**BYAGARI PAVAN PBL ID:** **J\_251890123**

ARRAYS

1. **Write a program to initialize an integer array and print the sum and average of the array.**

import java.util.Scanner; public class SumAndAverage {

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

System.out.print("Enter number of elements: "); int n = sc.nextInt();

int[] arr = new int[n]; int sum = 0;

System.out.println("Enter " + n + " integers:"); for (int i = 0; i < n; i++) {

arr[i] = sc.nextInt(); sum += arr[i];

}

double average = (double) sum / n;

System.out.println("Sum = " + sum); System.out.println("Average = " + average);

sc.close();

}

}

1. **Write a program to initialize an integer array and find the maximum and minimum value of the array.**

import java.util.Scanner; public class MaxMinArray {

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

System.out.print("Enter number of elements: "); int n = sc.nextInt();

int[] arr = new int[n]; System.out.println("Enter " + n + " integers:"); for (int i = 0; i < n; i++) {

arr[i] = sc.nextInt();

}

int max = arr[0];

int min = arr[0];

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

if (arr[i] > max) max = arr[i]; if (arr[i] < min) min = arr[i];

}

System.out.println("Maximum value = " + max); System.out.println("Minimum value = " + min);

sc.close();

}

}

1. **Write a program to initialize an integer array with values and check if a given number is present in the array or not.**

**If the number is not found, print -1**

**Else print the index value of the given number. Example 1: Array {1,4,34,56,7}, search 90 → Output: -1**

**Example 2: Array {1,4,34,56,7}, search 56 → Output: 3**

import java.util.\*;

public class SearchInArray {

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 key = sc.nextInt(); int index = -1;

for (int i = 0; i < n; i++) { if (arr[i] == key) {

index = i; break;

}

}

System.out.println(index); sc.close();

}

}

Example 1:

Enter number of elements: 6 Enter 6 integers:

1 4 34 56 7 8

Enter the element to search: 90

-1

Example 2:

Enter number of elements: 6 Enter 6 integers:

1 4 34 56 7 8

Enter the element to search: 56 3

1. **Initialize an integer array with ASCII values and print the corresponding character values in a single row.**

public class AsciiToChar {

public static void main(String[] args) {

int[] asciiValues = {72, 101, 108, 108, 111}; // ASCII for "Hello"

for (int value : asciiValues) { System.out.print((char) value);

}

System.out.println();

}

}

1. **Write a Java program to find the largest two numbers and the smallest two numbers in a given integer array**.

import java.util.Scanner; import java.util.Arrays;

public class LargestAndSmallestTwo { public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Read size of array

System.out.print("Enter the number of elements in the array: "); int n = sc.nextInt();

// Read array elements int[] arr = new int[n];

System.out.println("Enter " + n + " integers:"); for (int i = 0; i < n; i++) {

arr[i] = sc.nextInt();

}

// Sort the array Arrays.sort(arr);

// Smallest two numbers int smallest1 = arr[0];

int smallest2 = arr[1];

// Largest two numbers int largest1 = arr[n - 1]; int largest2 = arr[n - 2];

// Display results

System.out.println("Smallest two numbers: " + smallest1 + " and " + smallest2); System.out.println("Largest two numbers: " + largest1 + " and " + largest2);

sc.close();

}

}

# write a program to intialize an array and print them in a sorted order

import java.util.Arrays; public class SortArray {

public static void main(String[] args) {

// Initialize the array int[] arr = {5, 2, 8, 1, 7};

// Display original array

System.out.println("Original array: " + Arrays.toString(arr));

// Sort the array in ascending order Arrays.sort(arr);

// Display sorted array

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

}

}

1. **write a program to remove the duplicate elements in an array and print the same.**

**example:**

# input:{12,34,12,45,67,89}

**output:{12,34,45,67,89}**

import java.util.Arrays;

public class RemoveDuplicatesArray { public static void main(String[] args) {

int[] arr = {12, 34, 12, 45, 67, 89};

int n = arr.length;

System.out.println("Original array: " + Arrays.toString(arr));

// Remove duplicates int[] temp = new int[n]; int newLength = 0;

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

boolean isDuplicate = false;

for (int j = 0; j < newLength; j++) { if (arr[i] == temp[j]) {

isDuplicate = true; break;

}

}

if (!isDuplicate) { temp[newLength] = arr[i]; newLength++;

}

}

// Create result array with correct size int[] uniqueArray = new int[newLength]; for (int i = 0; i < newLength; i++) {

uniqueArray[i] = temp[i];

}

System.out.println("Array after removing duplicates: " + Arrays.toString(uniqueArray));

}

}

1. **write a program to print the sum of the elements of an array following the given below condition.**

**If the array has 6 and 7 in succeeding orders, ignore the numbers between 6 and 7 and consider the other number for calculation of sum.**

**eg1: Arrray Elements-10,3,6,1,2,7,9 output :22 [i.e 10+3+9]**

**eg2:Array Element-7,1,2,3,6 output:19**

**eg3:Array Element-1,6,4,7,9 output:10**

import java.util.Arrays;

public class SumWithSkip {

public static void main(String[] args) {

// Test with multiple examples int[][] testCases = {

{10, 3, 6, 1, 2, 7, 9}, // Expected: 22

{7, 1, 2, 3, 6}, // Expected: 19

{1, 6, 4, 7, 9} // Expected: 10

};

for (int[] arr : testCases) {

System.out.println("Array Elements: " + Arrays.toString(arr)); System.out.println("Sum according to condition: " + calculateSum(arr)); System.out.println();

}

}

public static int calculateSum(int[] arr) { int sum = 0;

boolean skip = false;

for (int num : arr) { if (num == 6) {

skip = true; // Start skipping continue;

}

if (num == 7 && skip) {

skip = false; // Stop skipping continue;

}

if (!skip) {

sum += num;

}

}

return sum;

}

}

1. **print a version of the given array where all the 10's have been removed.The remaining elements should shift left towards the start of the arrray as needed, and the empty spaces at the end of the array should be 0. So {1, 10, 10, 2} yields {1, 2, 0, 0} .You may modify and display the given array or make a new array.**

**withoutTen([1, 10, 10, 2])->[1, 2, 0, 0]**

**withoutTen([10, 2, 10])->[ 2, 0, 0]**

**withoutTen([1, 99, 10])->[1, 99, 0]**

import java.util.Arrays; public class WithoutTen {

public static void main(String[] args) {

// Test cases

int[] arr1 = {1, 10, 10, 2};

int[] arr2 = {10, 2, 10};

int[] arr3 = {1, 99, 10};

System.out.println("withoutTen " + Arrays.toString(arr1) + " -> " + Arrays.toString(withoutTen(arr1)));

System.out.println("withoutTen " + Arrays.toString(arr2) + " -> " + Arrays.toString(withoutTen(arr2)));

System.out.println("withoutTen " + Arrays.toString(arr3) + " -> " + Arrays.toString(withoutTen(arr3)));

}

public static int[] withoutTen(int[] nums) { int[] result = new int[nums.length];

int index = 0;

// Copy all non-10 elements to the front for (int num : nums) {

if (num != 10) { result[index] = num; index++;

}

}

// Remaining elements will automatically be 0 (default value) return result;

}

}

1. **Print an array that contains the exact same numbers as the given array,but rearranged so that all the even numbers come before all the odd numbers.Others than that, the numbers can be in any order.You may modify and print the given array, or make a new array.**

**Evenodd([1, 0, 1, 0, 0, 1, 1])->[0, 0, 0, 1, 1, 1, 1]**

**Evenodd([3, 3, 2])->[2, 3, 3]**

**Evenodd([2, 2, 2])->[2, ,2, 2]**

import java.util.Arrays;

public class EvenOddSimple {

public static void main(String[] args) { int[] arr = {1, 0, 1, 0, 0, 1, 1};

int[] result = new int[arr.length]; int index = 0;

// Put even numbers first for (int num : arr) {

if (num % 2 == 0) result[index++] = num;

}

// Then odd numbers for (int num : arr) {

if (num % 2 != 0) result[index++] = num;

}

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

}

}

1. **Given an array of type int,print true if every element is 1 or 4. Only14([1, 4, 1, 4])->true**

**Only14([1, 4, 2, 4])->false**

**Only14([1, 1])->true**

public class Only14 {

public static void main(String[] args) { int[] arr1 = {1, 4, 1, 4};

int[] arr2 = {1, 4, 2, 4};

int[] arr3 = {1, 1};

System.out.println(only14(arr1)); // true System.out.println(only14(arr2)); // false System.out.println(only14(arr3)); // true

}

public static boolean only14(int[] nums) { for (int num : nums) {

if (num != 1 && num != 4) { return false;

}

}

return true;

}

}

1. **Given 2 int array, a and b, each length 3, form a new array of length 2, containing their middle elements.**

**Middleway ([1, 2, 3], [4, 5, 6]) -> [2, 5]**

**Middleway ([7, 7, 7], [3, 8, 0]) -> [7, 8]**

**Middleway ([5, 2, 9], [1, 4, 5]) -> [2, 4]**

import java.util.Arrays;

public class MiddleWay {

public static void main(String[] args) { int[] a1 = {1, 2, 3};

int[] b1 = {4, 5, 6};

int[] a2 = {7, 7, 7};

int[] b2 = {3, 8, 0};

int[] a3 = {5, 2, 9};

int[] b3 = {1, 4, 5};

System.out.println(Arrays.toString(middleWay(a1, b1))); // [2, 5]

System.out.println(Arrays.toString(middleWay(a2, b2))); // [7, 8]

System.out.println(Arrays.toString(middleWay(a3, b3))); // [2, 4]

}

public static int[] middleWay(int[] a, int[] b) { return new int[]{a[1], b[1]};

}

}

1. **Write a program to reverse the elements of a given 2\*2 array.Four integers numbers needs to be passed as command line arguments.**

**Example 1:**

**C : \>java sample 1 2 3**

**Output: please enter 4 integers numbers Example 2:**

**C : \>java sample 1 2 3 4 Output:**

**The given array is : 1 2**

**3 4**

**The reverse of the array is: 4 3**

**2 1**

public class Sample {

public static void main(String[] args) {

// Check if exactly 4 integers are passed if (args.length != 4) {

System.out.println("please enter 4 integers numbers"); return;

}

int[][] arr = new int[2][2]; int k = 0;

// Fill the array

for (int i = 0; i < 2; i++) { for (int j = 0; j < 2; j++) {

arr[i][j] = Integer.parseInt(args[k++]);

}

}

// Print original array System.out.println("The given array is:"); for (int i = 0; i < 2; i++) {

for (int j = 0; j < 2; j++) { System.out.print(arr[i][j] + " ");

}

System.out.println();

}

// Print reverse array

System.out.println("The reverse of the array is:"); for (int i = 1; i >= 0; i--) {

for (int j = 1; j >= 0; j--) { System.out.print(arr[i][j] + " ");

}

System.out.println();

}

}

}

1. **Write a program to find the biggest number in a 3\*3 array.The program is supposed to receive 9 integer numbers as command line arguments.**

**Example1:**

**C : \>java sample 1 2 3**

**Output: please enter 9 integers numbers Example 2:**

**C : \>java sample 1 23 45 55 121 222 56 77 89 Output:**

**The given array is :**

**1 23 45**

**55 121 222**

**56 77 89**

**The biggest number in the array is 222**

import java.util.\*;

public class Main {

public static void main(String[] args) { if (args.length != 9) {

System.out.println("please enter 9 integers numbers"); return;

}

int[][] arr = new int[3][3];

int k = 0, big = Integer.MIN\_VALUE;

for (int i = 0; i < 3; i++) { for (int j = 0; j < 3; j++) {

arr[i][j] = Integer.parseInt(args[k++]); big = Math.max(big, arr[i][j]);

}

}

System.out.println("The given array is:"); for (int[] row : arr) {

for (int n : row) System.out.print(n + " "); System.out.println();

}

System.out.println("The biggest number in the array is " + big);

}

}