**Assignments Day 7**

**Assignment 1:** You are given a binary number n which contains only digits 0's and 1's. Your task is to make all digits same by just flipping one digit (i.e. 0 to 1 or 1 to 0) only. Write a function isFlip() to receive the binary number n and print ‘YES’ if it is possible to make all the digits same by just flipping one digit else print 'NO'.

**Example:**

**Input:** 101

**Output:** Yes

**Assignment 2:** Write a Python function that accepts a string and calculate the number of upper case letters and lower case letters.   
**Example:**

**Input:** The quick Brown Fox

**Output:** No. of Upper case characters : 3  
 No. of Lower case characters : 13

**Assignment 3:** Write a Python function to check whether a number is perfect or not. In number theory, a perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself (also known as its aliquot sum). Equivalently, a perfect number is a number that is half the sum of all of its positive divisors (including itself).

*Example* : The first perfect number is 6, because 1, 2, and 3 are its proper positive divisors, and 1 + 2 + 3 = 6. Equivalently, the number 6 is equal to half the sum of all its positive divisors: ( 1 + 2 + 3 + 6 ) / 2 = 6. The next perfect number is 28 = 1 + 2 + 4 + 7 + 14. This is followed by the perfect numbers 496 and 8128.

**Input:** 496

**Output:** True

**Assignment 4:** Write a Python function that accepts a hyphen-separated sequence of words as input and prints the words in a hyphen-separated sequence after sorting them alphabetically.  
**Input:** green-red-yellow-black-white

**Output:** black-green-red-white-yellow

**Assignment 5:** Write a Python program of recursion list sum.   
**Input:** [1, 2, [3,4], [5,6]]  
**Output:** 21

**Assignment 6:** Write a recursive function to get the sum of the digits of a non-negative integer until it is reduced to a single digit.

**Input:** 3459

**Output:** 3

**Input:** 45

**Output:** 9

**Program:**

def sumDigits(n):

if n < 10:

return n

else:

return sumDigits(n % 10 + sumDigits(int(n / 10)))

print(sumDigits(3459))

**Assignment 7:** Write a recursive function to calculate the sum of the positive integers of n + (n−2) + (n−4) ... (until n−x =< 0).

**Input:** 6

**Output:** 12

**Input:** 10

**Output:** 30

**Assignment 8:** Write a recursive function to calculate the geometric sum of n.  
S(n) : a + ar + ar2 + ... + ar(n-1)

**Input:** 5 2 4 (a=5, r=2, n=4)

**Output:** 75

**Assignment 9:** Write a recursive function to calculate the value of 'a' to the power 'b'.

**Input:** 3,4

**Output:** 81

**Assignment 10:** Write a recursive function to find the prime factors of a number.

**Input:** 100

**Output:** 2 2 5 5