HOMEWORK 14

CS 125

due at 11:45pm (classtime) on Tuesday, October 13

Write one or more Python *functions* to solve each of the following problems. Plan each function on paper before you implement it in code.

Prepare your solutions in a single Python file. Use comments to clearly state the problem number for each of your solutions. Provide test cases to show that your functions produce the desired output. Upload your file to the Homework 14 assignment on Moodle.

1. **Reversing a list**: Write a recursive function `revList(alist)` that reverses a list. Your function should take a list as its argument, and it should return a new list containing the same elements as the original list, but in the reverse order.

   For example, `revList(['a', 2, True])` should return `[True, 2, 'a']`.

2. **Palindromes**: A *palindrome* is a word or phrase that read the same forward and backwards, such as "racecar". Write a recursive function `isPalindrome(text)` that determines whether or not a string is a palindrome.

   For example, `isPalindrome('racecar')` should return `True`, while `isPalindrome('computer')` should return `False`.

3. **Fibonacci sequence:** The *Fibonacci sequence* is the sequence of numbers 1, 1, 2, 3, 5, 8, 13, …. The first two numbers of the sequence are each 1; otherwise, each number is the sum of the preceding two. Write a recursive function, `fibonacci(n)`, that computes the nth term of the Fibonacci sequence.

   Specifically, let $F_1 = 1$, $F_2 = 1$, and $F_n = F_{n-1} + F_{n-2}$ for $n > 2$. A call to `fibonacci(n)` should return the value of F_n.

   For example, `fibonacci(4)` should return `3`, and `fibonacci(15)` should return `610`.

4. **Recursive trees**: Modify the recursive tree program in the text to produce a more realistic tree. Specifically, implement any *three* of the following ideas:

- Modify the thickness of the branches so that as `branchLen` gets smaller, the line gets thinner.
- Modify the color of the branches so that as `branchLen` gets very short, the branch is colored like a leaf.
- Modify the angle used in turning the turtle so that at each branch point the angle is selected at random in some range. For example, choose the angle between 15 and 45 degrees. Play around to see what looks good.
- Modify `branchLen` recursively so that instead of always subtracting the same amount you subtract a random amount in some range.