1) <https://leetcode.com/problems/roman-to-integer/>

Sol:

class Solution:

def romanToInt(self, s: str) -> int:

dict1={'I':1,'V':5,'X':10,'L':50,'C':100,'D':500,'M':1000,

'A':4,'B':9,'E':40,'F':90,'G':400,'H':900}

s=s.replace('IV','A')

s=s.replace('IX','B')

s=s.replace('XL','E')

s=s.replace('XC','F')

s=s.replace('CD','G')

s=s.replace('CM','H')

out=0

for i in s:

out=out+dict1[i]

return out

Screenshot:

A screenshot of a computer

Description automatically generated

2) <https://leetcode.com/problems/implement-strstr/>

Sol:

class Solution:

def strStr(self, haystack, needle):

if needle in haystack:

return haystack.index(needle)

else:

return -1

Screenshot:

A screenshot of a computer

Description automatically generated

3) <https://leetcode.com/problems/length-of-last-word/>

Sol:

class Solution:

def lengthOfLastWord(self, s: str) -> int:

s = s.strip()

s=s.split()

return len(s[-1])

Screenshot:

A screenshot of a computer

Description automatically generated

4) <https://leetcode.com/problems/add-binary/>

Sol:

class Solution:

def addBinary(self, a: str, b: str) -> str:

sum1 = bin(int(a,2)+int(b,2))

return sum1[2:]

Screenshot:

A screenshot of a computer

Description automatically generated

5) <https://leetcode.com/problems/valid-palindrome/>

Sol:

class Solution:

def isPalindrome(self, s):

list1=[str(i) for i in s]

str1=''

for i in list1:

if i.isalnum():

str1=str1+i

str1=str1.lower()

if str1 == str1[::-1]:

return True

else:

return False

Screenshot:

A screenshot of a computer

Description automatically generated

6) <https://leetcode.com/problems/excel-sheet-column-title/>

Sol:

class Solution:

def convertToTitle(self, columnNumber):

res=0

string =""

n=columnNumber

while n:

res = (n-1)%26

string = chr(65+res)+string

n=(n-1)//26

return string

Screenshot:

A screenshot of a computer

Description automatically generated

7) <https://leetcode.com/problems/excel-sheet-column-number/>

Sol:

class Solution:

def titleToNumber(self, columnTitle):

list1 = [str(i) for i in columnTitle]

list1.reverse()

i=0

for k in range(0,len(list1)):

s=ord(list1[k])-64

i+=s\*26\*\*k

return i

Screenshot:

A screenshot of a computer

Description automatically generated

8) <https://leetcode.com/problems/reverse-string/>

Sol:

class Solution:

def reverseString(self, s: List[str]) -> None:

return s.reverse()

Screenshot:

A screenshot of a computer

Description automatically generated

9) <https://leetcode.com/problems/add-strings/>

Sol:

class Solution:

def addStrings(self, num1, num2):

num1 = int(num1)

num2= int(num2)

str1 = num1+num2

str2 = [str(i) for i in str(str1)]

str3 = ''.join(str2)

return str3

Screenshot:

A screenshot of a computer

Description automatically generated

10) <https://leetcode.com/problems/number-of-segments-in-a-string/>

Sol:

class Solution:

def countSegments(self, s: str) -> int:

list1=str(s)

list2=[]

str1=''

for i in list1:

if i!=' ':

str1=str1+i

else:

if str1 !='':

list2.append(str1)

str1=''

if len(str1)>0:

list2.append(str1)

return len(list2)

Screenshot:

A screenshot of a computer

Description automatically generated