Assignment 7 - Recursion

1. Take as input str, a number in form of a string. Write a recursive function to find the sum of digits in the string. Print the value returned.
2. Take as input str, a number in form of a string. Write a recursive function to convert the number in string form to number in integer form. E.g. for “1234” return 1234. Print the value returned.
3. Take as input str1 and str2, both strings. Write a function which tests if str2 is reverse of str1 or not and returns a Boolean value. Print the value returned.
4. Take as input str, a string. Write a function which tests if the string is a palindrome or not and returns a Boolean value. Print the value returned.
5. Take as input str, a string. Write a recursive function which returns a new string in which all duplicate consecutive characters are separated by a ‘*’. E.g. for “hello” return “hel*lo”. Print the value returned.
6. Take as input str, a string. Write a recursive function which returns a new string in which all duplicate consecutive characters are reduced to a single char. E.g. for “hello” return “helo”. Print the value returned.
7. Take as input str, a string. Write a recursive function which moves all ‘x’ from the string to its end. E.g. for “abexexdexed” return “abeedeedxxx”. Print the value returned.
8. Take as input str, a string.
    a. Write a recursive function which counts the number of times ‘hi’ appears in the string. Print the value returned.
    b. Write a recursive function which removes all ‘hi’ in the string. Print the value returned.
    c. Write a recursive function which replaces all ‘hi’ in the string with ‘bye’. Print the value returned.
9. Take as input str, a string.
    a. Write a recursive function which counts the number of times ‘hi’ appears in the string – but skip all such ‘hi’ which are followed by ‘t’ to form ‘hit’. Print the value returned.
    b. Write a recursive function which removes all ‘hi’ in the string – but skip all such ‘hi’ which are followed by ‘t’ to form ‘hit’. Print the value returned.
    c. Write a recursive function which replaces all ‘hi’ in the string with ‘bye’ – but skip all such ‘hi’ which are followed by ‘t’ to form ‘hit’. Print the value returned.
10. Take as input str, a string. A “twin” is defined as two instances of a char separated by a char. E.g. "AxA" the A's make a “twin”. “twins” can overlap, so "AxAxA" contains 3 “twins” - 2 for A and 1 for x. Write a function which recursively counts number of “twins” in a string. Print the value returned.
11. Take as input str, a string

   a. Write a recursive function which counts, the number of times “aaa” appears in the string. Print the value returned.
   b. Write a recursive function which counts, the number of times “aaa” appears in the string, but only such instances of “aaa” should be considered which do not overlap. Print the value returned.

12. Take as input str, a string. Assume that value of a=1, b=2, c=3, d=4, ... z=26. Write a recursive function (return type void) to print all possible codes for the string. E.g. for “1123” possible codes are aabc, kbc, alc, aaw, kw.
13. Take as input str, a string. Write a recursive function that checks if the string was generated using the following rules and returns a Boolean value:
    a. the string begins with an 'a'
    b. each 'a' is followed by nothing or an 'a' or "bb"
    c. each "bb" is followed by nothing or an 'a'

    Print the value returned.
14. Take as input str, a string. The string contains a single pair of parenthesis “(..)”. Write a recursive function which returns the contents between the parenthesis. E.g. for “xyz(abc)def” return “abc”. Print the value returned.
15. Take as input str, a string. The string is a mathematical expression e.g. “[a + {b + (c + d) + e} + f]”. Write a recursive function which tests if the brackets in expression are balanced or not and returns a Boolean value. Print the value returned.