A. Linear Algebra Functions:

a. Write a function, *MatVecMult*, to calculate matrix vector multiplication of a matrix to a vector:

```
double* MatVecMult(double **mat,int n1, int n2, double const *vec, int k);
```

i. It takes in 1 matrix, 1 vector, and *returns* their product (another vector)
ii. You need to check for dimensions to make sure their product exists.
    1. If no, return a nullptr
    2. Otherwise
        a. Allocate a dynamic double array of appropriate size
        b. make your calculation and return the vector back
iii. Test your code with the following matrix and vectors:
    1. $m = \begin{pmatrix} -1 & 4.5 & 6.2 \\ 2 & -3.4 & -2 \end{pmatrix}$
    2. $v_1 = \begin{pmatrix} 7 \\ -1.2 \end{pmatrix}$, $v_2 = \begin{pmatrix} 1 \\ 2 \\ 1.5 \end{pmatrix}$

** You need to manage the dynamic array returned from the MatVecMult function call*

Write the result to an output file

b. Write a ***recursive*** function, *det*, to calculate the determinant of a given square matrix:

```
double det(double **mat, int n);
```

Test your function with this matrix:

$$\begin{bmatrix} 2 & -3 & 1 \\ 2 & 0 & -1 \\ 1 & 4 & 5 \end{bmatrix}$$

Write the result to an output file

c. Write a function, MatrixMult, that multiplies two matrices. The resulting matrix is one of the parameters:

```
bool MatrixMult(double **mat1, int row1, int col1,
                double **mat2, int row2, int col2,
                double **prod);
```

i. You need to check the given matrix dimensions to make sure their product exists.
    1. If product does not exist, return false immediately
    2. Otherwise, make your calculation to fill the product matrix, *prod*, which has the dimension of row1 x col2. Assume the caller has already allocated the necessary memory for *prod*. At the completion of the calculation, the function returns true.
ii. Test your function with following matrices:
    1. $m_1 = \begin{bmatrix} 2 & -3 & 1 \\ 2 & 0 & -1 \\ 1 & 4 & 5 \end{bmatrix}$, $m_2 = \begin{pmatrix} -1 & 4.5 & 6.2 \\ 2 & -3.4 & -2 \end{pmatrix}$

2. $m_1 = \begin{pmatrix} -1 & 4.5 & 6.2 \\ 2 & -3.4 & -2 \end{pmatrix}$, $m_2 = \begin{bmatrix} 2 & -3 & 1 \\ 2 & 0 & -1 \\ 1 & 4 & 5 \end{bmatrix}$

Write the result to an output file

B. Compile your source code, and run your programs. Submit
   1. Your source-code
   2. The output files

Happy coding!