**Assignment – Core Java Arrays**

**1.**  Write a program to count a total number of duplicate and unique elements in the given array.

**Sample Input:**

Enter array size: 5

Array elements are: 2 5 7 2 4

**Sample Output:**

No of duplicate element: 1 No of unique elements: 3

**Ans: -**

package Arrys;

import java.util.\*;

public class Duplicate\_UniqueArray {

public static void main(String args[]) {

int[] arrElements = new int[5];

System.***out***.println("Enter 5 elements");

Scanner sc = new Scanner(System. ***in***);

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

for(int i = 0; i < 5; ++i) {

arrElements[i] = sc.nextInt();

map.merge(arrElements[i], 1 , Integer::*sum*);

}

sc.close();

long unique = map.values().stream().filter(n -> n == 1).count();

long duplicates = map.size() - unique;

System.***out***.println("Number of duplicate elements:: " + duplicates);

System.***out***.println("Number of Unique elements:: " + unique);

}

}

}

**2.**  There are N friends in a group. Each of them has Xi chocolates. Write a Java Program to check whether they can share all of these chocolates among themselves such that each one of them has equal number of chocolates.

**Input:**

            First line contains of a single line of input, an integer N denoting no. of friends in the group.

Next line contains N space separated integers Xi denoting the no. chocolates with friend has.

**Output:**

        Output "Yes" if it is possible to share equally else "No" (without " ").

**Sample Input:**

3

1 2 3

**Sample Output:**

Yes

**Ans:-**

**package Array;**

**import java.util.Scanner;**

**public class ShareChocolates {**

**public static void main(String[] args) {**

**Scanner scanner = new Scanner(System.in);**

**System.out.print("Enter the number of friends: ");**

**int n = scanner.nextInt();**

**int[] chocolates = new int[n];**

**System.out.println("Enter the number of chocolates each friend has:");**

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

**chocolates[i] = scanner.nextInt();**

**}**

**if (canShareEqually(chocolates)) {**

**System.out.println("Yes");**

**} else {**

**System.out.println("No");**

**}**

**}**

**public static boolean canShareEqually(int[] chocolates) {**

**int totalChocolates = 0;**

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

**totalChocolates += chocolates[i];**

**}**

**if (totalChocolates % chocolates.length == 0) {**

**return true;**

**} else {**

**return false;**

**}**

**}**

**}**

**3.**  Consider an array contains both positive and negative numbers in random order. Write a Java Program to rearrange the array elements so that all negative numbers appear before all positive numbers.

**Input:**

First line contains of a single line of input, an integer N denoting the size of an array.

Next line contains N space separated integers (positive or negative)

**Output:**

Set of negative integers followed by positive integers

**Sample Input:**

9

-12 11 -13 -5 6 -7 5 -3 -6

**Sample Output:**

-12 -13 -5 -7 -3 -6 11 6 5

**Ans:-**

**package Array;**

**import java.util.Scanner;**

**public class ArrayRearrange {**

**public static void main(String[] args) {**

**Scanner scanner = new Scanner(System.in);**

**System.out.print("Enter the size of the array: ");**

**int size = scanner.nextInt();**

**int[] arr = new int[size];**

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

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

**arr[i] = scanner.nextInt();**

**}**

**rearrangeArray(arr);**

**System.out.println("Rearranged array:");**

**for (int num : arr) {**

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

**}**

**}**

**private static void rearrangeArray(int[] arr) {**

**int n = arr.length;**

**int[] temp = new int[n];**

**int start = 0;**

**int end = n - 1;**

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

**if (arr[i] < 0) {**

**temp[start] = arr[i];**

**start++;**

**} else {**

**temp[end] = arr[i];**

**end--;**

**}**

**}**

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

**arr[i] = temp[i];**

**}**

**}**

**}**

**4.**  Arun and Naveen are playing a dice game. Each one will get one chance to roll the dice. Dice values are recorded in two different tables. After 10 turns, the winner has to be decided by calculating each value recorded in each table.

**Sample Input:**

            Arun Dice Choice: 5, 6, 2, 1, 2, 5, 6, 3, 4, 2

Naveen Choice: 5, 5, 4, 3, 3, 5, 6, 2, 1, 1

**Sample Output:**

Naveen Wins!!!

Ans:-

package Array;

import java.util.Arrays;

public class DiceGame {

public static void main(String[] args) {

int[] arunChoices = {5, 6, 2, 1, 2, 5, 6, 3, 4, 2};

int[] naveenChoices = {5, 5, 4, 3, 3, 5, 6, 2, 1, 1};

int arunScore = calculateScore(arunChoices);

int naveenScore = calculateScore(naveenChoices);

if (arunScore > naveenScore) {

System.out.println("Arun Wins!!!");

} else if (naveenScore > arunScore) {

System.out.println("Naveen Wins!!!");

} else {

System.out.println("It's a Tie!");

}

}

private static int calculateScore(int[] choices) {

int score = 0;

for (int choice : choices) {

score += choice;

}

return score;

}

}

**5.**  Write a program to find the minimum and maximum element of each row and column in the given two-dimensional arrays.

**Sample Input:**

Enter row size: 3 Enter column size: 3

Enter 3 \* 3 array elements are: 4

1

2

5

3

6

3

7

8

[Text Wrapping Break]

**Sample Output:**

**Given Array is:**

4  1  2

5  3  6

3  7   8

The minimum and maximum element of 1st row is: 1, 4

The minimum and maximum element of 2nd row is: 3, 6

The minimum and maximum element of 3rd row is: 6, 8

The minimum and maximum element of 1st column is: 3, 5

The minimum and maximum element of 2nd column is: 1, 7

The minimum and maximum element of 3rd column is: 2, 8

Ans:-

package Array;

import java.util.Scanner;

public class MinMaxArray {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter row size: ");

int rows = scanner.nextInt();

System.out.print("Enter column size: ");

int columns = scanner.nextInt();

int[][] array = new int[rows][columns];

System.out.println("Enter " + rows + " \* " + columns + " array elements:");

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

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

array[i][j] = scanner.nextInt();

}

}

System.out.println("\nGiven Array is:\n");

printArray(array);

findRowMinMax(array);

findColumnMinMax(array);

}

private static void printArray(int[][] array) {

for (int[] row : array) {

for (int element : row) {

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

}

System.out.println();

}

System.out.println();

}

private static void findRowMinMax(int[][] array) {

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

int min = array[i][0];

int max = array[i][0];

for (int j = 1; j < array[i].length; j++) {

if (array[i][j] < min) {

min = array[i][j];

}

if (array[i][j] > max) {

max = array[i][j];

}

}

System.out.println("The minimum and maximum element of " + (i + 1) + "st row is: " + min + ", " + max);

}

System.out.println();

}

private static void findColumnMinMax(int[][] array) {

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

int min = array[0][i];

int max = array[0][i];

for (int j = 1; j < array.length; j++) {

if (array[j][i] < min) {

min = array[j][i];

}

if (array[j][i] > max) {

max = array[j][i];

}

}

System.out.println("The minimum and maximum element of " + (i + 1) + "st column is: " + min + ", " + max);

}

}

}