Class 6: Challenge problems

1) **Recursive functions with Turtle Graphics**
A turtle module for simple graphics has been included with standard distributions of Python, starting with Python 2. The "turtle" is a cursor that you can control to draw on a two-dimensional palette. The key of this approach is that you are the turtle and therefore all the drawing is done relative to the present position of the turtle. It could turn or move forward/backward but it can never go to a specific Cartesian coordinate.

All you'll need to know for this project about the Turtle module is the following:
First,

```
import turtle
from turtle import *
```

The "turtle" cursor in its window won't appear until you actually run the program. But remember that the default starting point is always a cursor in the center of its window facing the right side of the screen.

The methods you need here are:
`forward(`*`distance`*`)` : Move forward distance in current direction.
`backward(`*`distance`*`)` : Move backward distance in the opposite direction.
`right(`*`angle`*`)` : Turn right by angle degrees
`left(`*`angle`*`)` : Turn left by angle degrees
`speed(`*`speed`*`)` : You can also set the drawing speed as int in range 1–10 (1 slowest, 10 fastest)
`dot()` : mark current position with dot
`stamp()` : mark current position with a triangular shape

Before starting this project, I would suggest practicing the above commands several times.

Write a recursive function to create one of the diagrams below. (left: recursive tree, right: Sierpinski triangle) The number of stages (for e.g., 5 levels for the recursive tree) shouldn't be hard coded in the recursive program. But when you call it in the main function, test the recursive function by specifying the layers.

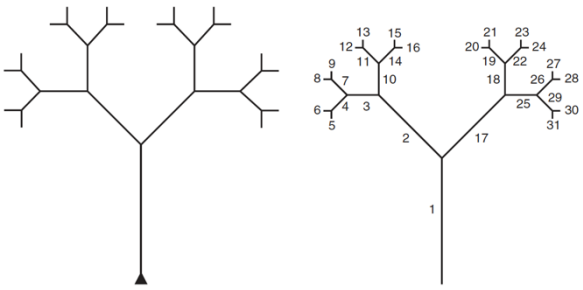

FIGURE 16.4 Recursive tree: (a) Python-drawn on left; (b) order-of-drawing on right.

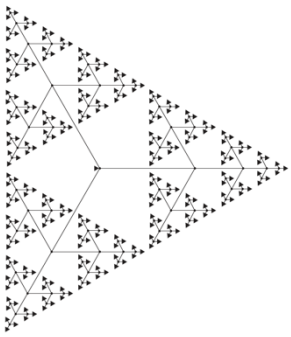

FIGURE 16.5 Sierpinski triangle.

2) Review of advanced I/O: Make a cheatsheet for Scrabble

Download the file VtoZ.txt
This is a portion of the dictionary file publicly shared on GitHub.

These are the methods you need in your Scrabble class:
lengthX(x): returns an array of all words that are x characters long

startsLetter(x, letter): returns an array of all words that are x characters long and starts with letter (Please ignore uppercase/lowercase)

hasLetter(x, letter): returns an array of words that are x characters long and contains the letter but does not start with it

Write a ScrabbleCheat module with these methods, then write a tester to test your methods according to user input. Deal with exceptions (i.e. when the user puts in a weird input, such as wanting a letter that starts with an A when this only contains V to Z) The tester should output a text file with the desired output as specified by the user.