CS410C FALL 2019 DUE: 11/15/2019
LAB 15L Submit code: **15L**

Problem

The purpose of this assignment is to give you practice with using 2D Arrays for Matrix operations. A matrix is a mathematical object with a rectangular arrangement of numbers and therefore a 2-D array becomes the perfect candidate to represent a matrix.

Let's start with matrix **A** of a certain size (based on the input data file). ***You can assume that the maximum size of any matrix in this program is going to be 25 x 25.***

Write a program that inputs the matrix A from a data file. The first line of this file will contain the matrix dimensions. So for a 10 x 5 matrix, this line would be:

10 5
Subsequent lines will be the rows of the matrix (in the above case there will be exactly 10 lines of data that follow, each line containing 5 values). Look at the sample input files first to understand how the data files are structured.

First display the matrix you just read. Next find the transpose of this matrix $\mathbf{A^t}$. The transpose of a matrix is a new matrix whose rows are the columns of the original. (This makes the columns of the new matrix the rows of the original). Now display the matrix $\mathbf{A^t}$, which would be a 5 x 10 matrix if you started with a 10 x 5 matrix.

Next we want to multiply the transpose matrix $\mathbf{A^t}$ with the original matrix **A** to create $\mathbf{A^t A}$. The new matrix (product matrix) will end up being a square matrix!

You can use integer arrays for both matrices, since the input data files will only contain integer data.

Input

The input data is coming from a data files whose name is supplied by the user.

Output

The output must first print out the starting matrix **A** that is read from the data file. Then it must print the transpose matrix $\mathbf{A^t}$. You must be able to use the same function print_matrix() to print each matrix at different times (in other words you must not write code every time to print the contents of a matrix, but just use this function with the right arguments).

Requirements

- You must use at least the following functions (all of them must be commented):
    - Function read_data() which reads information into the 2-D array from the data file. It takes 4 arguments, the data file name (character string array), the array that needs to be filled, the actual number of rows of the input matrix, the actual number of columns of the input matrix (the number of rows and number of columns must be output arguments).

- Function print_matrix() which prints a matrix. It takes 3 arguments, the array that needs to be printed, the number of rows, and the number of columns (all of them are input arguments).
- Function compute_transpose_matrix() which computes the transpose matrix. It takes 4 arguments, the array holding the first matrix, number of rows of the first matrix, number of columns of the first matrix, a resulting array which is the transpose of the first matrix.
- Function matrix_multiply() which computes the product of two matrices. It takes 6 arguments, the first matrix (array), the second matrix (array), the number of rows of the first matrix, the number of columns of the first matrix, the number of columns of the second matrix, the resulting product matrix (array). This set of arguments will make the function generic enough so that it can be used to multiply any two matrices of the proper sizes.

- The program must be split into multiple files:
  - The main must be the only function in one code file (.c)
  - All the other functions should be in a second code file (.c)
  - The other functions should be forward declared (prototype declarations) in a header file (.h) and included in both code files
  - **Be sure to submit all three files!**

- All function definitions must have a block comment describing what they do.

- Work on the program using the modularity of the functions, for example write a program that reads the data and prints it out. Once you have these working you can move on to the other functions

- As always test your program carefully and follow good programming style.

Submission

Submit this assignment with the code 15L followed by the names of all 3 files:
For example:

```
submit 15L matrix.c functions.c header.h
```