**Array:**

**Level 1:**

**Easy:**

**1. Write a program to add even and odd numbers from 1 to 10. Store and display their results in two separate arrays.**

class EvenOddArray {

    public static void main(String[] args) {

        int[] even = new int[5];

        int[] odd = new int[5];

        int e = 0, o = 0;

        for (int i = 1; i <= 10; i++)

{

            if (i % 2 == 0) even[e++] = i;

            else odd[o++] = i;

        }

        System.out.println("Even Numbers: " + Arrays.toString(even));

        System.out.println("Odd Numbers: " + Arrays.toString(odd));

    }

}

Output:

Even Numbers: [2, 4, 6, 8, 10]

Odd Numbers: [1, 3, 5, 7, 9]

**2. Write a program to sort the elements of an array in sequence.**

class SortArray {

    public static void main(String[] args) {

        int[] arr = {5, 2, 9, 1, 6, 8, 3};

        Arrays.sort(arr);

        System.out.println("Sorted Array: " + Arrays.toString(arr));

    }

}

Output:

Sorted Array: [1, 2, 3, 5, 6, 8, 9]

**3. Find max or min value in an array of primitives using C#.**

class MaxMinArray {

    public static void main(String[] args) {

        int[] arr = {5, 12, 8, 3, 7, 9};

        int max = arr[0], min = arr[0];

        for (int num : arr) {

            if (num > max)

             max = num;

            if (num < min)

min = num;

        }

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

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

    }

}

Output:

Max: 12

Min: 3

**4. Write a program to find the frequency of each element in the array.**

class FrequencyArray {

    public static void main(String[] args) {

        int[] arr = {2, 3, 2, 4, 5, 3, 4, 4};

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

        for (int num : arr) freq.put(num, freq.getOrDefault(num, 0) + 1);

        System.out.println("Element Frequencies: " + freq);

    }

}

Output:

Element Frequencies: {2=2, 3=2, 4=3, 5=1}

**5. Identify the greatest element in an n-dimensional array.**

class GreatestElement {

    public static void main(String[] args) {

        int[][] arr = {{3, 8, 12}, {5, 9, 7}, {15, 6, 4}};

        int max = arr[0][0];

        for (int[] row : arr)

            for (int num : row)

                if (num > max) max = num;

        System.out.println("Greatest Element: " + max);

    }

}

Output:

Greatest Element: 15

**6. Write a program to find the duplicate number in a given integer array.**

public class DuplicateNumber {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        int n = sc.nextInt();

        int[] arr = new int[n];

        System.out.println("Enter " + n + " numbers:");

        for (int i = 0; i < n; i++) {

            arr[i] = sc.nextInt();

        }

        System.out.print("Duplicate:");

        boolean found = false;

        for (int i = 0; i < n; i++) {

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

                if (arr[i] == arr[j]) {

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

                    found = true;

                    break;

                }

            }

        }

        if (!found) {

            System.out.print("No duplicates found.");

        }

        sc.close();

    }

}

Output:

Enter the number of elements: 6

Enter 6 numbers:

1 3 5 3 7 1

Duplicate: 3 1

**7. Write a program to merge two arrays into a third array. Display the values of three arrays.**

package ArrayPractice;

import java.util.\*;

public class MergeArray {

public static void main(String args[]) {

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

int size=sc.nextInt();

int arr1[]=new int[size];

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

arr1[i]=sc.nextInt();

}

int arr2[]=new int[size];

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

arr2[i]=sc.nextInt();

}

int arr3[]=new int[(size+size)];

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

arr3[i]=arr1[i];

}

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

arr3[i]=arr2[i-size];

}

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

System.*out*.print(arr3[i]+" ");

}

sc.close();

}

}

**8. Write a program to find the missing number in a given integer array of 1 to 100.**

public class MissingNumber {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the total number of elements (including missing one): ");

        int n = sc.nextInt();

        int[] arr = new int[n - 1];

        System.out.println("Enter " + (n - 1) + " numbers:");

        int actualSum = 0;

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

            arr[i] = sc.nextInt();

            actualSum += arr[i];

        }

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

        int missingNumber = expectedSum - actualSum;

        System.out.println("Missing Number: " + missingNumber);

        sc.close();

    }

}

Output:

Enter the total number of elements (including missing one): 5

Enter 4 numbers:

1 2 3 5

Missing Number: 4

**Arrays:**

**Level 2:**

**Medium**

1. **Write a program to check whether a specific element is present in this Array or not. Input: arr[] = [5, 1, 1, 9, 7, 2, 6, 10], key = 7 Output: true Input: arr[] = [-1, 1, 5, 8], key = -2 Output: false**

class ElementSearch {

    static boolean isElementPresent(int[] arr, int key) {

        for (int num : arr) {

            if (num == key) return true;

        }

        return false;

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int[] arr = {5, 1, 1, 9, 7, 2, 6, 10};

        System.out.print("Enter key to search: ");

        int key = sc.nextInt();

        System.out.println(isElementPresent(arr, key));

        sc.close();

    }

}

**Output:**

Enter key to search: 7

true

Enter key to search: -2

false

**2. Roll the dice 100 times and display the frequency of each number rolled using arrays.**

**package** ArrayPractice;

**import** java.util.\*;

**public** **class** DiceFrequency {

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

**int** frequency[]=**new** **int**[6];

Random random=**new** Random();

**for**(**int** i=0;i<100;i++) {

**int** roll=random.nextInt(6)+1;

frequency[roll-1]++;

}

System.***out***.println("Frequency of numbers rolled:");

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

      System.***out***.println((i + 1) + ": " + frequency[i]);

}

   }

}

Output:

Frequency of numbers rolled:

1: 16

2: 13

3: 18

4: 20

5: 18

6: 15

**3. 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**

package ArrayPractice;

public class DupUniq {

public static void main(String args[]) {

int size=5;

int count=0;

int a[]=new int[128];

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

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

a[arr[i]]++;

}

if(a[arr[i]]==0) {

count++;

System.*out*.println("No of unique elements: "+count);

}

else {

System.*out*.println("No of duplicate elements: "+count);

}

}

}

**4. Ivan has a list of n elements and he need to eliminate the duplicate numbers from the list and to create the new list to hold the unique number.**

**import** java.util.\*;

**public** **class** List {

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

**int** arr[]=**new** **int**[10];

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

**int** n=arr.length;

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

arr[i]=sc.nextInt();

}

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

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

**if**(arr[i]==arr[j]) {

arr[j]=-1;

}

}

}

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

**if**(arr[i]!=-1) {

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

}

}

sc.close();

}

}

Output:

1 2 3 4 5 5 6 7 8 6

1 2 3 4 5 6 7 8

**5. Teena having elements as 10,10,10,10,20,20,20,20,40,40,50,50,30 and she maintained in the data structure “my\_record”. She needs to find the frequency of element in my\_report and display the count.**

**import** java.util.\*;

**public** **class** Teena {

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

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

**int** n=arr.length;

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

**int** count =1;

**if**(arr[i]!=-1) {

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

**if**(arr[i]==arr[j]) {

count++;

arr[j]=-1;

}

}

System.***out***.println(arr[i]+" "+count);

  }

}

}

}

Output:

10 4

20 4

40 2

50 2

30 1

**6. Mark enters randomly twenty five numbers from the keyboard and stores it into an array. He wants to search if the number is present in the array and if it is present, he needs to display the number of times it appears in the array.**

**import** java.util.\*;

**public** **class** MarkRandom {

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

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

Random random=**new** Random();

**int** key=sc.nextInt();

**int** count=0;

**int** arr[]=**new** **int**[25];

**for**(**int** i=0;i<25;i++) {

arr[i]=random.nextInt(100)+1;

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

}

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

**for**(**int** i=0;i<25;i++) {

**if**(arr[i]==key) {

count++;

}

}

System.***out***.println("key is found "+count);

sc.close();

}

}

**Arrays:**

**Level 3**

**Hard**

**1. There are N friends in a group. Each of them has Xi chocolates. Write a 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 ith friend has. Output: Output "Yes" if it is possible to share equally else "No" (without " "). Sample Input: 3 1 2 3 Sample Output: Yes**

**package** ArrayPractice;

**import** java.util.\*;

**public** **class** Chocolate {

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

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

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

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

**int** total=0;

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

chocolate[i]=sc.nextInt();

total += chocolate[i];

}

**if**(total%n ==0) {

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

        } **else** {

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

        }

sc.close();

}

}

**2. 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!!!**

package ArrayPractice;

import java.util.\*;

public class ArunNaveen {

    public static void main(String[] args) {

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

        int[] arunDice = new int[10];

        for (int i = 0; i < 10; i++) {

            arunDice[i] = sc.nextInt();

        }

        int[] naveenDice = new int[10];

        for (int i = 0; i < 10; i++) {

            naveenDice[i] = sc.nextInt();

        }

        int arunTotal = 0;

        int naveenTotal = 0;

        for (int i = 0; i < 10; i++) {

            arunTotal += arunDice[i];

            naveenTotal += naveenDice[i];

        }

        if (arunTotal > naveenTotal) {

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

        } else if (naveenTotal > arunTotal) {

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

        } else {

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

        }

        sc.close();

    }

}

1. **Consider the program that reads integers from the standard input (until it gets a negetive number) and puts them into an array. After that it calls processArray on the array, and then prints the contents of the array on standard output. For this question, we define a black number as any number that is divisible by 7, and a white number that is divisible by 8. Currently, processArray does not modify the array. You have to change this program so that any number in the array is replaced by -2 if it is a blck number, with -9 if it is a white number, and with -6 if it is both a black and white number. To do this, you have to put your code inside processArray. Do not change anything else in the program. Sample input: 84 13 96 56 -1 Sample output: -2 13 -9 -6**

package ArrayPractice;

import java.util.\*;

public class BlackNumber {

public static void main(String args[]) {

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

int[] numbers = new int[100];

        int count = 0;

        while (true) {

            int num = sc.nextInt();

            if (num < 0) {

                break;

            }

            numbers[count++] = num;

        }

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

            int num = numbers[i];

            if (num % 7 == 0 && num % 8 == 0) {

                numbers[i] = -6;

            } else if (num % 7 == 0) {

                numbers[i] = -2;

            } else if (num % 8 == 0) {

                numbers[i] = -9;

            }

        }

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

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

        }

        sc.close();

}

}

1. **In a school there are Indian, NRI and Foreigners studying. Their heights are given in three different values. To represent their class, they need to stand in height order in a single line. Given three arrays X, Y, Z of different sizes M, N and O, write a program to merge the three arrays and make it a single sorted array. Sample Input: 5 2 7 9 15 25 4 5 8 13 20 5 2 45 3 4 3 Sample Output: 2 2 3 3 4 5 7 8 9 13 15 20 25 45**

package ArrayPractice;

import java.util.\*;

public class XYZsizes {

  public static void main(String args[]) {

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

      int M = sc.nextInt();

      int[] X = new int[M];

      for (int i = 0; i < M; i++) {

          X[i] = sc.nextInt();

      }

      int N = sc.nextInt();

      int[] Y = new int[N];

      for (int i = 0; i < N; i++) {

          Y[i] = sc.nextInt();

      }

      int O = sc.nextInt();

      int[] Z = new int[O];

      for (int i = 0; i < O; i++) {

          Z[i] = sc.nextInt();

      }

      int[] mergedArray = new int[M + N + O];

      int index = 0;

      for (int i = 0; i < M; i++) {

          mergedArray[index++] = X[i];

      }

      for (int i = 0; i < N; i++) {

          mergedArray[index++] = Y[i];

      }

      for (int i = 0; i < O; i++) {

          mergedArray[index++] = Z[i];

      }

      Arrays.*sort*(mergedArray);

      for (int num : mergedArray) {

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

      }

          sc.close();

  }

}