Part 2a Documentation

Program Description

The program reads the output files of part one as inputs and writes the matrix addition of each unique two matrix combination into output files. Since matrix addition is symmetric, only five choose 2 with repetition, or 15, files are generated.

Important Library Details

- Eigen
    - Library path: the headers for the Eigen library are located in /usr/include/eigen3 on my Linux machine.
    - Library version: I have installed Eigen version 3.4.0.

Marginal Cases

- Invalid inputs:
    - If, for some reason, the outputs of part one have not been generated, an assertion in ReadMatFile ensures the program will end harmlessly.
    - The outputs of part one have already been verified, therefore it is not problematic to assume all inputs are two-dimensional double matrices with correctly labeled dimensions.
    - The WriteMatSumFile wrapper methods for the MatSum methods check for the dimensions of input arrays and perform matrix addition only if the dimensions are identical.
- Invalid computations:
    - All important computations in the Eigen implementation methods were handled by Eigen, and the outputs have been checked.
    - All custom implementation methods simply added two doubles and stored them in the corresponding double element of a result matrix. The outputs have also been checked.

Design Choices

- The MatSum method was split into two implementations:
    - MatSumEigen which uses the built in Eigen matrix addition functionality. I made this implementation for the sake of familiarizing myself with the Eigen library.
    - MatSumCustom which implements a custom matrix addition algorithm. I made this implementation to demonstrate that I understand how matrix addition works.
- vThe potential combinations of matrix additions were hardcoded, as I do not feel implementing a mathematical combination function for matrix addition is worth it.

- The choice of words in the assignment “name_p2a_out12.txt”, “name_p2a_out13.txt”, … etc. as opposed to “name_p2a_out11.txt”, “name_p2a_out12.txt”, … etc. seems to imply that matrices added to themselves might not need to be generated. However, just to be safe, they will be generated anyway.

Pseudocode

```
// Adds two matrices in a custom implementation and returns the sum.  Assumes input matrices
// can be added.
Matrix MatSumCustom(Matrix input_1, Matrix input_2)

// Adds two matrices using Eigen and returns the sum.  Assumes input matrices can be added.
Matrix MatSumEigen(Matrix input_1, Matrix input_2)

// Read the matrix at file_path’s data, create a matrix object with that data, and return the matrix
// object.
Matrix ReadMatFile(string read_file_path)

// Write a matrix mat’s dimensions and data to a file at file_path.
Void WriteMatFile(Matrix mat, string file_path)

// Write the matrix sum of the two input matrices, or an error message, to a file at output_path
// using MatSumCustom.
Void WriteMatSumFileCustom(Matrix input_1, Matrix input_2, string output_filepath)

// Write the matrix sum of the two input matrices, or an error message, to a file at output_path
// using MatSumEigen.
Void WriteMatSumFileEigen(Matrix input_1, Matrix input_2, string output_filepath)

Int main():
        Const string kMat1Path = “../part_one/jhartt_p1_mat1.txt”
        Const string kMat2Path = “../part_one/jhartt_p2_mat1.txt”
        Const string kMat3Path = “../part_one/jhartt_p3_mat1.txt”
        Const string kMat4Path = “../part_one/jhartt_p4_mat1.txt”
        Const string kMat5Path = “../part_one/jhartt_p5_mat1.txt”
        Const Matrix kMat1 = ReadMatFile(kMat1Path)
        Const Matrix kMat2 = ReadMatFile(kMat2Path)
        Const Matrix kMat3 = ReadMatFile(kMat3Path)
        Const Matrix kMat4 = ReadMatFile(kMat4Path)
        Const Matrix kMat5 = ReadMatFile(kMat5Path)

        Const string kOut11Path = “jhartt_p2a_out11.txt”
```

```
        Const string kOut12Path = “jhartt_p2a_out12.txt”
        Const string kOut13Path = “jhartt_p2a_out13.txt”
        Const string kOut14Path = “jhartt_p2a_out14.txt”
        Const string kOut15Path = “jhartt_p2a_out15.txt”
        Const string kOut22Path = “jhartt_p2a_out22.txt”
        Const string kOut23Path = “jhartt_p2a_out23.txt”
        Const string kOut24Path = “jhartt_p2a_out24.txt”
        Const string kOut25Path = “jhartt_p2a_out25.txt”
        Const string kOut33Path = “jhartt_p2a_out33.txt”
        Const string kOut34Path = “jhartt_p2a_out34.txt”
        Const string kOut35Path = “jhartt_p2a_out35.txt”
        Const string kOut44Path = “jhartt_p2a_out44.txt”
        Const string kOut45Path = “jhartt_p2a_out45.txt”
        Const string kOut55Path = “jhartt_p2a_out55.txt”

        WriteMatSumFileCustom(kMat1, kMat1, kOut11Path)
        WriteMatSumFileEigen(kMat1, kMat2, kOut12Path)
        WriteMatSumFileCustom(kMat1, kMat3, kOut13Ppath)
        WriteMatSumFileEigen(kMat1, kMat4, kOut14Path)
        WriteMatSumFileCustom(kMat1, kMat5, kOut15Path)
        WriteMatSumFileEigen(kMat2, kMat2, kOut22Path)
        WriteMatSumFileCustom(kMat2, kMat3, kOut23Path)
        WriteMatSumFileEigen(kMat2, kMat4, kOut24Path)
        WriteMatSumFileCustom(kMat2, kMat5, kOut25Path)
        WriteMatSumFileEigen(kMat3, kMat3, kOut33Path)
        WriteMatSumFileCustom(kMat3, kMat4, kOut34Path)
        WriteMatSumFileEigen(kMat3, kMat5, kOut35Path)
        WriteMatSumFileCustom(kMat4, kMat4, kOut44Path)
        WriteMatSumFileEigen(kMat4, kMat5, kOut45Path)
        WriteMatSumFileCustom(kMat5, kMat5, kOut55Path)

        Return 0

Matrix MatSumCustom(Matrix input_1, Matrix input_2):
        Matrix out_mat(input_1.rows(), input_1.cols())

        Iterate through row indices:
                Iterate through column indices:
                        out_mat(row, col) = input_1(row, col) + input_2(row, col)

        Return out_mat

Matrix ReadMatFile(string read_file_path):
        Ifstream read_file;
```

```
read_file.open(read_file_path)

Assert read_file exists

Int rows = toDouble(readFile.nextWord())
Int cols = toDouble(readFile.nextWord())

Matrix out_mat(rows, cols);

String raw_element = “”;
For (int row = 0; row < rows; row++):
        For (int col = 0; col < cols; col++):
                out_mat(row, col) = toDouble(readFile.nextWord())

Return out_mat
```