**Integer Roman Convertion**

Problem Statement:

Take a ten-digit integer number (maximum) and convert it to an equivalent number representation in ROMAN and vice versa.

* Input example:

Enter number: 3698

Enter roman value: MMMDCXCVIII

* Output Example:

Roman => MMMDCXCVIII

Integer => 3698

Algorithm:

DEFINE FUNCTION intToRoman(num):

SET values TO [ 1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1 ]

SET numerals TO [ "M", "CM", "D", "CD", "C", "XC", "L", "XL", "X", "IX", "V", "IV", "I" ]

SET res TO ""

FOR i, v IN enumerate(values):

res += (num//v) \* numerals[i]

num %= v

RETURN res

DEFINE FUNCTION romanToInt(s):

p={'M':1000,'D':500,'C':100,'L':50,'X':10,'V':5,'I':1}

ans=0;

left=0

FOR v IN s:

IF left<p[v]:

ans=ans-2\*left

ans=ans+p[v]

left=p[v]

RETURN ans

IF \_\_name\_\_ EQUALS "\_\_main\_\_":

num=int(INPUT("Enter number: "))

OUTPUT("Roman => ",CALL intToRoman(num))

roman=INPUT("Enter roman value: ")

OUTPUT("Integer => ",CALL romanToInt(roman))

Proposed Python Code:

/\* ------- main.py ------- \*/

def intToRoman(num):

values = [ 1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1 ]

numerals = [ "M", "CM", "D", "CD", "C", "XC", "L", "XL", "X", "IX", "V", "IV", "I" ]

res = ""

for i, v in enumerate(values):

res += (num//v) \* numerals[i]

num %= v

return res

def romanToInt(s):

p={'M':1000,'D':500,'C':100,'L':50,'X':10,'V':5,'I':1}

ans=0;

left=0

for v in s:

if left<p[v]:

ans=ans-2\*left

ans=ans+p[v]

left=p[v]

return ans

if \_\_name\_\_ == "\_\_main\_\_":

num=int(input("Enter number: "))

print("Roman => ",intToRoman(num))

roman=input("Enter roman value: ")

print("Integer => ",romanToInt(roman))

/\* ---------------------- \*/

Conclusion:

The proposed algorithm has a runtime of O(n), where n is the length of the number & string.

Limitations and assumptions for this algorithm include:

1.The number should be with in integer range.

2.While entering the roman character every character should be in Capital case.