CSCI 2500 — Computer Organization
Homework 01 (document version 1.0) — Due September 12, 2022
Let's get Matrix Multiplyin'

- This homework is due by 11:59:59 pm EDT on the above date via a Submitty gradeable.
- This homework is to be completed **individually**. Do not share your code with anyone else.
- Homework assignments are available approximately seven calendar days before they are due. Plan to start each homework early. You can ask questions during office hours, in the Submitty forum, and during your lab session.

For this assignment, you will write code for multiplication of two non-square matrices. The purpose of this assignment is to give you experience with C programming, with special emphasis on pointers, memory allocation, and I/O.

For a tutorial on basic matrix multiplication (if you've forgotten), please goto the following:

- https://mathworld.wolfram.com/MatrixMultiplication.html

Here you will read into memory two non-square ($N \times M$, $M \times K$) matrices of doubles, and output their matrix multiplication product. You'll need to add functionality for:

1. Allocating space for each matrix in memory.
2. Reading the matrices into memory.
3. Printing the contents of each matrix.
4. Multiplying the matrices and storing the result.
5. Freeing the allocated memory space.

See the code template (`hw01.c`) for more detailed notes on what code you'll be adding and where you'll be adding it. Also, see the template notes along with the given test inputs (`test1.mat`, `test2.mat`) and output (`output.txt`) for formatting expectations on inputs and outputs. In addition, you'll need to do error checking for several things. Listed below are the specific you'll be checking for and the associated error messages you'll be outputting (just output these messages to STDOUT):

1. If you're unable to open the input matrix file (e.g., because it doesn't exist):
   - `mm_read:  failed to open file.`
2. If the input matrix file is empty:
   - `mm_read:  failed to read from file.`
3. If there are no row or column sizes in the input matrix file:

- `mm_read: failed to read matrix dimensions.`

4. If the matrix file doesn't contain enough values:
   - `mm_read: failed to read matrix values.`
5. If allocating space for a matrix fails, possibly from an invalid matrix dimension:
   - `mm_alloc: allocation failed.`
6. If matrix multiplication is impossible because of a dimension mismatch between the two inputs:
   - `mm_matrix_mult: dimension mismatch between matrices.`

If any of these errors are encountered, the program should immediately terminate. Note that for several of these possible errors, you won't need to add any explicit program logic – e.g., if your first call to `fgets()` on a file returns `NULL`, you're allowed to assume that means the file is empty. You don't need to make an additional function call to explicitly check the file size.

1 Grading Criteria

1. Correctness: 60%
   - Computes and outputs result matrix correctly across 6 tests.
2. Error handling: 30%
   - Correctly checks for and outputs messages for the above 6 errors.
3. TA checks: 10%
   - Allocating and freeing is done correctly. No memory leaks.
   - All requirements are satisfied. No modifications to `main`.