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. You should write functions to dynamically allocate the matrix, display it, add two matrices, and scalar multiplication. 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.

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 parameters.

```
/* 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 clear and succinct.

Submission:

- Write a makefile for your program. Include a "make clean" option. Your submission will be tested with "make clean" then "make". Your makefile should ensure that you are building with the -ansi, -pedantic, and -Wall flags. Your code should build with no warnings using gcc. Each warning, no matter how minor, will result in a 5% reduction in your grade.
- Create a document (.doc, .txt, or .pdf) with the following:
  1. A brief overview of your program's structure
  2. Your matrix struct and your new_matrix & add functions
  3. Sample output (please be clear & concise)
- 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