**1D Arrays Assignment Questions**

1. Write a program to print the sum of all the elements present on even indices in the given array. Input 1: arr[] = {3,20,4,6,9}

**Solution**:

public class Main {

public static void main(String[] args) {

int[] array = {1, 2, 3, 4, 5, 6, 7, 8, 9}; // Example array

int sum = sumOfEvenIndices(array);

System.out.println("Sum of elements at even indices: " + sum);

}

public static int sumOfEvenIndices(int[] array) {

int sum = 0;

for (int i = 0; i < array.length; i += 2) { // Incrementing by 2 to get even indices

sum += array[i];

}

return sum;

}

}

1. Write a program to traverse over the elements of the array using for each loop and print all even

elements.

**Solution**:

public class Main {

public static void main(String[] args) {

int[] array = {1, 2, 3, 4, 5, 6, 7, 8, 9}; // Example array

System.out.println("Even elements in the array:");

printEvenElements(array);

}

public static void printEvenElements(int[] array) {

for (int num : array) {

if (num % 2 == 0) {

System.out.println(num);

}

}

}

}

1. Write a program to calculate the maximum element in the array.

**Solution**:

public class FindMax {

public static void main(String[] args) {

// Sample array

int[] arr = {25, 11, 7, 75, 56};

// Initialize max with the first element

int max = arr[0];

// Loop through the array

for (int i = 1; i < arr.length; i++) {

// Compare elements with max and update if greater

if (arr[i] > max) {

max = arr[i];

}

}

// Print the maximum element

System.out.println("Largest element present in given array: " + max);

}

}

1. Write a program to find out the second largest element in a given array.

Solution:

public class Main {

public static void main(String[] args) {

int[] array = {10, 5, 20, 15, 25}; // Example array

int secondLargest = findSecondLargest(array);

System.out.println("Second largest element in the array: " + secondLargest);

}

public static int findSecondLargest(int[] array) {

int largest = Integer.MIN\_VALUE;

int secondLargest = Integer.MIN\_VALUE;

for (int num : array) {

if (num > largest) {

secondLargest = largest;

largest = num;

} else if (num > secondLargest && num != largest) {

secondLargest = num;

}

}

return secondLargest;

}

}

1. Given an array. Find the first peak element in the array. A peak element is an element that is greater than its just left and just right neighbour.

Solution:

public class Main {

public static void main(String[] args) {

int[] array = {1, 3, 4, 5, 2, 6}; // Example array

int peakIndex = findFirstPeak(array);

if (peakIndex != -1) {

System.out.println("The first peak element is at index: " + peakIndex);

System.out.println("The peak element is: " + array[peakIndex]);

} else {

System.out.println("There is no peak element in the array.");

}

}

public static int findFirstPeak(int[] array) {

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

if (array[i] > array[i - 1] && array[i] > array[i + 1]) {

return i;

}

}

return -1; // No peak element found

}

}

**2D Arrays Assignment Questions**

1. Take m and n input from the user and m \* n integer inputs from user and print the following:

number of positive numbers

number of negative numbers

number of odd numbers

number of even numbers

number of 0.

**Solution**:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int m = scanner.nextInt();

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

int n = scanner.nextInt();

int[][] matrix = new int[m][n];

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

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

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

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

}

}

scanner.close();

int positiveCount = 0;

int negativeCount = 0;

int oddCount = 0;

int evenCount = 0;

int zeroCount = 0;

// Counting numbers in the matrix

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

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

int num = matrix[i][j];

if (num > 0) {

positiveCount++;

} else if (num < 0) {

negativeCount++;

}

if (num % 2 == 0 && num != 0) {

evenCount++;

} else if (num % 2 != 0) {

oddCount++;

} else if (num == 0) {

zeroCount++;

}

}

}

System.out.println("Number of positive numbers: " + positiveCount);

System.out.println("Number of negative numbers: " + negativeCount);

System.out.println("Number of odd numbers: " + oddCount);

System.out.println("Number of even numbers: " + evenCount);

System.out.println("Number of zeros: " + zeroCount);

}

}

1. write a program to print the elements above the secondary diagonal in a user inputted

square matrix.

Solution:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the size of the square matrix: ");

int size = scanner.nextInt();

int[][] matrix = new int[size][size];

System.out.println("Enter the elements of the square matrix:");

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

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

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

}

}

System.out.println("Elements above the secondary diagonal:");

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

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

if (i + j < size - 1) {

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

}

}

}

scanner.close();

}

}

1. write a program to print the elements of both the diagonals in a user inputted square matrix in any order.

Solution:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the size of the square matrix: ");

int size = scanner.nextInt();

int[][] matrix = new int[size][size];

System.out.println("Enter the elements of the square matrix:");

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

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

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

}

}

System.out.println("Elements of the main diagonal:");

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

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

}

System.out.println();

System.out.println("Elements of the secondary diagonal:");

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

System.out.print(matrix[i][size - i - 1] + " ");

}

System.out.println();

scanner.close();

}

}

1. Write a program to find the largest element of a given 2D array of integers.

Solution:

public class Main {

public static void main(String[] args) {

int[][] array = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

}; // Example 2D array

int largest = findLargestElement(array);

System.out.println("The largest element in the 2D array is: " + largest);

}

public static int findLargestElement(int[][] array) {

int largest = Integer.MIN\_VALUE;

for (int[] row : array) {

for (int num : row) {

if (num > largest) {

largest = num;

}

}

}

return largest;

}

}

1. Write a function which accepts a 2D array of integers and its size as arguments and displays the elements of middle row and the elements of middle column. Printing can be done in any order. [Assuming the 2D Array to be a square matrix with odd dimensions i.e. 3x3, 5x5, 7x7 etc...]

Solution:

public class Main {

public static void main(String[] args) {

int[][] array = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

}; // Example 2D array

displayMiddleElements(array, 3);

}

public static void displayMiddleElements(int[][] array, int size) {

// Display middle row

int middleRow = size / 2;

System.out.println("Middle row elements:");

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

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

}

System.out.println();

// Display middle column

int middleColumn = size / 2;

System.out.println("Middle column elements:");

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

System.out.println(array[i][middleColumn]);

}

}

}