SANYAM AGRAWAL – SE21UCSE192 – CSE3

OOPS Lab Assignment 9(Matrix Operations)

Source Code:->

```
import java.util.Scanner;

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

        // Hardcoded 3x3 matrix
        int[][] matrixA = {
                {1, 19, 3},
                {4, 11, 6},
                {7, 8, 13}
        };

        // Display the hardcoded matrix
        System.out.println("Matrix A:");
        printMatrix(matrixA);

        // Get user input for the second matrix
        System.out.println("Enter the second matrix B (3x3):");
        int[][] matrixB = new int[3][3];
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
```

```
                System.out.print("Enter element at position (" + (i + 1) + "," +
(j + 1) + "): ");
                matrixB[i][j] = scanner.nextInt();
            }
        }

        // Display the second matrix
        System.out.println("Matrix B:");
        printMatrix(matrixB);

        // Perform matrix operations based on user input
        System.out.println("Select the matrix operation:");
        System.out.println("1. Matrix Addition");
        System.out.println("2. Matrix Subtraction");
        System.out.println("3. Scalar Matrix Addition");
        System.out.println("4. Scalar Matrix Subtraction");
        System.out.println("5. Matrix Multiplication");
        System.out.println("6. Matrix Transposition");
        System.out.println("7. Matrix Determinant");
        System.out.println("8. Matrix Inversion");

        int choice = scanner.nextInt();

        switch (choice) {
            case 1:
                matrixAddition(matrixA, matrixB);
                break;
            case 2:
                matrixSubtraction(matrixA, matrixB);
```

```
            break;
        case 3:
            scalarMatrixAddition(matrixA, 2, "hardcoded");
            scalarMatrixAddition(matrixB, 2, "user specified");
            break;
        case 4:
            scalarMatrixSubtraction(matrixA, 2, "hardcoded");
            scalarMatrixSubtraction(matrixB, 2, "user specified");
            break;
        case 5:
            matrixMultiplication(matrixA, matrixB);
            break;
        case 6:
            matrixTransposition(matrixA, "hardcoded");
            matrixTransposition(matrixB, "user specified");
            break;
        case 7:
            matrixDeterminant(matrixA, "hardcoded");
            matrixDeterminant(matrixB, "user specified");
            break;
        case 8:
            matrixInversion(matrixA, "hardcoded");
            matrixInversion(matrixB, "user specified");
            break;
        default:
            System.out.println("Invalid choice");
    }

    scanner.close();
```

```
    }

    // Helper method to print a matrix
    private static void printMatrix(int[][] matrix) {
        for (int[] row : matrix) {
            for (int element : row) {
                System.out.print(element + " ");
            }
            System.out.println();
        }
    }

    // Matrix Addition
    private static void matrixAddition(int[][] matrixA, int[][] matrixB) {
        int[][] result = new int[3][3];
        System.out.println("Result of Matrix Addition:");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                result[i][j] = matrixA[i][j] + matrixB[i][j];
                System.out.print(result[i][j] + " ");
            }
            System.out.println();
        }
    }

    // Matrix Subtraction
    private static void matrixSubtraction(int[][] matrixA, int[][] matrixB) {
        int[][] result = new int[3][3];
        System.out.println("Result of Matrix Subtraction:");
```

```
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                result[i][j] = matrixA[i][j] - matrixB[i][j];
                System.out.print(result[i][j] + " ");
            }
            System.out.println();
        }
    }

    // Scalar Matrix Addition
    private static void scalarMatrixAddition(int[][] matrix, int scalar, String
matrixType) {
        int[][] result = new int[3][3];
        System.out.println("Result of Scalar Matrix Addition on " + matrixType +
" matrix:");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                result[i][j] = matrix[i][j] + scalar;
                System.out.print(result[i][j] + " ");
            }
            System.out.println();
        }
    }

    // Scalar Matrix Subtraction
    private static void scalarMatrixSubtraction(int[][] matrix, int scalar,
String matrixType) {
        int[][] result = new int[3][3];
        System.out.println("Result of Scalar Matrix Subtraction on " + matrixType
+ " matrix:");
```

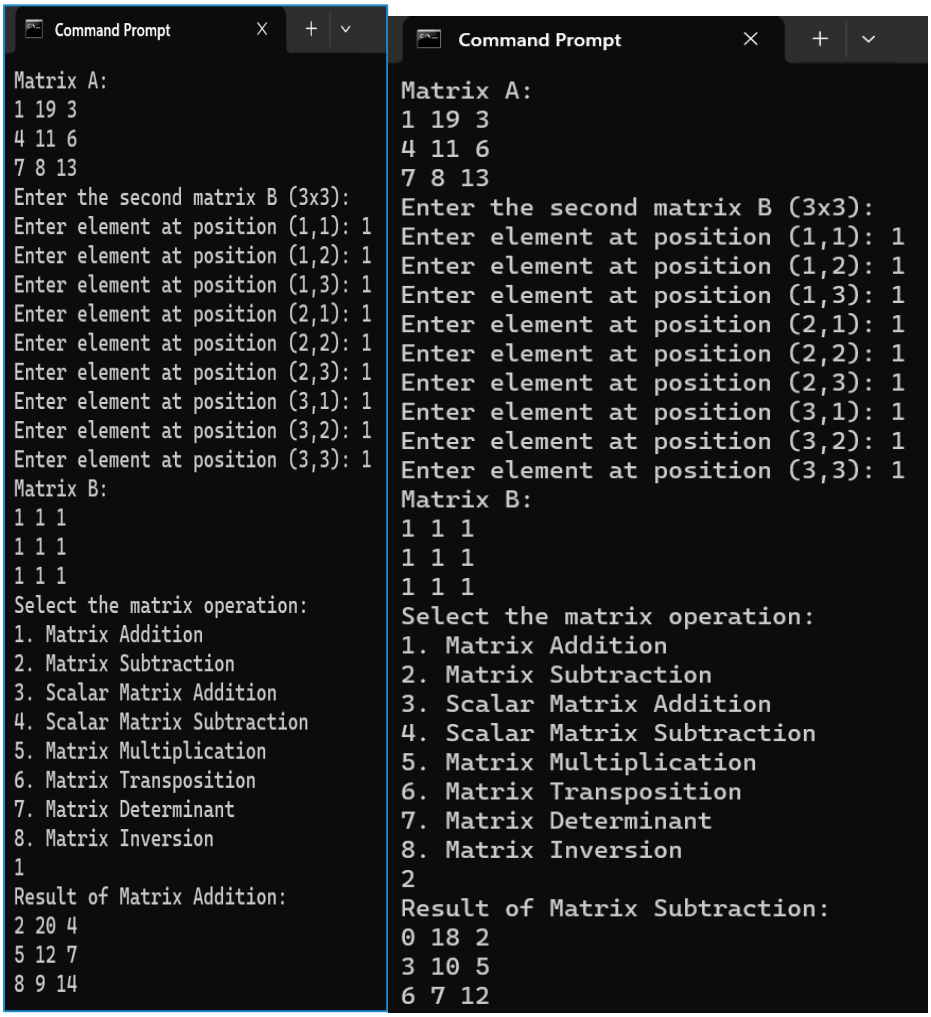
```
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                result[i][j] = matrix[i][j] - scalar;
                System.out.print(result[i][j] + " ");
            }
            System.out.println();
        }
    }

    // Matrix Multiplication
    private static void matrixMultiplication(int[][] matrixA, int[][] matrixB) {
        int[][] result = new int[3][3];
        System.out.println("Result of Matrix Multiplication:");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                for (int k = 0; k < 3; k++) {
                    result[i][j] += matrixA[i][k] * matrixB[k][j];
                }
                System.out.print(result[i][j] + " ");
            }
            System.out.println();
        }
    }

    // Matrix Transposition
    private static void matrixTransposition(int[][] matrix, String matrixType) {
        int[][] result = new int[3][3];
        System.out.println("Result of Matrix Transposition on " + matrixType + "
matrix:");
```

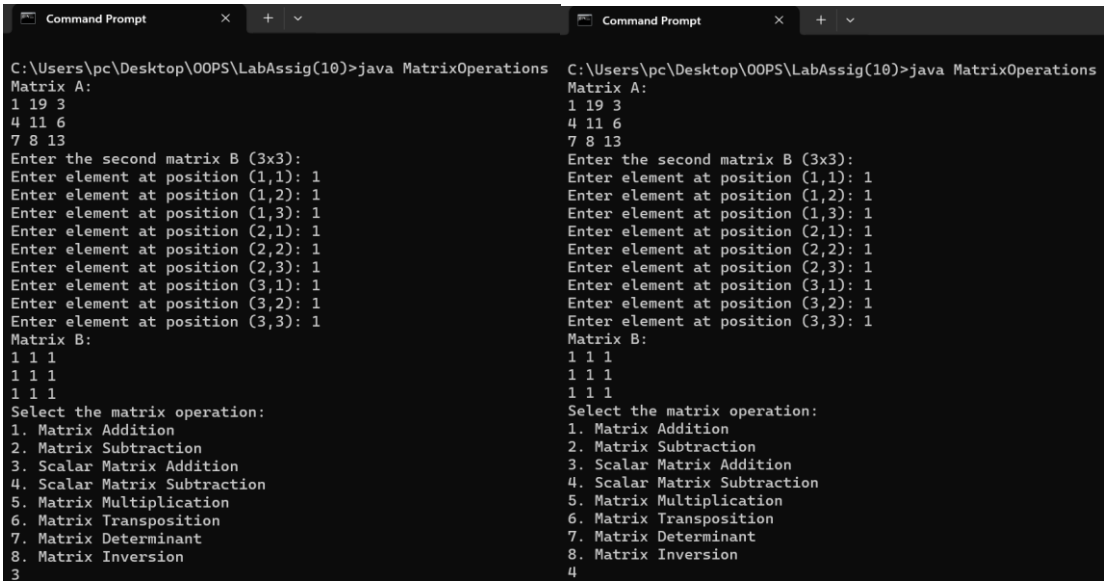
```
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                result[i][j] = matrix[j][i];
                System.out.print(result[i][j] + " ");
            }
            System.out.println();
        }
    }

    // Matrix Determinant
    private static void matrixDeterminant(int[][] matrix, String matrixType) {
        int determinant = matrix[0][0] * (matrix[1][1] * matrix[2][2] -
matrix[1][2] * matrix[2][1])
                - matrix[0][1] * (matrix[1][0] * matrix[2][2] - matrix[1][2] *
matrix[2][0])
                + matrix[0][2] * (matrix[1][0] * matrix[2][1] - matrix[1][1] *
matrix[2][0]);

        System.out.println("Determinant on " + matrixType + " matrix: " +
determinant);
    }

    // Matrix Inversion
    private static void matrixInversion(int[][] matrix, String matrixType) {
        int determinant = matrix[0][0] * (matrix[1][1] * matrix[2][2] -
matrix[1][2] * matrix[2][1])
                - matrix[0][1] * (matrix[1][0] * matrix[2][2] - matrix[1][2] *
matrix[2][0])
                + matrix[0][2] * (matrix[1][0] * matrix[2][1] - matrix[1][1] *
matrix[2][0]);

        if (determinant == 0) {
```

```
            System.out.println("Matrix is not invertible (determinant is
zero).");
            return;
        }

        int[][] result = new int[3][3];

        result[0][0] = (matrix[1][1] * matrix[2][2] - matrix[1][2] *
matrix[2][1]);
        result[0][1] = (matrix[0][2] * matrix[2][1] - matrix[0][1] *
matrix[2][2]);
        result[0][2] = (matrix[0][1] * matrix[1][2] - matrix[0][2] *
matrix[1][1]);

        result[1][0] = (matrix[1][2] * matrix[2][0] - matrix[1][0] *
matrix[2][2]);
        result[1][1] = (matrix[0][0] * matrix[2][2] - matrix[0][2] *
matrix[2][0]);
        result[1][2] = (matrix[0][2] * matrix[1][0] - matrix[0][0] *
matrix[1][2]);

        result[2][0] = (matrix[1][0] * matrix[2][1] - matrix[1][1] *
matrix[2][0]);
        result[2][1] = (matrix[0][1] * matrix[2][0] - matrix[0][0] *
matrix[2][1]);
        result[2][2] = (matrix[0][0] * matrix[1][1] - matrix[0][1] *
matrix[1][0]);

        System.out.println("Result of Matrix Inversion on " + matrixType + "
matrix:");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                result[i][j] /= determinant;
```

```
                System.out.print(result[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

Some Output Screenshots:->



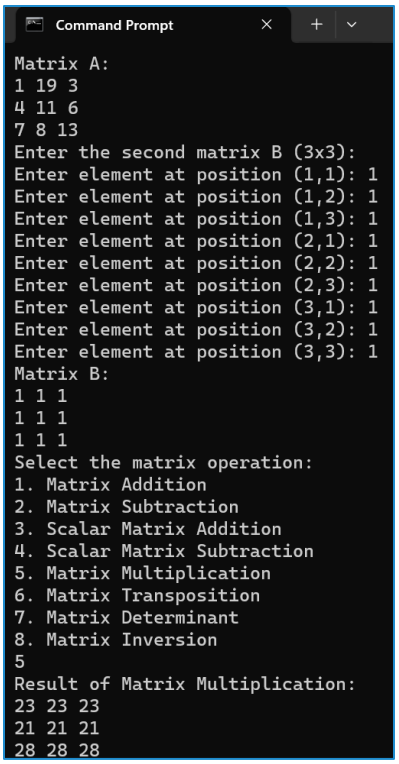


```
Command Prompt

C:\Users\pc\Desktop\OOPS\LabAssig(10)>java MatrixOperations
Matrix A:
1 19 3
4 11 6
7 8 13
Enter the second matrix B (3x3):
Enter element at position (1,1): 1
Enter element at position (1,2): 1
Enter element at position (1,3): 1
Enter element at position (2,1): 1
Enter element at position (2,2): 1
Enter element at position (2,3): 1
Enter element at position (3,1): 1
Enter element at position (3,2): 1
Enter element at position (3,3): 1
Matrix B:
1 1 1
1 1 1
1 1 1
Select the matrix operation:
1. Matrix Addition
2. Matrix Subtraction
3. Scalar Matrix Addition
4. Scalar Matrix Subtraction
5. Matrix Multiplication
6. Matrix Transposition
7. Matrix Determinant
8. Matrix Inversion
3
```

```
Command Prompt

C:\Users\pc\Desktop\OOPS\LabAssig(10)>java MatrixOperations
Matrix A:
1 19 3
4 11 6
7 8 13
Enter the second matrix B (3x3):
Enter element at position (1,1): 1
Enter element at position (1,2): 1
Enter element at position (1,3): 1
Enter element at position (2,1): 1
Enter element at position (2,2): 1
Enter element at position (2,3): 1
Enter element at position (3,1): 1
Enter element at position (3,2): 1
Enter element at position (3,3): 1
Matrix B:
1 1 1
1 1 1
1 1 1
Select the matrix operation:
1. Matrix Addition
2. Matrix Subtraction
3. Scalar Matrix Addition
4. Scalar Matrix Subtraction
5. Matrix Multiplication
6. Matrix Transposition
7. Matrix Determinant
8. Matrix Inversion
4
```

```
Result of Scalar Matrix Addition on hardcoded matrix:
3 21 5
6 13 8
9 10 15
Result of Scalar Matrix Addition on user specified matrix:
3 3 3
3 3 3
3 3 3
```

```
Result of Scalar Matrix Subtraction on hardcoded matrix:
-1 17 1
2 9 4
5 6 11
Result of Scalar Matrix Subtraction on user specified matrix:
-1 -1 -1
-1 -1 -1
-1 -1 -1
```


Command Prompt
Matrix A:
1 19 3
4 11 6
7 8 13
Enter the second matrix B (3x3):
Enter element at position (1,1): 1
Enter element at position (1,2): 1
Enter element at position (1,3): 1
Enter element at position (2,1): 1
Enter element at position (2,2): 1
Enter element at position (2,3): 1
Enter element at position (3,1): 1
Enter element at position (3,2): 1
Enter element at position (3,3): 1
Matrix B:
1 1 1
1 1 1
1 1 1
Select the matrix operation:
1. Matrix Addition
2. Matrix Subtraction
3. Scalar Matrix Addition
4. Scalar Matrix Subtraction
5. Matrix Multiplication
6. Matrix Transposition
7. Matrix Determinant
8. Matrix Inversion
5
Result of Matrix Multiplication:
23 23 23
21 21 21
28 28 28