C **Programming Project #2 - Matrix**

Write a program to create *dynamically allocated* matrices and perform simple operations on them. You should create a type called matrix_t that holds the dimensions of the matrix and a pointer to where the data is stored. Your matrix structure & functions should be defined in a separate file (matrix.c and matrix.h) and you should write a C program called project2.c that tests your code.

Your program should take 2 command line parameters which are filenames of files in the format below that define the values of a matrix. The first line indicates the number of rows and columns in the matrix.

```
2 3
2 4 6
8 10 12
```

Include the following functions. For functions that return a pointer to a matrix you should allocate a *new* matrix inside the function. *Do not modify the function signatures or your code will not test sucessfully*.

```
/* Create a new matrix */
matrix_t * new_matrix(int rows, int cols)

/* Neatly print the matrix */
void print_matrix(const matrix_t * matrix)

/* Get/Set the value at row, col */
int get_value(const matrix_t * matrix, int row, int col)
void set_value(matrix_t * matrix, int row, int col, int value)

/* Return the sum of two matrices */
matrix_t * add(const matrix_t * m1, const matrix_t * m2)

/* Return the transpose of a matrix */
matrix_t * transpose(const matrix_t * matrix)
```

Note that the above only describes the *minimum* requirements for the assignment. Feel free to add any data or functions you find helpful, but only the above will be tested for your grade. Assume the matrix only will hold integers. You should only allocate enough memory for the matrix. You should exercise good C coding style and your code should be clean, clear, and succinct.

Submission:

- Write a makefile for your program that generates an executable program called matrix. Include a "make clean" option. Your submission will be tested with the following sequence of commands:

```
$ make clean
$ make
$ ./matrix matrix1.txt matrix2.txt
```

- Your code should build with no warnings using gcc and the -ansi -Wall -pedantic flags. Each warning, no matter how minor, will result in a 5% reduction in your grade.
- Complete the submission document with all the requested information.
- Zip the directory that contains your project and the makefile. Your project will be tested automatically so be sure that you include **all code** required to build your program.
- Submit the document and your code **separately** to Canvas. Make sure you follow all directions.

Grading Rubric:

1. Matrix structure - 20 points
2. Matrix functions - 50 points
3. Valid output & error handling – 20 points
4. Submission - 10 points