Problem F.1

Problem 265: Binary Circles

Due Date: 5/3/2019
Folder: `FinalProject`
File Name: `F1_Prob265_Name.py`
Points: 20 points

Problem Background

Assume we have 2^N binary digits, that is 0's or 1's (henceforth called *bits*). These bits can be arranged in a circle in many different ways. For a visual of this, see Problem 265. These are called *binary circles.* At this link, an example of $N = 3$ is given. If we examine all the N element subsequences of the binary circles, taking each subsequence in clockwise order, we want to determine which circles exist such that all 2^N subsequences are distinct.

Obviously, once you find such a circle, you can simply rotate the elements and get another circle. In light of that, we will ignore all rotations of a circle. When $N = 3$, ignoring circle rotations, there are only 2 such binary circles.

Program Criteria

Write a program that does the following:

- Has an input variable `N` to determine the 2^N number of bits on the circle.
- Determine how many binary circles have all distinct N length subsequences. Be sure to ignore rotations of the circle. (That is, all 2^N rotations of a circle with distinct N length subsequences will count as just one such binary circle.)
- Print out the number of binary circles, as well as the elements of the circles, starting at the top and going in clockwise order.

Deliverables

Place the following in a folder named `FinalProject` in your repository:

- A Python file `F1_Prob265_Name.py` that satisfies the program criteria.
- A PDF file `F1_Prob265_Name.pdf` that describes how your program works. This should be a description of how you went about solving this problem. You should go into some detail about your solution method, but I don't want to see something about every `if` statement and `for` loop.