1.

import java.util.\*;

public class ZigzagSum

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

int m, n, sum = 0, row=0, col=0, diag=0;

System.out.print("\nEnter the number of rows:");

m = sc.nextInt();

        System.out.print("\nEnter the number of columns:");

n = sc.nextInt();

int i, j;

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

System.out.print("\nInput the matrix elements \n");

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

{

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

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

}

            for(i = 0; i < m; i++)

{

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

                {

System.out.print(mat[i][j]+"\t");

                }

                System.out.println("\n");

}

for(i = 0; i <= 0; i++)

{

for(j = 0; j < n-1; j++)

{

row=row+mat[i][j];

}

}

for(j = n-1; j==n-1; j--)

{

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

{

col=col+mat[j][i];

}

}

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

{

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

{

if ((i + j) == (m-1))

{

diag += mat[i][j];

}

if(j==0 && i==m-1)

col=col-mat[i][j];

}

}

sum=diag+row+col;

System.out.print("Sum of Zig-Zag pattern is " + sum);

    }

}

A screenshot of a computer program

Description automatically generated

2.

import java.util.Scanner;

public class CountNumbers

{

    public static void main(String[] args)

    {

        Scanner scanner = new Scanner(System.in);

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

        int n = scanner.nextInt();

        int[] array = new int[n];

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

        for (int i = 0; i < n; i++) {

            array[i] = scanner.nextInt();

        }

        int positiveCount = 0;

        int negativeCount = 0;

        int oddCount = 0;

        int evenCount = 0;

        int zeroCount = 0;

        for (int num : array)

        {

            if (num > 0)

            {

                positiveCount++;

            } else if (num < 0)

            {

                negativeCount++;

            }

            if (num % 2 == 0)

            {

                evenCount++;

            }

            else

            {

                oddCount++;

            }

            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);

    }

}

A screenshot of a computer

Description automatically generated

3.

import java.util.Scanner;

public class MatrixTranspose {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

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

        int rows = scanner.nextInt();

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

        int cols = scanner.nextInt();

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

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

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

        {

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

            {

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

            }

        }

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

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

        {

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

            {

                transpose[j][i] = matrix[i][j];

            }

        }

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

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

        {

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

            {

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

            }

        }

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

        for (int i = 0; i < cols; i++)

        {

            for (int j = 0; j < rows; j++)

            {

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

            }

            System.out.println();

        }

    }

}

A screenshot of a computer

Description automatically generated

4.

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

public class ElementFrequency {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

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

        int n = scanner.nextInt();

        int[] array = new int[n];

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

        for (int i = 0; i < n; i++) {

            array[i] = scanner.nextInt();

        }

        // Using a HashMap to store element-frequency pairs

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

        for (int num : array) {

            // If the element is already in the map, increment its frequency

            if (frequencyMap.containsKey(num)) {

                frequencyMap.put(num, frequencyMap.get(num) + 1);

            } else {

                // If the element is not in the map, add it with frequency 1

                frequencyMap.put(num, 1);

            }

        }

        // Printing the frequency of each element

        System.out.println("Frequency of each element:");

        for (Map.Entry<Integer, Integer> entry : frequencyMap.entrySet()) {

            System.out.println(entry.getKey() + ": " + entry.getValue());

        }

    }

}

A screenshot of a computer program

Description automatically generated

5.

import java.util.Scanner;

public class MatrixOperations

{

     public static void main(String[] args)

     {

        Scanner scanner = new Scanner(System.in);

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

        int rows = scanner.nextInt();

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

        int cols = scanner.nextInt();

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

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

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

        readMatrixElements(matrixA, scanner);

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

        readMatrixElements(matrixB, scanner);

        int[][] sumMatrix = performMatrixOperation(matrixA, matrixB, "add");

        int[][] diffMatrix = performMatrixOperation(matrixA, matrixB, "subtract");

        int[][] productMatrix = performMatrixOperation(matrixA, matrixB, "multiply");

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

        printMatrix(matrixA);

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

        printMatrix(matrixB);

        System.out.println("Matrix A + Matrix B:");

        printMatrix(sumMatrix);

        System.out.println("Matrix A - Matrix B:");

        printMatrix(diffMatrix);

        System.out.println("Matrix A \* Matrix B:");

        printMatrix(productMatrix);

    }

    public static void readMatrixElements(int[][] matrix, Scanner scanner)

    {

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

        {

            for (int j = 0; j < matrix[i].length; j++)

            {

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

            }

        }

    }

    public static int[][] performMatrixOperation(int[][] matrixA, int[][] matrixB, String operation)

    {

        int rows = matrixA.length;

        int cols = matrixA[0].length;

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

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

        {

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

            {

                if (operation.equals("add"))

                {

                    resultMatrix[i][j] = matrixA[i][j] + matrixB[i][j];

                }

                else if (operation.equals("subtract"))

                {

                    resultMatrix[i][j] = matrixA[i][j] - matrixB[i][j];

                }

                else if (operation.equals("multiply"))

                {

                    resultMatrix[i][j] = matrixA[i][j] \* matrixB[i][j];

                }

            }

        }

        return resultMatrix;

    }

    public static void printMatrix(int[][] matrix)

    {

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

        {

            for (int j = 0; j < matrix[i].length; j++)

            {

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

            }

        }

    }

}

A screenshot of a computer program

Description automatically generated