



Fundamental of Computer Programming

«Matrix Calculator»

Author: Orivri Solomon

Instructor: Dr Inz. Wojciech Dudzik

Year: 2021/2022

Lab Group: Group 14:30 – 16:00

Deadline: 2022-01-31

$$\begin{bmatrix} 2 & -1 \\ 0 & 1 \\ 1 & 3 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 2 & -1 \\ 0 & -2 \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 2 & 0 \\ 1 & 1 \end{bmatrix}$$

```
typedef vector< vector<int> > Matrix;
```

Fig 1.0

The figure above shows two matrixes of three rows and two columns each.

The data in the table can be stored, retrieved, and manipulated using different structural constructs and in the case above a vector of vector. An addition operation is performed on the matrix A and B, providing a Matrix C with similar rows and columns as both Matrix A and B.

I. Matrix Calculator

Matrix calculator (loading from file and saving to file) Add, subtract, multiply, transform, determinant, inverse of the matrix operations needs to be implemented. Matrix in files can have different sizes. The format of matrix file is size (2 numbers divided by semicolon, lines start with #) in the first line and then matrix data divided by semicolons example:

```
#2;2
```

```
1;1
```

```
2;2
```

Program should get following switches from console:

-a input file for A matrix

-b input file for B matrix

-o output file with results of calculation or error message if calculations cannot be performed

-operation 'name' – name of operation to be performed, example „operation add”

II. Analysis of the task

The task focuses on the operations performed on a matrix or matrices. Across different disciplines, a matrix (plural matrices) is a rectangular array or table of numbers or data arranged in rows and columns, which is used to represent a mathematical object or a property of such an object.

A. Data Structure

The size of a matrix is defined by the number of rows and columns it contains.

A matrix with m rows and n columns is called an $m \times n$ matrix.

An $m \times n$ matrix: the m rows are horizontal, and the n columns are vertical.

Each element of a matrix is often denoted by a variable with two subscripts.

$$\begin{array}{c}
 1 \\
 2 \\
 3 \\
 \vdots \\
 m
 \end{array}
 \begin{bmatrix}
 \overset{1}{a_{11}} & \overset{2}{a_{12}} & \dots & \overset{n}{a_{1n}} \\
 a_{21} & a_{22} & \dots & a_{2n} \\
 a_{31} & a_{32} & \dots & a_{3n} \\
 \vdots & \vdots & \vdots & \vdots \\
 a_{m1} & a_{m2} & \dots & a_{mn}
 \end{bmatrix}$$

Fig 2.1

For example, $a_{2,1}$ in fig 2.1 represents the element at the second row and first column of the matrix.

B. Algorithm

The program reads a matrix from file and performs one of the operations on the data read from file. The Add, subtract, multiply, transform, determinant,

transpose and inverse operations can be performed on these matrices in question.

Several of the operations require some underlying condition to be met before execution. The results or errors from a calculation is to be saved to a file.

III. External specification

This is a command line program. The program requires names of input and output files.

Program should get following switches from console:

- a input file for A matrix – “#2;2”
- b input file for B matrix - “#3;3”
- o output file with results of calculation or error message if calculations cannot be performed
- operation 'name' – name of operation to be performed, example “add” or “subtract”.

IV. Internal specification

The program is implemented with structural paradigm. User interface is separated from program's logic.

A. Program overview

The main function accepts 9 variables, the first been the program name and the other 8 serve to input different switches that verifies parameters of the program using an if statement. If the verification is negative, an appropriate message is printed to the file.

In case of positive verification, data are read with read_matrix function. The function opens a file stream, reads numbers from the stream, and inserts

numbers into vector or vector matrix (`std::vector<std::vector<double>>` `matrix_Name`). The `Convert` function accepts the operation name and compares the string values with an Enum class called `operation`. After verification of the operation type, the function returns a value with the right operation type to be executed.

A switch break construct is used switch the returned operation type and is executed when found.

Finally, the program saves the error or result of an operation into an output file in the memory.

B. Description of types and functions

Description of types and functions is moved to the Document attachment named Doxygen documentation.

V. Testing

The program has been tested with various types of files. Incorrect files (with no numbers, numbers in incorrect format) are detected and an error message is printed and saved to file. An empty input file does not cause failure – an empty output file is created. Larger files result in a bad allocation error. The program has no memory leaks.

VI. Conclusions

The program implements a matrix calculation algorithm with a variety of operations. The most challenging tasks is manual memory management without any memory leaks. It is especially difficult when the data are read partially and then an error in data occurs.

For some data the program elaborates incorrect results on some machines. This is caused by various integer representation – some compilers use 2B integers whereas some – 4 B. For large numbers in input files 2B integers are too small.

References

- Enumeration declaration, <https://en.cppreference.com/w/cpp/language/enum>
- Jordi Cortadella, RicardGavaldà, Fernando Orejas, Dept. of Computer Science, UPC. Introduction to Programming (in C++) Multi-dimensional vectors
- Wikipedia contributors. (2022, January 25). Matrix (mathematics). In Wikipedia, The Free Encyclopedia. Retrieved 20:02, February 3, 2022: [https://en.wikipedia.org/w/index.php?title=Matrix_\(mathematics\)&oldid=1067866229](https://en.wikipedia.org/w/index.php?title=Matrix_(mathematics)&oldid=1067866229)

