}

// OUTPUT:



Q13. Count total positives, negatives and zeros from an array

Given an array of integers and we have to count total negatives,

positives and zeros using java program.

Input/Output:

Input:

Array elements: 20, -10, 15, 00, -85

Output:

Positive Numbers are: 2

Negative Numbers are: 2

Zeros are: 1

// SOURCE CODE

import java.util.Scanner;

public class CountNumbers {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter number of elements: ");

int n = input.nextInt();

int[] array = new int[n];

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

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

array[i] = input.nextInt();

}

int positiveCount = 0, negativeCount = 0, zeroCount = 0;

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

if (array[i] > 0) {

positiveCount++;

} else if (array[i] < 0) {

negativeCount++;

} else {

zeroCount++;

}

}

System.out.println("Positive Numbers are: " + positiveCount);

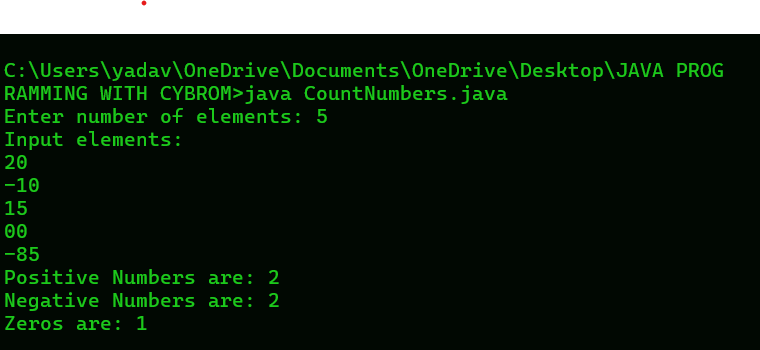
System.out.println("Negative Numbers are: " + negativeCount);

System.out.println("Zeros are: " + zeroCount);

}

}

// OUTPUT:



Q14. Access elements of character array using for each loop

In this program, we will create an array of characters and access its

elements using for each loop and print it on the screen.

// SOURCE CODE

public class CharArrayAccess {

public static void main(String[] args) {

char[] charArray = {'A', 'B', 'C', 'D', 'E', 'F'};

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

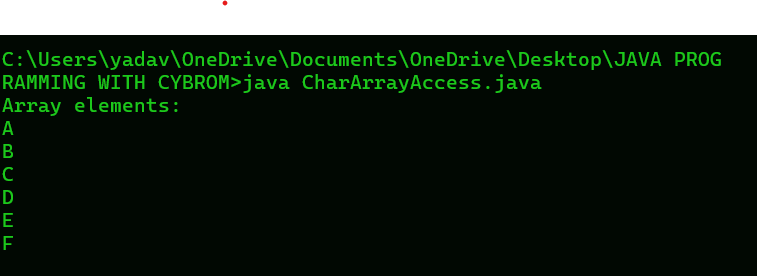
for (char ch : charArray) {

System.out.println(ch);

}

}

}



Q15. Find prime and non-prime numbers in the array

Given/input an integer array, we have to find prime and non-prime

numbers in the array.

Input/Output:

Input:

int intArr[] = {10, 11, 13, 15, 17, 19, 23, 25, 30};

Output:

10 - Not Prime

11 - Prime

13 - Prime

15 - Not Prime

17 - Prime

19 - Prime

23 - Prime

25 - Not Prime

30 - Not Prime

// SOURCE CODE

public class PrimeNumbersInArray {

public static void main(String[] args) {

int[] intArr = {10, 11, 13, 15, 17, 19, 23, 25, 30};

for (int num : intArr) {

if (isPrime(num)) {

System.out.println(num + " - Prime");

} else {

System.out.println(num + " - Not Prime");

}

}

}

static boolean isPrime(int number) {

if (number <= 1) {

return false;

}

for (int i = 2; i <= Math.sqrt(number); i++) {

if (number % i == 0) {

return false;

}

}

return true;

}

}

