Lab 04: Recursion

AUTHOR
Christian Osendorfer

Implement in Python

Unless otherwise stated, don't use any modules that implement a solution to the questions asked. Come up with your doctests and compare these with those from other students.

- Without using iteration, write the recursive function `oddCount(L)` which given a possibly-empty list `L` of integers, returns the number of odd integers in `L`.
- Without using iteration, write the recursive function `oddSum(L)` which given a possibly-empty list `L` of integers, returns the sum of the odd integers in `L`. Do not create a new list. Return 0 if the list has no odd integers in it.
- Without using iteration, write the recursive function `oddsOnly(L)` which given a possibly-empty list `L` of integers, returns a new list containing only the odd integers in `L` in the same order they appear in `L`.
- Without using iteration, write the recursive function `maxOdd(L)` which given a possibly-empty list `L` of integers, returns the largest odd integer in `L`, or `None` if `L` does not contain any odd integers.
- Without using iteration, write the recursive function `hasConsecutiveDigits(n)` that takes a possibly-negative int value `n` and returns True if that number contains two consecutive digits that are the same, and False otherwise.
- Without using iteration, write the function `alternatingSum(L)` that takes a possibly-empty list of numbers `L`, and returns the alternating sum of the list, where every other value is subtracted rather than added. For example: `alternatingSum([1,2,3,4,5])` returns `1-2+3-4+5` (that is, 3). If `L` is empty, return 0.

> **Note**
>
> A *palindrome* is a sequence of characters (a string) that has the same elements in normal and reverse order.

- Implement the function `is_palindrome(strng)` that tests whether a string `strng` of length at least 1 is a palindrom. Implement two versions: One implementation should only use recursion, the other version should use a python loop.
- Write a recursive function `num_eights` that takes a positive integer `pos` and returns the number of times the digit 8 appears in `pos`.
- Implement `merge(dict1, dict2)` that merges two dictionaries `dict1` and `dict2` having either `int` or `float` values and returns a new dictionary. If a key appears in both dictionaries the value

for this key in the new dictionary is the sum of the values.

Python Questions

Answer the following questions in a concise manner (that is, not more than 10 sentences).

- What is the definition of a type in Python? List at least four examples of types built into Python.
- What is an *expression*? What is a *statement*? Give an example of each.
- What is the difference between a *mutable* and an *immutable* type? Give an example of each.
- What is a *parameter* ? What is an *argument*? How are they related?
- What is the difference between a *function definition* and a *function call* ? Give an example of each.