COSC 237 – ASSIGNMENT 01

Submission Instruction

Failure to follow the below instructions will result in a 20% penalty.

1. Do each programming problem in a single file with the name format: ***"<your_last_name>_HW_XX_YY.java"*** where ***XX*** is the assignment number and ***YY*** is the problem number. For instance, for assignment 01 and student name "David Smith", the filenames would be "Smith_HW_01_01.java", "Smith_HW_01_02.java", etc for questions #1, #2, and so on.
2. When you are done, create a folder named ***"<first_name>_<last_name>_HW_XX"*** and put all your ***.java*** and input files or test files (NO ***.o*** or ***.class*** files) in this folder. For example: folder name *"David_Smith_HW_01"* for student name "David Smith". No *.class* files or test files should be included.
3. Finally, create a ZIP file of this folder with the same name (i.e., "David_Smith_HW_01.zip") and submit it over Blackboard. If you have an issue, change the file extension to ***.txt*** (i.e., change "David_Smith_HW_01.zip" to "David_Smith_HW_01.txt") and resubmit.

For this assignment, you will work with a partner and both of you should contribute significantly to the solution for every question. I suggest that both of you do the programs alone, pass your version to your partner, criticize the version written by your partner, and present a solution that has the best features. You should submit only one copy of the homework with both your names on it. If you do discuss it with someone else, you should say this in the preamble of the homework. You should only submit work that is completely your own and you should be able to explain all your homework to me. Make sure you handle **invalid input, including type mismatch** - input validation is a requirement! Focus on good design, good style, and, most importantly, reuse of code. Start early and remember the rules about late assignments/syllabus → **no late assignments!**

Question 1 [40pts]. Design a Java program to implement matrix arithmetic for square matrices (same number of rows and columns). You will need to use your math textbook or the Internet to review operations with matrices.
Make sure that your program is calling methods to perform (at least) the following operations:
Generate: Generate a matrix with values 1 - 10
Addition: See http://mathworld.wolfram.com/MatrixAddition.html
Subtraction: Figure it out!
Multiplication
-- Multiply two matrices: See http://mathworld.wolfram.com/MatrixMultiplication.html
-- Multiply a matrix by a constant: Equivalent to multiplication of each element by that constant.
Transposition: See http://mathworld.wolfram.com/Transpose.html
Matrix Trace: See http://mathworld.wolfram.com/MatrixTrace.html
Print (use printf)
Have another method called **menu()**, that should list the options for matrix operations for which you designed the previous functions, and an option to exit the program. Your program should loop until the user chooses to exit. In this loop you are required to use a **switch** statement for all possible cases.

SAMPLE OUTPUT:

```
Your options are:
-----------------
        1) Add 2 matrices
        2) Subtract 2 matrices
        3) Multiply 2 matrices
        4) Multiply matrix by a constant
        5) Transpose matrix
        6) Matrix trace
        0) EXIT
Please enter your option: 2

Enter the size of square matrices: 4
First matrix is:
   3   6   7   4
  10  10   2   5
  10   5   7   6
   3   8   7   2
Second matrix is:
   9   7   6   4
   7   5   3   4
   2   9   6   7
   2   2  10   9
The resulting matrix is:
  -6  -1   1   0
   3   5  -1   1
   8  -4   1  -1
   1   6  -3  -7
                        Command number 1 completed.

Your options are:
-----------------
        1) Add 2 matrices
        2) Subtract 2 matrices
        3) Multiply 2 matrices
        4) Multiply matrix by a constant
        5) Transpose matrix
        6) Matrix trace
        0) EXIT
Please enter your option: 1

Enter the size of square matrices: 3
First matrix is:
   6   7   6
   2   1   6
   7   3   3
Second matrix is:
   3  10   7
   4   3   3
   9   9   2
The resulting matrix is:
   9  17  13
   6   4   9
  16  12   5
                        Command number 2 completed.

Your options are:
-----------------
        1) Add 2 matrices
        2) Subtract 2 matrices
        3) Multiply 2 matrices
        4) Multiply matrix by a constant
        5) Transpose matrix
        6) Matrix trace
        0) EXIT
Please enter your option: 4

Enter the size of the square matrix: 5
Enter the multiplication constant: 4
The matrix is:
   3   9   2   5  10
  10   9   2   4   5
   5   3   1   2   7
   3   1   8  10  10
   1   9   9   8   7
The original matrix multiplied by 4 is:
  12  36   8  20  40
  40  36   8  16  20
  20  12   4   8  28
  12   4  32  40  40
   4  36  36  32  28
                        Command number 3 completed.

Your options are:
-----------------
        1) Add 2 matrices
        2) Subtract 2 matrices
        3) Multiply 2 matrices
        4) Multiply matrix by a constant
        5) Transpose matrix
        6) Matrix trace
        0) EXIT
Please enter your option: 3

Enter the size of square matrices: 3
First matrix is:
   6   9  10
   4   7   8
  10   8   7
Second matrix is:
   4  10   6
   3   8   4
   8   2   5
The resulting matrix is:
 131 152 122
 101 112  92
 120 178 127
                        Command number 4 completed.

Your options are:
-----------------
        1) Add 2 matrices
        2) Subtract 2 matrices
        3) Multiply 2 matrices
        4) Multiply matrix by a constant
        5) Transpose matrix
        6) Matrix trace
        0) EXIT
Please enter your option: 5

Enter the size of the square matrix: 7
The matrix is:
   4   9   4   3   7   2   6
   2   2  10   6   3   9   9
   7   4   2   8   9   9   2
   9   1   7   5  10  10   5
   8   6   8   7   7   1   8
   3   2   9   8  10   6   2
   6  10   6  10   6   6   5
The transposed matrix is:
   4   2   7   9   8   3   6
   9   2   4   1   6   2  10
   4  10   2   7   8   9   6
   3   6   8   5   7   8  10
   7   3   9  10   7  10   6
   2   9   9  10   1   6   6
   6   9   2   5   8   2   5
                        Command number 5 completed.

Your options are:
-----------------
        1) Add 2 matrices
        2) Subtract 2 matrices
       3) Multiply 2 matrices
        4) Multiply matrix by a constant
        5) Transpose matrix
        6) Matrix trace
        0) EXIT
Please enter your option: 6

Enter the size of the square matrix: 5
The matrix is:
   2   2   5   9   4
   7   3   1   5   4
   8   9  10   4   4
   4   6  10   4   8
   6  10  10   9   8
The trace for this matrix is: 27
                        Command number 6 completed.

Your options are:
-----------------
        1) Add 2 matrices
        2) Subtract 2 matrices
        3) Multiply 2 matrices
        4) Multiply matrix by a constant
        5) Transpose matrix
        6) Matrix trace
        0) EXIT
Please enter your option: 0
Testing completed.
```

Question 2 [60pts]. A magic square is an arrangement of the numbers from 1 to n^2 in an n x n matrix, with each number occurring exactly once, and such that the sum of the entries of any row, any column, or any main diagonal is the same.(**NOTE**: This sum must be $n*(n^2+1)/2$)

The simplest magic square is the 1x1 magic square whose only entry is the number 1:

1

The next simplest is the 3x3 magic square (all rows, columns, and diagonals add up to 15):

8	1	6
3	5	7
4	9	2

The following is a 5 x 5 magic square (all rows, columns, and diagonals add up to 65):

17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

In order to construct an n x n magic square for any odd integer n, you are going to use the following procedure, called the Siamese method:

First, place 1 in the middle of the top row. Then, after integer k has been placed, move up one row and one column to the right to place the next integer k + 1, unless one of the following occurs:

1. If a move takes you above the top row in the jth column, move to the bottom of the jth column and place the integer k + 1 there.
2. If a move takes you outside to the right of the square in the ith row, place k + 1 in the ith row at the left side.
3. If a move takes you to an already filled square or if you move out of the square at the upper right-hand corner, place k + 1 immediately below k.

Click here for a flash animation (will show you how to fill the cells of a 5x5 magic square.)

Your Java program should call a method to generate the magic square and a method to print it. In addition, print the "magic number" that the diagonals, the rows, and the columns add up to. Also, your program should work in a loop, for more than just one magic square. To design the interface, look at the following sample output.

SAMPLE OUTPUT:

```
Enter the size of magic square (positive & odd): 4
INPUT ERROR!!! Invalid size.
Enter the size of magic square (positive & odd): 3
The magic square with size = 3 is:
    8    1    6
    3    5    7
    4    9    2
The 3x3 magic square adds up to 15

Do you want to continue (Y/N): Y
Enter the size of magic square (positive & odd): 5
The magic square with size = 5 is:

   17   24    1    8   15
   23    5    7   14   16
    4    6   13   20   22
   10   12   19   21    3
   11   18   25    2    9
The 5x5 magic square adds up to 65

Do you want to continue (Y/N): y
Enter the size of magic square (positive & odd): 7
The magic square with size = 7 is:

   30   39   48    1   10   19   28
   38   47    7    9   18   27   29
   46    6    8   17   26   35   37
    5   14   16   25   34   36   45
   13   15   24   33   42   44    4
   21   23   32   41   43    3   12
   22   31   40   49    2   11   20
The 7x7 magic square adds up to 175

Do you want to continue (Y/N): n.
```