Python Scripts and Problems

Websites

* CodingBat Python
* Project Euler
* HackerRank

1. Q: If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23.

Find the sum of all the multiples of 3 or 5 below 1000.

1. Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.

1. The prime factors of 13195 are 5, 7, 13 and 29.

What is the largest prime factor of the number 600851475143?

1. Given an int n, return the absolute difference between n and 21, except return double the absolute difference if n is over 21.  
   diff21(19) → 2  
   diff21(10) → 11  
   diff21(21) → 0
2. Given 2 ints, a and b, return True if one of them is 10 or if their sum is 10.  
   makes10(9, 10) → True  
   makes10(9, 9) → False  
   makes10(1, 9) → True
3. Given a string and a non-negative int n, return a larger string that is n copies of the original string.  
   string\_times('Hi', 2) → 'HiHi'  
   string\_times('Hi', 3) → 'HiHiHi'  
   string\_times('Hi', 1) → 'Hi'
4. Given an array of ints, return True if one of the first 4 elements in the array is a 9. The array length may be less than 4.  
   array\_front9([1, 2, 9, 3, 4]) → True  
   array\_front9([1, 2, 3, 4, 9]) → False  
   array\_front9([1, 2, 3, 4, 5]) → False
5. Given a non-empty string like "Code" return a string like "CCoCodCode".

string\_splosion('Code') → 'CCoCodCode'  
string\_splosion('abc') → 'aababc'  
string\_splosion('ab') → 'aab'

1. Given two strings, a and b, return the result of putting them together in the order abba, e.g. "Hi" and "Bye" returns "HiByeByeHi".  
   make\_abba('Hi', 'Bye') → 'HiByeByeHi'  
   make\_abba('Yo', 'Alice') → 'YoAliceAliceYo'  
   make\_abba('What', 'Up') → 'WhatUpUpWhat’
2. Given two strings, return True if either of the strings appears at the very end of the other string, ignoring upper/lower case differences (in other words, the computation should not be "case sensitive"). Note: s.lower() returns the lowercase version of a string.  
   end\_other('Hiabc', 'abc') → True  
   end\_other('AbC', 'HiaBc') → True  
   end\_other('abc', 'abXabc') → True
3. Given 3 int values, a b c, return their sum. However, if any of the values is a teen -- in the range 13..19 inclusive -- then that value counts as 0, except 15 and 16 do not count as a teens. Write a separate helper "def fix\_teen(n):"that takes in an int value and returns that value fixed for the teen rule. In this way, you avoid repeating the teen code 3 times (i.e. "decomposition"). Define the helper below and at the same indent level as the main no\_teen\_sum().  
   no\_teen\_sum(1, 2, 3) → 6  
   no\_teen\_sum(2, 13, 1) → 3  
   no\_teen\_sum(2, 1, 14) → 3
4. Given three ints, a b c, return True if one of b or c is "close" (differing from a by at most 1), while the other is "far", differing from both other values by 2 or more. Note: abs(num) computes the absolute value of a number.  
   close\_far(1, 2, 10) → True  
   close\_far(1, 2, 3) → False  
   close\_far(4, 1, 3) → True
5. Return the "centered" average of an array of ints, which we'll say is the mean average of the values, except ignoring the largest and smallest values in the array. If there are multiple copies of the smallest value, ignore just one copy, and likewise for the largest value. Use int division to produce the final average. You may assume that the array is length 3 or more.  
   centered\_average([1, 2, 3, 4, 100]) → 3  
   centered\_average([1, 1, 5, 5, 10, 8, 7]) → 5  
   centered\_average([-10, -4, -2, -4, -2, 0]) → -3
6. Write a function to reverse a string without using string reverse.
7. Implement a linked list
   1. Explanation of linked list: https://www.youtube.com/watch?v=njTh\_OwMljA
8. Reverse a linked list **in place**
9. Insertion sort is a method to sort a list. Implement it!
   1. Explanation of insertion sort: <http://cs.armstrong.edu/liang/animation/web/InsertionSort.html>
10. Write a Python script to generate and print a dictionary that contains a number (between 1 and n) in the form (x, x\*x).    
    Sample Dictionary ( n = 5) :   
    Expected Output : {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
11. Write a Python program to sort Counter by value.    
    Sample data : {'Math':81, 'Physics':83, 'Chemistry':87}  
    Expected data: [('Chemistry', 87), ('Physics', 83), ('Math', 81)]
12. Write a Python class to reverse a string word by word.   
    Input string : 'hello .py'  
    Expected Output : '.py hello'