Programming Assignment 5

All solutions are to be written using Python 3. Make sure you provide comments including the file name, your name, and the date at the top of the file you submit. Also make sure to include appropriate docstrings for all functions.

The names of your functions must exactly match the names given in this assignment. The order of the parameters in your parameter list must exactly match the order given in this assignment.

For any given problem below, you may want to write additional functions other than those specified for your solution. That's fine with us.

Keep in mind that the point of this assignment is to give you practice with nested loops and nested lists, not to find ways to avoid them.

In the world of mathematics, a matrix is a collection of numbers arranged into a fixed number of rows and columns. We can represent a matrix as a two-dimensional table or array. Mathematicians have defined many operations on matrices. In this assignment, you'll implement two of those operations as Python functions.

Problem 1

A matrix is symmetric if every element of the matrix at row i and column j is equal to the element at row j and column i. A matrix can only be symmetric if it is a square matrix (that is, the matrix has the same number of rows as columns). For example, this matrix

$$\begin{bmatrix} 5 & 4 & 3 \\ 4 & 0 & 7 \\ 3 & 7 & -1 \end{bmatrix}$$

is symmetric, while the following matrix

$$\begin{bmatrix} 5 & 4 & 3 \\ 4 & 0 & 7 \\ 3 & 6 & -1 \end{bmatrix}$$

is not symmetric. Your task is to construct a Python function called `symmetric` which expects one argument: a two-dimensional table representing a square matrix. Your function should return True if the matrix is symmetric and False otherwise. You may assume that the table passed as the argument will have the same number of rows as columns and that it will contain only numbers. Here are some examples of how your function should behave:

```
>>> m1 = [[5, 4, 3], [4, 0, 7], [3, 7, -1]]
>>> m2 = [[5, 4, 3], [4, 0, 6], [3, 7, -1]]
>>> m3 = [[0, 0], [0, 0]]
>>> m4 = [[0, 0], [1, 0]]

>>> symmetric(m1)
True
>>> symmetric(m2)
False
>>> symmetric(m3)
True
>>> symmetric(m4)
False
```

Problem 2

Your task now is to construct a Python function called `transpose` which expects one argument: a two-dimensional table representing a matrix. This matrix does not have to be a square matrix. The transpose of a matrix is a new matrix whose rows are the columns of the original and whose columns are the rows of the original. If you want a more formal definition, here is one copied from Wikipedia: "The transpose A^T of a matrix A can be obtained by reflecting the elements along its main diagonal. Repeating the process on the transposed matrix returns the elements to their original position."

In other words, given this matrix:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

The transpose of the matrix would be this:

$$\begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$$

Your function should create a new table representing the transpose of the original table and return that new table. Your function should not alter the original table. Here are some examples of how your function should behave:

```
m5 = [[1, 2, 3, 4], [5, 6, 7, 8]]
m6 = [[1, 2, 3, 4]]
m7 = [[1, 2], [3, 4], [5, 6]]

>>> transpose(m5)
[[1, 5], [2, 6], [3, 7], [4, 8]]
>>> transpose(m6)
[[1], [2], [3], [4]]
>>> transpose(m7)
[[1, 3, 5], [2, 4, 6]]
```

Where to do the assignment

You can do this assignment on your own computer, or in the labs. In either case, use the IDLE development environment -- that's what we'll use when we grade your program. Put all the functions you created in a file called "prog5.py".

Submitting the Assignment

We will be using SmartSite to turn in assignments. Go to SmartSite, go to ECS 010, and go to Assignments. Submit the file containing your functions as an attachment.
Do NOT cut-and-paste your functions into a text window. Do NOT hand in a screenshot of your functions' output. We want one file from you: "prog5.py". **Please note that if you submit multiple files instead of just one "prog5.py" file, you will not receive any credit for this assignment.**

Saving your work

If you are working in the lab, you will need to copy your program to your own flash-drive or save the program to your workspace on SmartSite. To save it on flash-drive, plug the flash-drive into the computer (your TAs or the staff in the labs can help you figure out how), open the flash-drive, and copy your work to it by moving the folder with your files from the Desktop onto the flash-drive. To copy the file to your SmartSite workspace, go to Workspace, select Resources, and then use the Add button next to "My Workspace". Your TAs can help you with this if you run into trouble.