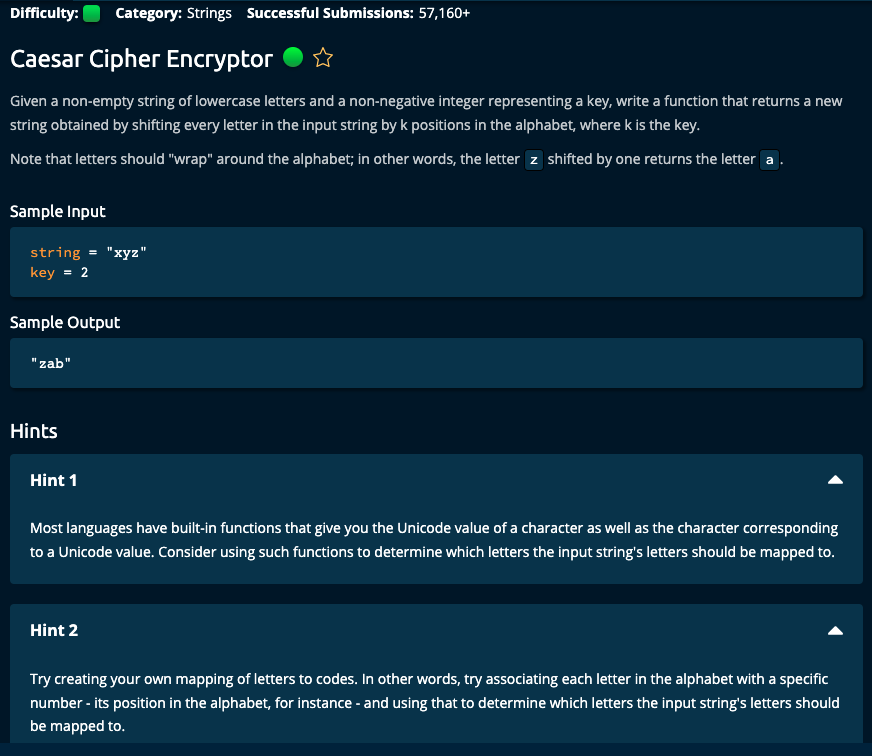
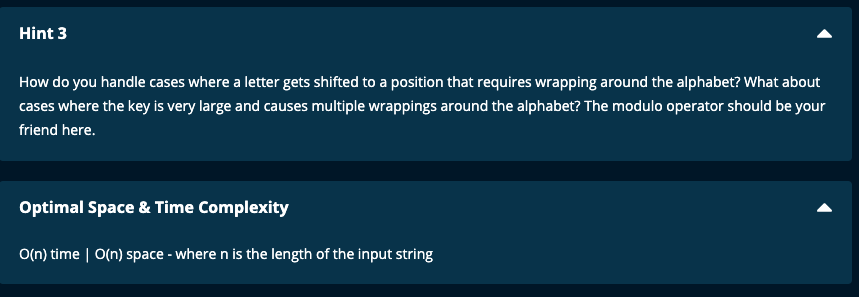
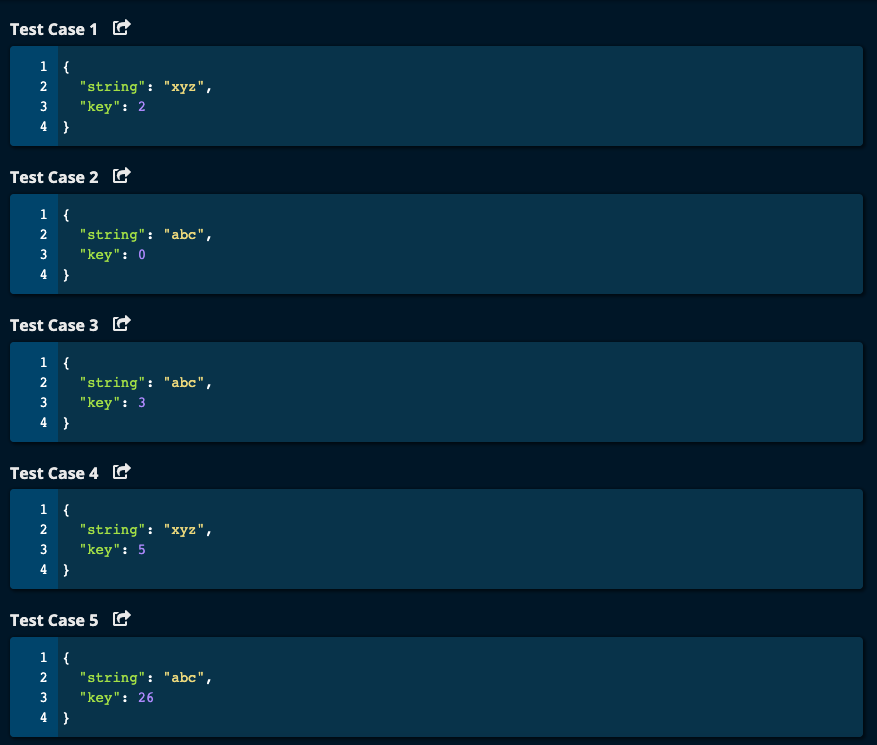
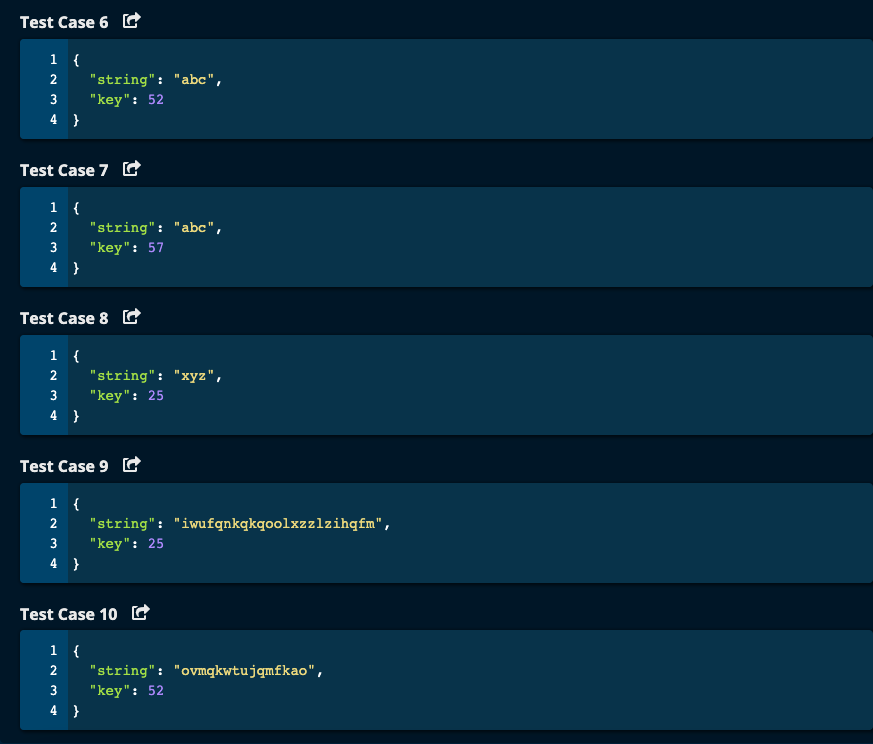
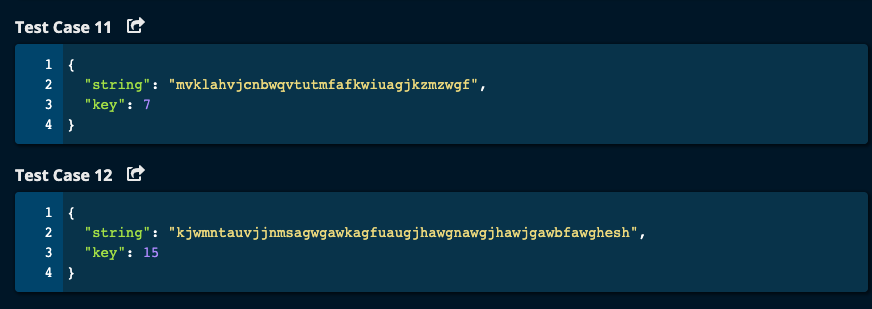
Caesar Cipher Encryptor (Easy)











My Solutions:

Solution 1:

# My Solution -- O(n) Time | O(n) Space

def caesarCipherEncryptor(string, key):

myDict = {}

myReverseDict = {}

numAlphabets = 26

for i in range(numAlphabets):

myDict[chr(ord('a') + i)] = i

myReverseDict[i] = chr(ord('a') + i)

result = []

for c in string:

result.append(myReverseDict[(myDict[c] + key) % numAlphabets])

print("c = ", c, "result = ", result)

return "".join(result)

JJ Notes:

1. Create a dictionary myDict with letters as keys and corresponding code, where ‘a’ is 0, ‘b’ is 1 and so on up to ‘z’ is 25. Also create the reverse dictionary for lookup.
2. Iterate through the string. For each letter c, get its newcode which is code + key. Now, take modulo of newcode modulo 26. Once we have the newcode, get its corresponding letter from the reverse dictionary.
3. Return the result array as a string.

--------------------------------------------------------

Solution 2:

def caesarCipherEncryptor(string, key):

result = []

for char in string:

newchar = chr((ord(char) - ord('a') + key) %26 + ord('a'))

print("ord(char) = ", ord(char))

print("newchar = ", newchar)

result.append(newchar)

#result.append(chr((ord(char) + key) % 26))

return "".join(result)

JJ Notes:  
Here we use ASCII codes were ord(‘a’) = 97. So we shift by key and get its modulo value with 26. Then again add ord(‘a’) and get the newchar.

Algoexpert Solution:

# Algoexpert Solution – O(n) Time | O(n) Space

def caesarCipherEncryptor(string, key):

newLetters = []

newKey = key % 26

alphabet = list("abcdefghijklmnopqrstuvwxyz")

for letter in string:

newLetters.append(getNewLetter(letter, newKey, alphabet))

return "".join(newLetters)

def getNewLetter(letter, key, alphabet):

newLetterCode = alphabet.index(letter) + key

return alphabet[newLetterCode % 26]

JJ Notes:

1. Initialize newLetters as an empty list which will hold the letters after applying the Ceasar Cipher Encryptor.
2. Find the newKey which is modulo 26 applied to the key.
3. Form an array with all the aphabets in order and the index of ‘a’ is 0, ‘b’ is 1 and so on.
4. Iterate through the string.
5. Convert the letter to the new letter and append it to newLetters using the function getNewLetter. This function takes the letter and finds its index in alphabet array. Then adds the key to this index to get the newLetter code. Apply modulo 26 to the newLetterCode and use this as the index to get the new letter from alphabet array.
6. Return newLetters as a string.