1.Write a Python program to perform Cyclic Redundancy Check

CRC uses Generator Polynomial which is available on both sender and receiver side. An example generator polynomial is of the form like x3 + x + 1. This generator polynomial represents key 1011. Another example is x2 + 1 that represents key 101.  
Data word to be sent - 100100  
Key - 1101 [ Or generator polynomial x3 + x2 + 1]

def xor(a, b):

result = []

for i in range(1, len(b)):

if a[i] == b[i]:

result.append('0')

else:

result.append('1')

return ''.join(result)

def mod2div(divident, divisor):

pick = len(divisor)

tmp = divident[0 : pick]

while pick < len(divident):

if tmp[0] == '1':

tmp = xor(divisor, tmp) + divident[pick]

else:

tmp = xor('0'\*pick, tmp) + divident[pick]

pick += 1

if tmp[0] == '1':

tmp = xor(divisor, tmp)

else:

tmp = xor('0'\*pick, tmp)

checkword = tmp

def encodeData(data, key):

l\_key = len(key)

appended\_data = data + '0'\*(l\_key-1)

remainder = mod2div(appended\_data, key)

codeword = data + remainder

print("Remainder : ", remainder)

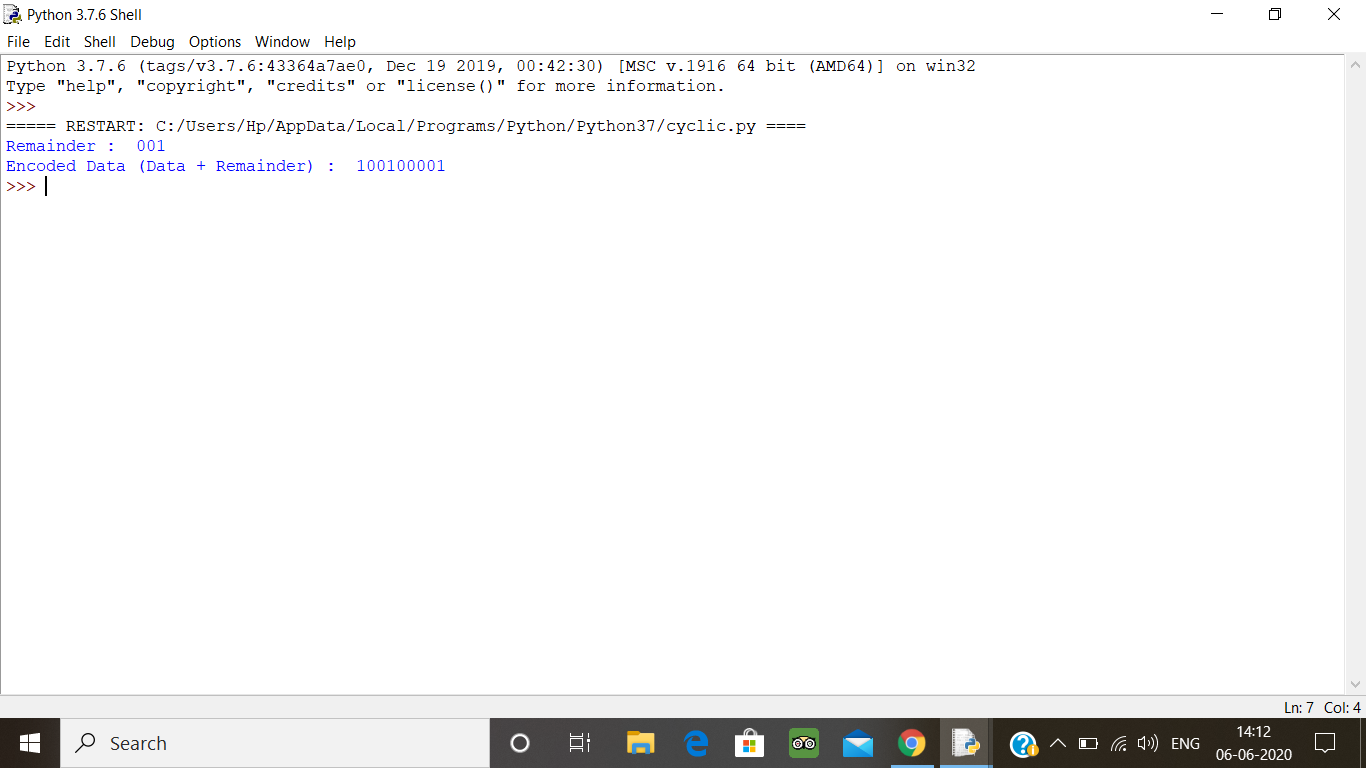
print("Encoded Data (Data + Remainder) : ",

data = "100100"

key = "1101"

encodeData(data, key)

**output:**



2. Description:  
Write a Python program to count the number of strings, provided string length is 2 or more and the first and last character are same from a given list of strings.  
Eg:  
Input  
list1['hia', 'aba' , '363']  
Output:  
Number of strings with first and last cahracter is same: 2

def match\_words(words):

ctr = 0

for word in words:

if len(word) > 1 and word[0] == word[-1]:

ctr += 1

return ctr

print(match\_words(['hia', 'aba' , '363']))

**output:**

