Part 1 Documentation

Program Description

The program performs row-major and column-major operations to fill multiple matrices with values before writing their dimensions and contents into a corresponding file each. The matrices' size is often determined by the number of letters in my first and last names, but sometimes not.

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: there were no inputs for part one, therefore input validation was omitted from the program.
- Invalid computations: these are not a concern for the following reasons: all important computations were handled by the C++ library Eigen, and the results were checked in the test cases.

Design Choices

- A doubles matrix was used for all five matrices since the usage of int matrices is not intended with the Eigen library - although it is possible.
- The WriteMatFile method was made because there was enough code repetition when writing matrices to file that the process warranted its own method. It will also write the dimensions of the array on the first line of the file to aid with parsing later in the assignment.

Pseudocode

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

Int main():
    Const string kFirstName = "Jacob"
    Const string kLastName = "Hartt"
    Const int kFirstNameLen = kFirstName.length()
    Const int kLastNameLen = kLastName.length()
```

```
Matrix mat_1(kLastNameLen, kFirstNameLen)
Double counter_1 = 1
Iterate across mat_1 in row-major order:
        mat_1(row, col) = counter_1
        counter_1++
WriteMatFile(mat_1, “jhartt_p1_mat1.txt”)

Matrix mat_2(kFirstNameLen, kLastNameLen)
Double counter_2 = 3
Iterate across mat_2 in column-major order:
        mat_2(row, col) = counter_2
        Counter_2 += 5
WriteMatFile(mat_2, “jhartt_p1_mat2.txt”)

Matrix mat_3(kFirstNameLen, kLastNameLen)
Double counter_3 = 0.33
Iterate across mat_3 in column-major order:
        mat_3(row, col) = counter_3
        Counter_3 += 0.6
WriteMatFile(mat_3, “jhartt_p1_mat3.txt”)

Matrix mat_4(5, 6)
Double counter_4 = 3
Iterate across mat_4 in column-major order:
        mat_4(row, col) = counter_4
        Counter_4 += 2
WriteMatFile(mat_4, “jhartt_p1_mat4.txt”)

Matrix mat_5(6, 5)
Double counter_5 = -10
Iterate across mat_5 in row-major order:
        mat_5(row, col) = counter_5
        counter_5++
WriteMatFile(mat_5, “jhartt_p1_mat5.txt”)

Return 0
```