Magic Squares through Permutation (Due 19 Oct 2020)

A magic square is a square of numbers from 1 to n^2 such that the sum of every row, the sum of every column, and the sum of the two main diagonals is the same number. There are many algorithms to generate odd magic squares of order 3 or greater. There are also algorithms to generate even magic squares of order 4 or greater. But the algorithms to generate even order magic squares are more complicated and you cannot get **all** magic squares of a given order using these algorithms.

Magic Squares have fascinated mankind for centuries. They were thought to have magical properties about them and hence the name *magic square*. Here is a very famous magic square that is featured in the painting *Melancholia* by Albrecht Durer in 1514.

```
16   3   2  13
 5  10  11   8
 9   6   7  12
 4  15  14   1
```

Your input will be a single integer *n* that will be 3 or greater. For this assignment you will generate all magic squares through permutation. We will test you for order 3 magic squares but your code should be general enough that it can do higher order magic squares. The process is straight forward but time consuming. Here are the steps:

1. Create a 1-D list of integers 1 through n^2.
2. Permute this list of integers.
3. For each permutation check if this 1-D list is a magic square if converted to a 2-D list. If it is, then print the 1-D list.
4. Stop when you have gone through all the permutations.

The magic constant is the sum of the rows, columns or diagonals in a *n x n* magic square. The magic constant is given by $n * (n^2 + 1) / 2$, where *n* is the dimension of the magic square. For a *3 x 3* magic square the magic constant is 15. For a *4 x 4* magic square the constant is *34*.

In this program you will read a single integer *n* from standard input. For n = 3, the first line of your output will look as follows:

```
[2, 7, 6, 9, 5, 1, 4, 3, 8]
```

There should be 8 magic squares of order 3.

If you like the challenge, your should try to optimize your code to do n = 4. For the fourth order magic square the first and last lines in your output will look as follows:

```
[1, 2, 15, 16, 12, 14, 3, 5, 13, 7, 10, 4, 8, 11, 6, 9]

...

[16, 1, 15, 2, 5, 8, 10, 11, 4, 13, 3, 14, 9, 12, 6, 7]
```

There should be 7040 magic squares of order 4!

You *must* optimize your code so that it does not go through all the permutations. For example, if the first row does not add to the magic constant stop that permutation and go to the next one. If the second row does not add to the magic constant then stop that permutation and go to the next one. And similarly for the sum of the third row.

For this assignment you may work with a partner. Both of you must read the paper on Pair Programming and abide by the ground rules as stated in that paper.

Here is the template of the file called MagicSquare.py that you will be turning in. We are looking for clean and structured design using the standard coding conventions in Python. You may not change the function signatures but you may add as many helper functions as needed. The file will have a header of the following form:

```
#  File: MagicSquare.py

#  Description:

#  Student Name:

#  Student UT EID:

#  Partner Name:

#  Partner UT EID:

#  Course Name: CS 313E

#  Unique Number:

#  Date Created:

#  Date Last Modified:
```

To run this code on the command line on the Mac you will do

```
python3 MagicSquare.py
```

To run the code on the command line on a Windows machine you will do

```
python MagicSquare.py
```

If you are working with a partner both of you will be submitting only one program but make sure that you have your partner's name and eid in your program. If you are working alone, then remove the two lines that has the partner's name and eid in the header.

Use the Canvas system to submit your **MagicSquare.py** file. We should receive your work by 11 PM on Monday, 19 Oct 2020. There will be substantial penalties if you do not adhere to the guidelines. Remember Python is case sensitive. The name of your file must match exactly what we have specified.

- Your Python program should have the proper header.
- Your code must run before submission.
- You should be submitting your file through the web based *Canvas* program. We will not accept files e-mailed to us.

References

- Magic Square Article in Wolfram MathWorld
- Article on Magic Square in Wikipedia
- Even Order Magic Square
- Durer's Magic Square