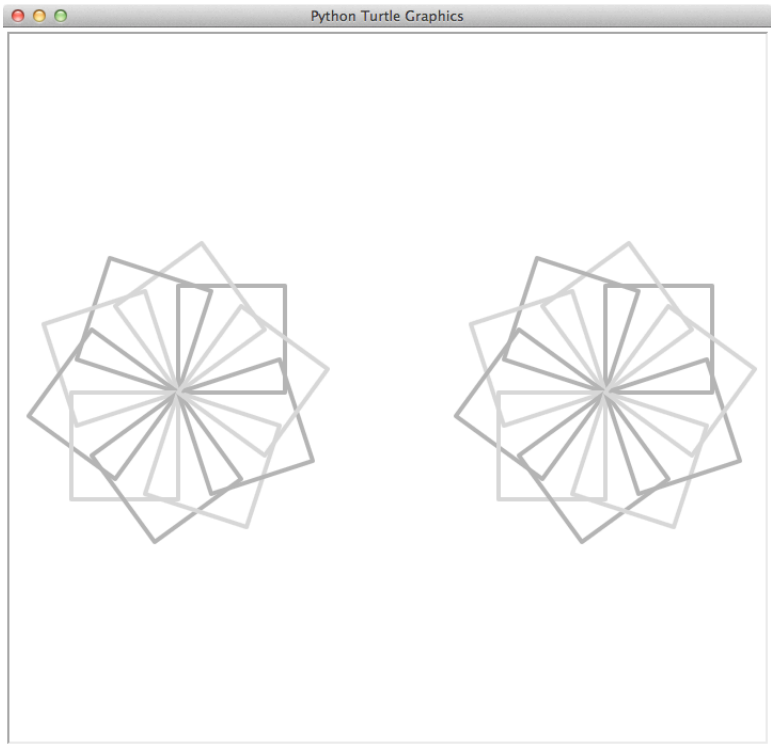
Problem-Based Intro. to Computer Science Flower Lab

09/20/2011

1 Problem

Using Python's turtle graphics module, we would like to design a program that draws two flower-like figures by rotating a square as shown below.



1.1 Problem-solving Session (20%)

You will work in a group of five to six students as determined by your instructor. Each group will work together to complete the following activities. During problem-solving, do not consider color; all pseudo-code should be about black-and-white images.

1. Write pseudo-code for a function `boxRec`. It will take one argument, a number, that is the length of a side of the box. Its effect is to draw a box in the turtle window. This pseudo-code must include a recursive helper function.
2. Write pseudo-code for a function `drawFlowerRec`. It will take two arguments, a natural number that is the number of petals, and a number that is the length of

a side of a box. Its effect is to draw a flower in the turtle window by repeatedly drawing boxes. This pseudo-code must include a recursive helper function.

3. Draw an execution diagram of `drawFlowerRec(2, 100)`. The diagram should *not* expand `boxRec`.
4. Write pseudo-code for a function `drawFlowerIt`. It will take two arguments, a natural number that is the number of petals, and a number that is the length of a side of a box. Its effect is to draw a flower in the turtle window by repeatedly drawing boxes. The implementation must use a looping construct. Your pseudo-code should make use of a box drawing function.

At the end of problem-solving, hand in your work, one copy per group, and number each item.

1.2 Implementation (80%)

Each student will *individually implement* and submit their own solution to the problem as a Python program named `flower.py`. This file should contain the following.

- Python implementations of the functions outlined in the problem-solving session.
- Python code that generates the flowers in alternating shades of gray (light-gray and dark-gray).
- A function `boxIt`. It will take one argument, a number, that is the length of a side of the box. Its effect is to draw a box in the turtle window. The implementation must use a looping construct.
- A function `main` that prompts the user to type in the number of petals and the size of the box/petal. It should then generate two flowers side-by-side; one using the recursive functions, and one using the iterative functions.

Also you will submit a text document, `readme.txt`, which contains a description of your design and your test cases.

1.2.1 Program Operation

The program shall be runnable from the command line of a terminal window as follows:

```
python3 flower.py
```

When run, the program must prompt the user for input, draw the two flowers (that alternate in shades of gray), and then pause and wait for the user to press the ENTER/RETURN key before the program terminates.

1.2.2 Grading

Your grade is based on these factors:

- 33%: The program draws the flowers in the turtle window.
- 32%: The box functionality and the flower functionality are implemented both iteratively and recursively.

- 5%: The code follows the style guidelines on the course web site.
- 10%: The `readme.txt` file contains appropriate design documentation, and test information. Make sure things are well documented here so that the grader of your work knows what to examine and do. The file should have the following items/sections:
  - A brief summary of your design; this should be no more than 4 sentences.
  - Features in the code.
  - Description of testing.

1.3 Submission

Compress your `flower.py` and `readme.txt` into one file named **flower.zip**, and submit that to the **myCourses dropbox** for this lab.

On machines that have the `zip` compression program, the command is shown here:

```
zip flower.zip flower.py readme.txt
```