University of Victoria
Department of Computer Science
CSC 110 Fundamentals of Programming
ASSIGNMENT 5

DUE: Friday, March 9, 2018 before 11:50 pm – By submission on ConneX only

Output: MUST Match Sample Output – Exactly

Indentation: MUST be Exactly 4 spaces or 1 tab

Input: Only 1 Scanner (new) object can be used:

- It is connected to the file `MatrixIn.txt`

How to hand in your work

Submit the file (called `Matrix.java`) that completely answers part II (below) through the Assignment #5 link on the CSC 110 conneX site. Please make sure you follow all the required steps for submission (including the final confirmation of your submission). If you have received an email confirming your submission, then you know it is submitted, if not, wait a few moments, then re-check what you have done and re-submit.

Learning Outcomes

When you have completed this assignment, you should be able to:

- Use file input.
- Produce file output.
- Populate 2-dimensional arrays
- Pass 2-dimensional arrays as parameters to methods
- Return 2-dimensional arrays from methods.
- Output 2-dimensional arrays.

Part 1: Problems from the Textbook:

Complete the *Chapter 7 Self-Check Problems* and compare your answers to the author's answers.

Part II: Matrix Operations

Background:

A matrix is a 2-dimensional mathematical structure used in Linear Algebra. We are going to write a Java program that inputs Matrices and performs some simply arithmetic using them.

Consider the following 2-by-2 matrices:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 1 \\ -1 & -2 \end{bmatrix}$$

Many of the familiar laws for arithmetic of numbers do hold in the case of matrix arithmetic.

The **sum** of two matrices A and B is only defined when A and B have the same dimensions. To add A and B, add their corresponding entries:

$$A + B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ -1 & -2 \end{bmatrix} = \begin{bmatrix} 1+1 & 2+1 \\ 3-1 & 4-2 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 2 & 2 \end{bmatrix}$$

Likewise, the **difference** of two matrices A and B is only defined when A and B have the same dimensions. To subtract B from A, subtract their corresponding entries:

$$A - B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} - \begin{bmatrix} 1 & 1 \\ -1 & -2 \end{bmatrix} = \begin{bmatrix} 1-1 & 2-1 \\ 3+1 & 4+2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 4 & 6 \end{bmatrix}$$

The **product** of a scalar (regular number), c, and a matrix A is computed by multiplying each entry of A by c:

$$3A = 3\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 3*1 & 3*2 \\ 3*3 & 3*4 \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ 9 & 12 \end{bmatrix}$$

The **dot product** of a two matrices A and D is a little bit tricky. First, it requires that the number of columns of A matches the number of rows of D. Then values in the rows of A are multiplied by values in the columns of D and summed:

$$A \bullet D = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \bullet \begin{bmatrix} 4 & 2 & 2 \\ 1 & 1 & 3 \end{bmatrix} = \begin{bmatrix} 1\times4+2\times1 & 1\times2+2\times1 & 1\times2+2\times3 \\ 3\times4+4\times1 & 3\times2+4\times1 & 3\times2+4\times3 \end{bmatrix}$$
$$= \begin{bmatrix} 6 & 4 & 8 \\ 16 & 10 & 18 \end{bmatrix}$$

Assignment Instructions:

For this assignment you will input numbers from a file (called `MatrixIn.txt`) that will be used to create matrices A, B, scalar number c, and matrix D. Then the program will calculate and output (to a file called `MatrixResult.txt`): A+B, A-B, cA and A•D. The input file will contain data for all 4 (A, B, c and D) in order and each matrix will be preceded by two integers, indicating the dimension of the matrix. The numbers of the matrix, which you can assume will be integers, will be in the file in row-major order.

If, for example, the input file has the following data:

```
2 2 1 2 3 4
2 2 1 1 -1 -2
3
2 3 4 2 2 1 1 3
```

then A, B, c and D would be:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, B = \begin{bmatrix} 1 & 1 \\ -1 & -2 \end{bmatrix}, c = 3, D = \begin{bmatrix} 4 & 2 & 2 \\ 1 & 1 & 3 \end{bmatrix}$$

And the program's output would be:

```
MATRIX ARITHMETIC

Inputting Matrices A, B and D and scalar c . . . . . .
. . . . Writing result to file: MatrixResult.txt
```

And the file `MatrixResult.txt` would contain:

```
A  =
           1     2
           3     4

B =
           1     1
           -1    -2

c =   3

D =
           4     2     2
           1     1     3

A + B =
           2     3
           2     2

A - B =
           0     1
           4     6

cA =
           3     6
           9     12

A dot D     =
           6     4     8
           16    10    18
```

Your program must make use of the following method signatures:

```
public static int[][] getMatrix(Scanner inFile)
public static int [][] add(int[][] one, int[][] theOther)
public static int [][] subtract(int[][] one, int[][] theOther)
public static int [][] scalarMultiply(int scalar, int[][] matrix)
public static int [][]dot(int[][] one, int[][] theOther)
public static void outputMatrix(int[][] matrix, PrintStream outFile)
```

Marking

Your mark will be based on:

- Your code compiling and running.
- Your code producing output screen and output file exactly as specified; including calculating the matrix results correctly.
- Your code follows the code convention guidelines.
- Your code uses the specified parameterized methods to ensure there is little or no code redundancy.