It’s hard enough to find an error in your code when you’re looking for it; it’s even harder when you’ve assumed your code is error-free.
~ *Steve McConnell*

CS1010 Programming Methodology

Week 7: Two-dimensional Arrays (Selected Answers)

4. To multiply two matrices A and B, the number of columns in A, let’s call it *n*, must be the same as the number of rows in B. The resulting matrix has the same number of rows as A and number of columns as B. Hence, multiplying a *k*×*n* matrix with an *n*×*p* matrix gives a *k*×*p* product matrix.

   To compute C = A × B, where A, B, C are matrices,

   $$C_{i,j} = (A_{i,0} \times B_{0,j}) + (A_{i,1} \times B_{1,j}) + \ldots + (A_{i,n\text{-}1} \times B_{n\text{-}1,j})$$

   Two examples are shown here:

   $$\begin{pmatrix} 1 & 2 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} -1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 2 & -1 \end{pmatrix} = \begin{pmatrix} 3 & 2 & 0 \\ 2 & 3 & -1 \\ -1 & 2 & -1 \end{pmatrix}$$

   $$\begin{pmatrix} 2 & 1 & 3 & 2 \\ 3 & 0 & 2 & 1 \end{pmatrix} \times \begin{pmatrix} 3 & 2 & 1 \\ 2 & 2 & 3 \\ 1 & 3 & 0 \\ 2 & 1 & 3 \end{pmatrix} = \begin{pmatrix} 15 & 17 & 11 \\ 13 & 13 & 6 \end{pmatrix}$$

   Write a function to perform the product of two matrices. You may assume that a matrix has at most 10 rows and 10 columns.

   Answer:

   See **MatrixOps_sol.c**

5. A **square matrix** is a two-dimensional array where the number of rows and columns are the same. Write a program **square_matrix.c** to read in values for an *n*×*n* square matrix containing integer values, and check whether the matrix is (a) a diagonal matrix, or (b) an upper-triangular matrix.

   A **diagonal matrix** is a square matrix in which the elements outside the main diagonal (⇲) are all zeroes, for example:

   $$\begin{bmatrix} 3 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & -2 \end{bmatrix} \qquad \begin{bmatrix} 12 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -5 & 0 \\ 0 & 0 & 0 & 7 \end{bmatrix}$$

   An **upper triangular matrix** (or right triangular matrix) is a square matrix U of the form:

$$U_{ij} = \begin{cases} a_{ij} & \text{for } i \le j \\ 0 & \text{for } i > j. \end{cases}$$

Written explicitly,

$$\mathrm{U} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix}$$

Note that a diagonal matrix is also an upper triangular matrix.

A sample run is shown below. The first line contains a single integer indicating the size of the square matrix, *n*. The next *n*×*n* values are the elements of the matrix. The output is in bold. You may assume that the matrix contains at most 10 rows and 10 columns.

```
5
2 -1 3 4 1
0 7 5 -2 0
0 0 6 0 4
0 0 0 0 8
0 0 0 0 2
Matrix read:
   2  -1   3   4   1
   0   7   5  -2   0
   0   0   6   0   4
   0   0   0   0   8
   0   0   0   0   2
Matrix is not a diagonal matrix.
Matrix is an upper triangular matrix.
```

You may download the incomplete program **square_matrix.c** from the module website (under "CA" → "Discussion"), or copy it over to your directory with this command:

```
cp  ~cs1010/discussion/prog/week7/square_matrix.c  .
```

Complete the program.

Answer:

See **square_matrix_complete.c**