CSc 110, Autumn 2017
Programming Project #1: Song (10 points)
Due Tuesday, August 29th 2017, 7:00 PM
Based on an assignment from the University of Washington by Stuart Reges and Marty Stepp

Program Description:

This program tests your understanding of functions and print statements. Write a Python program in a file named song.py. (Use exactly this file name, including identical capitalization.)

For this assignment, you will write a program that outputs the cumulative song below, "There Was An Old Lady Who Swallowed A Fly." A cumulative song is one where each verse builds upon previous verses. Examples of other well-known cumulative songs are "The House That Jack Built" and "The Twelve Days of Christmas."

There was an old woman who swallowed a fly.
I don't know why she swallowed that fly,
Perhaps she'll die.

There was an old woman who swallowed a spider,
That wriggled and iggled and jiggled inside her.
She swallowed the spider to catch the fly,
I don't know why she swallowed that fly,
Perhaps she'll die.

There was an old woman who swallowed a bird,
How absurd to swallow a bird.
She swallowed the bird to catch the spider,
She swallowed the spider to catch the fly,
I don't know why she swallowed that fly,
Perhaps she'll die.

There was an old woman who swallowed a cat,
Imagine that to swallow a cat.
She swallowed the cat to catch the bird,
She swallowed the bird to catch the spider,
She swallowed the spider to catch the fly,
I don't know why she swallowed that fly,
Perhaps she'll die.

There was an old woman who swallowed a dog,
What a hog to swallow a dog.
She swallowed the dog to catch the cat,
She swallowed the cat to catch the bird,
She swallowed the bird to catch the spider,
She swallowed the spider to catch the fly,
I don't know why she swallowed that fly,
Perhaps she'll die.

<< Your custom sixth verse goes here >>

There was an old woman who swallowed a horse,
She died of course.

(continued on back)
One way to write this program would be to simply write a series of print statements that output each line of the song in order. But such a solution would not receive full credit. Part of the challenge of this assignment lies in recognizing the structure and redundancy of the song and improving the code using functions.

**Style Guidelines:**

You should not have any code, except a call to your main function, outside of a function. You should not place any print statements inside of the main function. (It is okay to have print statements to just print blank lines in main.) Instead of printing outside of a function or in main, use additional functions for two reasons:

1. **Structure**
   
   You should write functions to capture the structure of the song. You should, for example, have a function for each of the verses of the song (including your custom verse) to print that verse's entire contents.

2. **Eliminating redundancy**
   
   You should use only one print statement for each distinct line of the song (other than blank lines). For example, the following line appears several times in the output, but you should have only one print statement in your program that prints that line of the song:
   
   ```
   Perhaps she'll die.
   ```
   
   A function that prints just one line is not good style. Instead, identify groups of lines that appear in multiple places in the song and create functions to represent those groups. There is a general cumulative structural redundancy to the song that you should eliminate with your functions. Recall that functions can call other functions if necessary (which can themselves call other functions, and so on). The key question to ask is whether you have repeated lines or groups of lines of code that could be eliminated if you structured your functions differently. This includes sequences of print statements and also repeated sequences of function calls.

You do not have to eliminate partial-line redundancy in lines that are similar but not identical, such as the lines that start with “There was an old woman who swallowed a.”

Include a comment at the beginning of your program with some basic information and a description of the program in your own words. For example:

```
# Suzy Student, CSc 110, Autumn 2049, Section XX
# Programming Assignment #1, 06/07/49
#
# This program's behavior is ...
```

For this assignment, you should limit yourself to the Python features covered in the first three lectures. Though we will cover more material while you work on this assignment, please do not use it on this program. Specifically, do not use mathematical expressions, print statements with an end parameter, or for loops.

**Submission and Grading:**

Turn in your Python source code file electronically from the Homework link on the course web site.

Part of your program's score will come from its "external correctness." External correctness measures whether the output matches exactly what is expected. We are very picky about the output matching exactly and expect every character and space to match. Use the output comparison tool to help you make sure your output is perfect. Programs that do not run will receive no external correctness points.

The rest of your program's score will come from its "internal correctness." Internal correctness measures whether your source code follows the stylistic guidelines specified in this document. This includes having an adequate comment header and capturing the structure and redundancy of the song as specified previously. You should also limit the lengths of all lines in your program to fewer than 100 characters.