**COAL Lab Project Report**

**National University of Computer and Emerging Sciences**

**Matrix Operations**

**Batch: 2023**

**Section: CY-3A**

**Group Members**

Talal Ali (23K-2003)

Daniyal Ahmed (23K-2011)

Muhammad Hammad (23K-2005)

**Lab Instructor**

Mr. Waseem Rauf

**Department of Cyber Security**

**National University of Computer and Emerging Sciences – FAST Karachi Campus**

**Acknowledgement**

We extend our heartfelt gratitude to Mr. Waseem Rauf for their guidance and support throughout this project. We also thank our peers for their feedback and our institution for providing essential resources that contributed to the successful completion of this project.

**Abstract**

This project presents a matrix calculator in assembly language, performing addition, subtraction, multiplication, and division. The program validates matrix dimensions before executing operations, providing accurate results. This low-level implementation demonstrates the practicality and efficiency of assembly in handling mathematical computations, offering a deeper understanding of matrix operations and computer architecture.

**Project Overview**

The matrix calculator is designed to perform essential matrix operations using assembly language. Users input two matrices and select an operation, with the program validating compatibility before displaying results. The project highlights matrix handling, error detection, and real-time computation, offering a hands-on application of linear algebra concepts, diverse environments, making it ideal for applications in robotics, automation, and education.

**Introduction**

Matrix operations are widely used in various fields, typically implemented in high-level languages for simplicity. This project takes on the challenge of implementing these operations in assembly language, combining concepts from Linear Algebra and COAL courses. By developing this calculator, we deepen our understanding of both matrix computations and low-level programming.

**Features:**

1. **Matrix Dimension Validation:**

Ensure that matrix dimensions are compatible for the selected operation (e.g., matching columns and rows for multiplication).

1. **Signed Integers Handling:**

Supports operations on matrices with signed integers, correctly managing both positive and negative values during calculations.

1. **Error Handling:**

Displays an error message if the chosen operation is not valid for the given matrices or if the dimensions are not valid for a particular operation.

1. **Matrix Addition:**

Adds corresponding elements of two matrices and outputs the resultant matrix.

1. **Matrix Subtraction:**

Subtracts elements of the second matrix from the first and displays the result.

1. **Matrix Multiplication:**

Multiplies compatible matrices and outputs the resultant product matrix.

1. **Matrix Transposition:**

Transposes the given matrix by swapping rows and columns, displaying the resulting matrix with dimensions reversed.

1. **Result Display:**

Outputs the final computed matrix in a clear, formatted manner.

**Source Code:**

A screenshot of a computer

Description automatically generated

A computer screen with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

A computer screen shot of a black screen

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer screen shot of a black screen

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A computer screen shot of white text

Description automatically generated

**Output**

1. **Matrix Transposition**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

1. **Matrix Addition**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **Matrix Subtraction**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

1. **Matrix Multiplication**

A screen shot of a computer

Description automatically generated

A screenshot of a computer

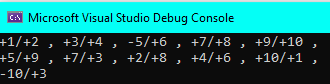
Description automatically generated

**5 Matrix Division**

**NOTE: division typically isn’t defined in matrices, so for division, we made the prototype using structures and fractions in a different program**

**(Code given in comment below the project)**

**for a hardcoded array: (dividing element 3 of matrix 1 with matrix 2 )**

****

**((-5/6) / (2/8)) = -5/6 \* 8/2 = -40/12 = -10/3**

**The division handles fractions, signed values, and simplification of the fractions using the following:**

**-) A structure that contains numerator and denominator**

**-) A function to Flip a fraction (convert division into multiplication)**

**-) A function to simplify the resultant fraction**