Homework 7

Due Date: June 4, 2009 at 5:00PM   **Points**: 100

Changes

All problems: The recursive functions can all be written without any looping. Please do *not* use any loops in your function.

Problem 1: Instead of ending the input with an end-of-file, end it with a blank line.

Written Exercises

None

Programming Exercises

Remember to turn in your error logs and refinement files.

1. (*30 points*) Write a recursive function *max* to find the largest number in a list. The maximum number is the larger of the first item and the maximum of all the other items. Then write a program that prompts the user to enter numbers. The user should enter 1 number per line, and use a blank line to terminate the input. Your program then calls the recursive function to determine the maximum. Your program then prints that number. Do *not* use any loops in your recursive function! Please turn in the program in the file `max.py`.
    [*text*, §13.6, Programming Exercises problem 4, modified]

2. (*30 points*) Write a recursive function that prints all the elements of a list of words, one per line. You may *not* use while or for loops to do the printing! The single parameter to the function is to be the word list. Then write a program to read a file containing a list of words and call your recursive function to print them. Do *not* use any loops in your recursive function! Please turn in the program in the file `print.py`.

3. (*40 points*) A palindrome is a sentence that contains the same sequence of letters reading it either backwards or forwards (ignoring the case of the letters). A classic example is: "Able was I, ere I saw Elba." Write a recursive function that detects whether a string is a palindrome. The basic idea is to check that the first and last letters of the string are the same letter; if they are, then the entire string is a palindrome if everything between those letters is a palindrome. There are a couple of special cases to check for. If either the first or last character of the string is not a letter, you can check to see if the rest of the string is a palindrome with that character removed. Also, when you compare letters, make sure that you do it in a case-insensitive way. Do *not* use any loops in your recursive function!

    Use your function in a program that prompts a user for a phrase and then tells whether or not it is a palindrome. Here's another classic for testing: "A man, a plan, a canal-Panama!"

    Please turn in the program in the file `rpal.py`.

    Hint: Try solving this without worrying about punctuation and spacing. Once you have that, then add in the code to handle punctuation and spacing.
    [*text*, §13.6, Programming Exercises problem 3, modified]