

CSE 465/565
Spring 2020
Homework #4

Notes:

- *Submit a zip file containing your completed files: MM.cs, mm.py, LTMatrix.cs, and lmat.py*
 - *Some sample code and data has been provided for you in the directory: ~zmudam/CSE465-565/HW4S20*
1. **(50/35 points)** Write equivalent Python 3 and C# class definitions to represent a lower triangular matrix. A lower triangular is a square matrix whose elements above the diagonal are defined to be 0. Thus, the matrix element $M_{r,c}$ can be non-zero only if $r \geq c$. The following is an example lower triangular matrix of size $N=4$. Row and column indexing starts at 0.

Row 0	1	0	0	0
Row 1	7	-8	0	0
Row 2	0	8	10	0
Row 3	9	7	3	-9

While it is possible to use a regular 2D array to represent an Upper Triangular matrix, doing so is wasteful with memory. It is preferable to store only the non-zero elements in a minimally-sized 1D array, such as:

2D indices	[0,0]	[1,0]	[1,1]	[2,0]	[2,1]	[2,2]	[3,0]	[3,1]	[3,2]	[3,3]
1D indices	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
Array values	1	7	-8	0	8	10	9	7	3	-9

You will write a class definition with the following methods:

- **Constructor** that takes one integer parameter, N . This constructor will, at a minimum, allocate a 1D array/list. capable of storing a $N \times N$ lower triangular matrix. An exception is thrown if any of the input values are nonsensical.
 - **Length** that returns the length of the 1D array representing the matrix.
 - **Size** that returns the width and height of the matrix, N .
 - **OneDArray** returns the 1D array that stores the element of the lower triangular matrix.
 - **TwoDToOneDIndex** that takes integers: row and column and returns the corresponding location in the 1D matrix. An exception is thrown if any of the values are nonsensical. To receive maximum points, this method should be $O(1)$; however, if you are unable to create the correct equation, you are free to develop a non-constant-time method, albeit with reduced points.
 - **Set** that takes integers: row, column, and V . It will set the matrix's value at the specific row/col position to V . An exception is thrown if any of the values are nonsensical.
 - **Get** that takes integers: row, column. It will return the matrix's value at the specific row/col. An exception is thrown if any of the values are nonsensical.
 - **+** that will take two LTMatrix and return an LTMatrix that represents the summation of the two matrices. An exception is thrown if the two matrices are not the same size.
2. **(0/15 points)** Add a class definition to the file you created in the previous problem. This class will represent $N \times N \times N$ lower triangular matrices. Lower triangular 3D matrices are non-zero for indices $M_{r,c,d}$ iff $r \geq c \geq d$. Like in the 2D case, elements in the matrix will be stored in order. For a $3 \times 3 \times 3$ lower triangular matrix, the values will be stored as follows:

$$M_{0,0,0}, M_{1,0,0}, M_{1,1,0}, M_{1,1,1}, M_{2,0,0}, M_{2,1,0}, M_{2,1,1}, M_{2,2,0}, M_{2,2,1}, M_{2,2,2}$$

For simplicity, the class will have only a subset of the methods that were defined in the 2D case:

- **Constructor** that takes one integer parameter, N. This constructor will allocate a 1D array/list. capable of storing a NxNxN lower triangular matrix. An exception is thrown if any of the values are nonsensical.
- **Length** that returns the length of the 1D array representing the matrix.
- **Size** that returns the width of the matrix, N.
- **ThreeDToOneDIndex** that takes integers: row, column, and depth and then returns the corresponding location in the 1D matrix. An exception is thrown if any of the values are nonsensical.

Notes:

- Sample drivers have been provided to you in Python and C# have been provided to you. Use these to determine the parameters, method names, etc. In the end, your code should be compatible with the provided testers. A comment section shows what the results should be.
- Here is the C# commands that will be used to test your code:
 - `$ mcs HW4MatTest.cs LTMatrix.cs`
 - `$ mono HW4MatTest.exe`
- Here is the Python command that will be used to test your code:
 - `$ python3 hw4mattest.py`

3. (50/50 points) Write equivalent mail merge programs, one in C# and one in Python 3.

Your mail merge programs will each take two inputs from command line arguments:

- The name of a file that contains CSV data
- The name of a file that contains the templated document

Here is an example of how your programs will be executed:

- `python3 mm.py HW1465.csv HW1465.tmp`
- `mcs MM.cs`
`mono MM.exe HW1465.csv HW1465.tmp`

Here are example files.

HW1465.csv

```
NAME,ID,COURSE,HOURS,LATE,DEDUCTION,P1,P1COMMENTS,P2,P2COMMENTS,SUBTOTAL,TOTAL
Steve Smith,smiths,465,1,5,50,Excellent work,50,Excellent work,100,95
Mark Watson,watsonm,465,ontime,0,50,Excellent work,40,Does not perform error handling,90,90
Kenny Briddle,briddlek,465,ontime,0,40,Does not perform error handling,40,Does not perform error handling,80,80
Bill Haygood,haygoodb,565,ontime,0,0,P1 not submitted,0,P2 not submitted,0,0
```

HW1465.txt

Name: <<NAME>> (<<COURSE>>)
UniqueID: <<ID>>
Total: <<TOTAL>>/100
Subtotal: <<SUBTOTAL>>
Hours Late: <<HOURSLATE>>
Late deduction: <<LATEDEDUCTION>>
<<P1>>/50 Problem #1: <<P1COMMENTS>>
<<P2>>/50 Problem #2: <<P2COMMENTS>>

When run, your program should produce one output file for each student in the file. The file should be named using the ID column, which is known to exist. In this case, the four files should be:

smiths.txt, watsonm.txt, briddlek.txt, and haygoodb.txt.

Each of these should contain an appropriate “merged” document. Here is what should be put into smiths.txt, for example:

Name: Steve Smith (465)
UniqueID: smiths
Total: 95/100
Subtotal: 100
Hours Late: 1
Late deduction: 5
50/50 Problem #1: Excellent work
50/50 Problem #2: Excellent work

Notes:

- References to fields (i.e., <<NAME>>) will appear with matching case and without intervening spaces. If your program sees something that appears like a field reference but does not match a column heading, assume that it is part of the text output and write it to the file untranslated.