Implementation of Generic Matrix Library

Jasraj Singh Anand
PES1201700171
CSE DEPT.
PES UNIVERSITY
BANGALORE, INDIA
awesomejasraj@gmail.com

***Abstract*—This document describes the implementation of a generic version of the Matrix library which supports different initializations, iterators and the basic operations which can be performed on a matrix.**
***Keywords*— Matrix, Iterators, Initializations, Generic, Operations.**

I. INTRODUCTION

Generic programming is a methodology for simplifying the development of libraries in which a set of algorithms have to be implemented for many data structures. **,** the C++ Standard Template Library (STL) [3] uses the API of one-dimensional sequences as the interface between data structures like arrays and lists, and algorithms like searching and sorting. The type systems of modern languages permit the data structure implementations and generic programs to be type-checked and compiled separately; a concrete implementation is produced by linking a generic program with a particular data structure implementation. We want to create a two-dimensional data structure which can be used for creating matrices. A matrix is a two-dimensional data structure made of m rows and n columns, therefore having a total of m*n elements. Our goal is to create a matrix library which supports different data types like int, float, double, and on which basic operations like displaying the cardinality, degree of the matrix, no. of elements present in the matrix, transpose of the matrix etc. can be performed with ease.

II. STRUCTURE OF THE LIBRARY

The library supports the following initializations while creating a Matrix:

1. Initialization using a 2D vector.
2. Initialization using a copy constructor.
3. Initialization by specifying the no. of rows, columns and an optional default value.
4. Initialization by an Identity Matrix of a given order.

The library supports 2 types of iterators:

1. Row Iterator
2. Column Iterator

III. FEATURES OF THE LIBRARY

The library supports the following most used operations which can be performed on matrices:

1. Details of the matrix
   a) Cardinality of the matrix.
   b) Degree of the matrix.
   c) No. of elements present in the matrix.
   d) Random access of an element present in the matrix.

2. Singular Operations
   a) Transpose of the matrix.
   b) Adjacent of the matrix.
   c) Co-factor of the matrix.
   d) Inverse of the matrix.
   e) Determinant of the matrix.
   f) Resizing the matrix.
   g) Power of a matrix

3. Binary Operations
   a) Addition of two matrices.
   b) Subtraction of two matrices.
   c) Multiplication of two matrices.

4. Scalar Operations
   a) Addition of a scalar to all the elements of the matrix.
   b) Subtraction of a scalar from all the elements of the matrix.
   c) Multiplication of a scalar to all the elements of the matrix.
   d) Division by a scalar from all the elements of the matrix.

5. Other features
   a) Display function.
   b) Displaying the matrix with cout (using std::ostream).
   c) Change operators like +=, -=, *=, /=.
   d) Check functions (Square matrix, Identity matrix).
   e) Append row/column.

IV. LITERATURE REVIEW

A. Square Matrix

A square matrix is a matrix with the same number of rows and columns. An n-by-n matrix is known as a square matrix of order n. Any two square matrices of the same order can be added and multiplied.

B. Identity Matrix

A square matrix in which all the elements of the principal diagonal are ones and all other elements are zeros. The effect of multiplying a given matrix by an identity matrix is to leave the given matrix unchanged.

C. Cardinality of a matrix

No. of rows in a matrix

D. Degree of a matrix

No. of columns in a matrix.

E. Transpose of a matrix

The transpose of a matrix is an operator which flips a matrix over its diagonal, that is it switches the row and column indices of the matrix by producing another matrix

F. Determinant of a matrix

The determinant is a scalar value that can be computed from the elements of a square matrix and encodes certain properties of the linear transformation described by the matrix. The determinant of a matrix A is denoted det, det A, or |A|.

G. Inverse of a matrix

The inverse of A is A^{-1} only when:

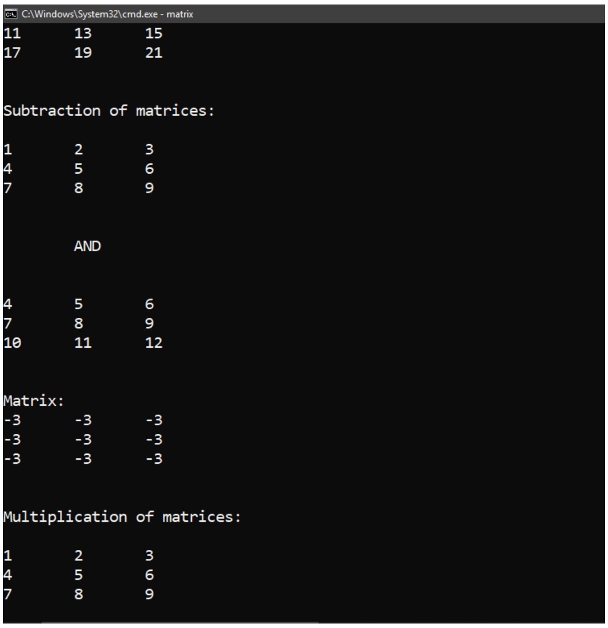
$$A \times A^{-1} = A^{-1} \times A = \mathbf{I}$$

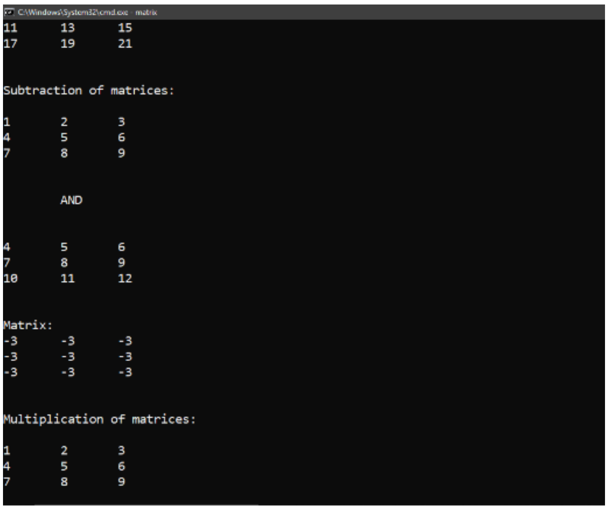
Sometimes there is no inverse at all. Mathematically, for a matrix whose inverse is possible is given as the Adjoint of the matrix divided by its determinant.

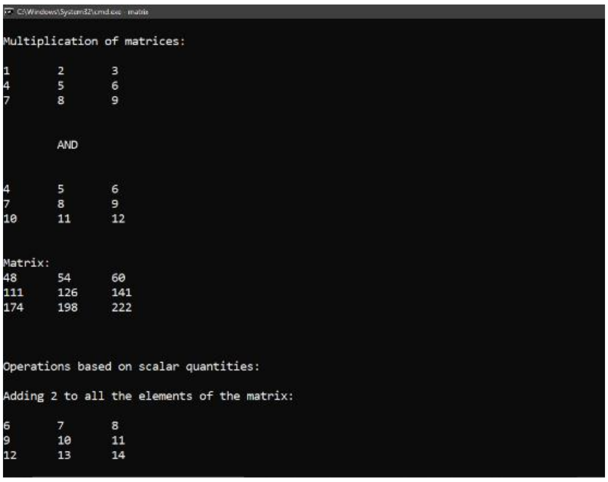
V. RESULTS

Some scalar operations

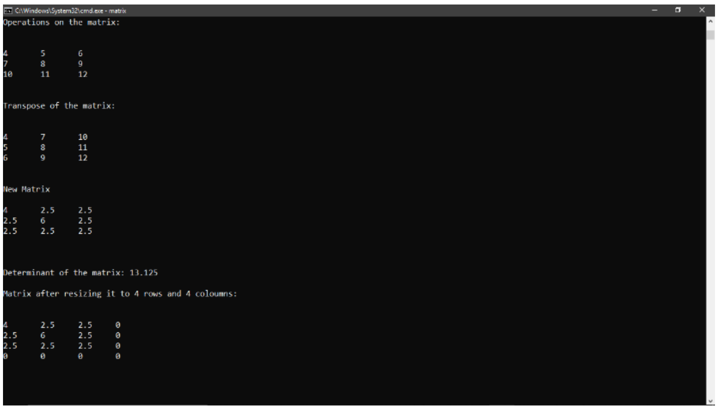
Some Binary operations







Other Results



VI. ACKNOWLEDGMENT

I would like to express my sincere gratitude to Prof Kumar NS for providing their guidance, comments and suggestions through the course of this project. Throughout this project I was able to experiment different designs for implementing the Matrix library for generic uses and for better efficiency.

VII. REFERENCES

[1] https://www.geeksforgeeks.org/
[2] https://www.mathsisfun.com/
[3] https://www.wikipedia.org/
[4] http://www.cplusplus.com/
[5] Kumar NS Sir's C++ Notes