ASSIGNMENT-4

1. Odd String Difference

def find\_odd\_string(words):

def string\_to\_diff(s):

return [ord(s[i + 1]) - ord(s[i]) for i in range(len(s) - 1)]

diff\_count = {}

for word in words:

diff = tuple(string\_to\_diff(word))

diff\_count[diff] = diff\_count.get(diff, 0) + 1

for key, value in diff\_count.items():

if value == 1:

odd\_diff = key

break

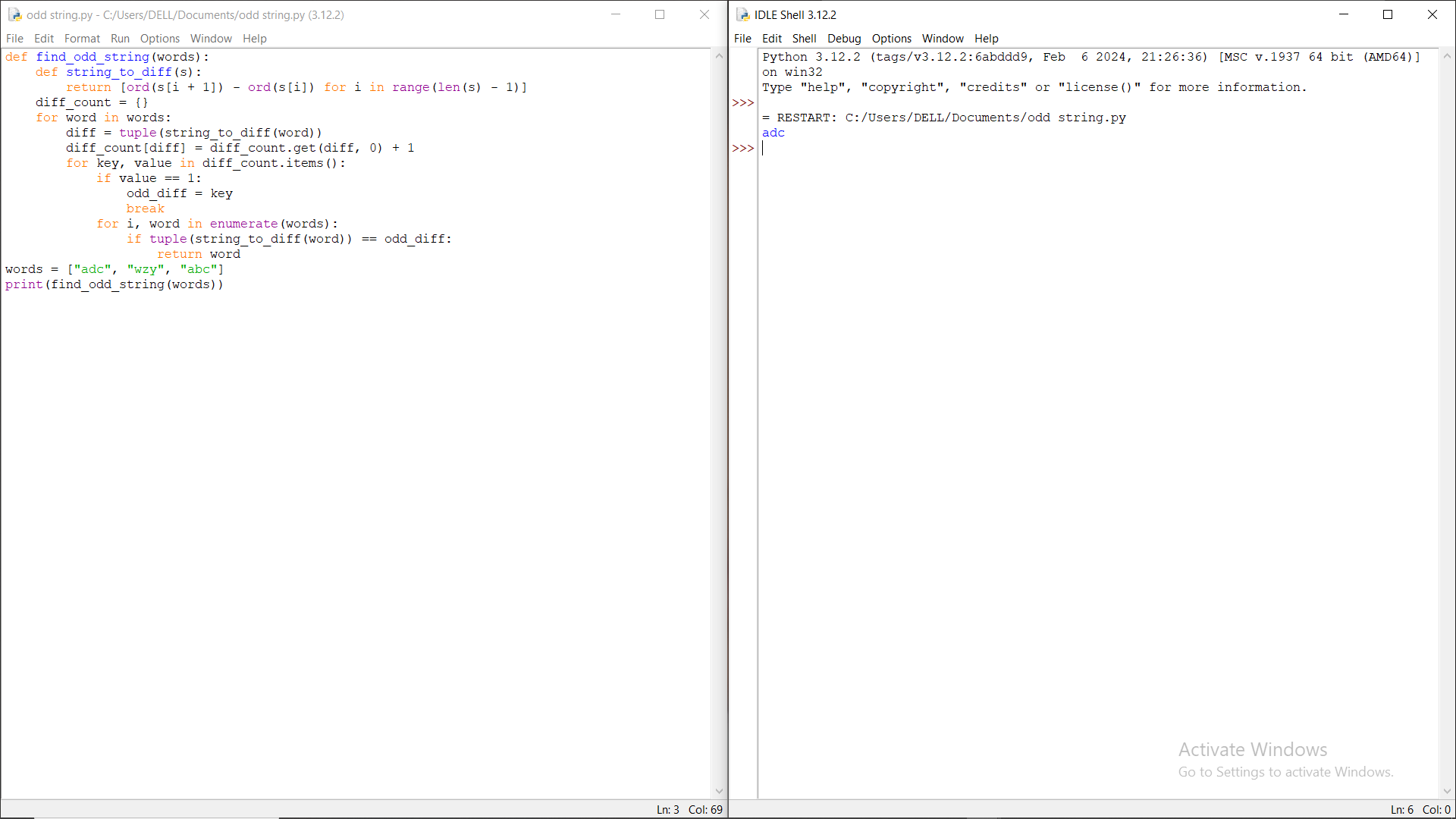
for i, word in enumerate(words):

if tuple(string\_to\_diff(word)) == odd\_diff:

return word

words = ["adc", "wzy", "abc"]

print(find\_odd\_string(words))



1. Words Within Two Edits of Dictionary

def isWithinTwoEdits(w1, w2):

return w1 == w2 or len(w1) == len(w2) and sum(c1 != c2 for c1, c2 in zip(w1, w2)) <= 1 or any(w1[:i] + chr(ord('a') + j) + w1[i+1:] == w2 for i in range(len(w1)) for j in range(26))

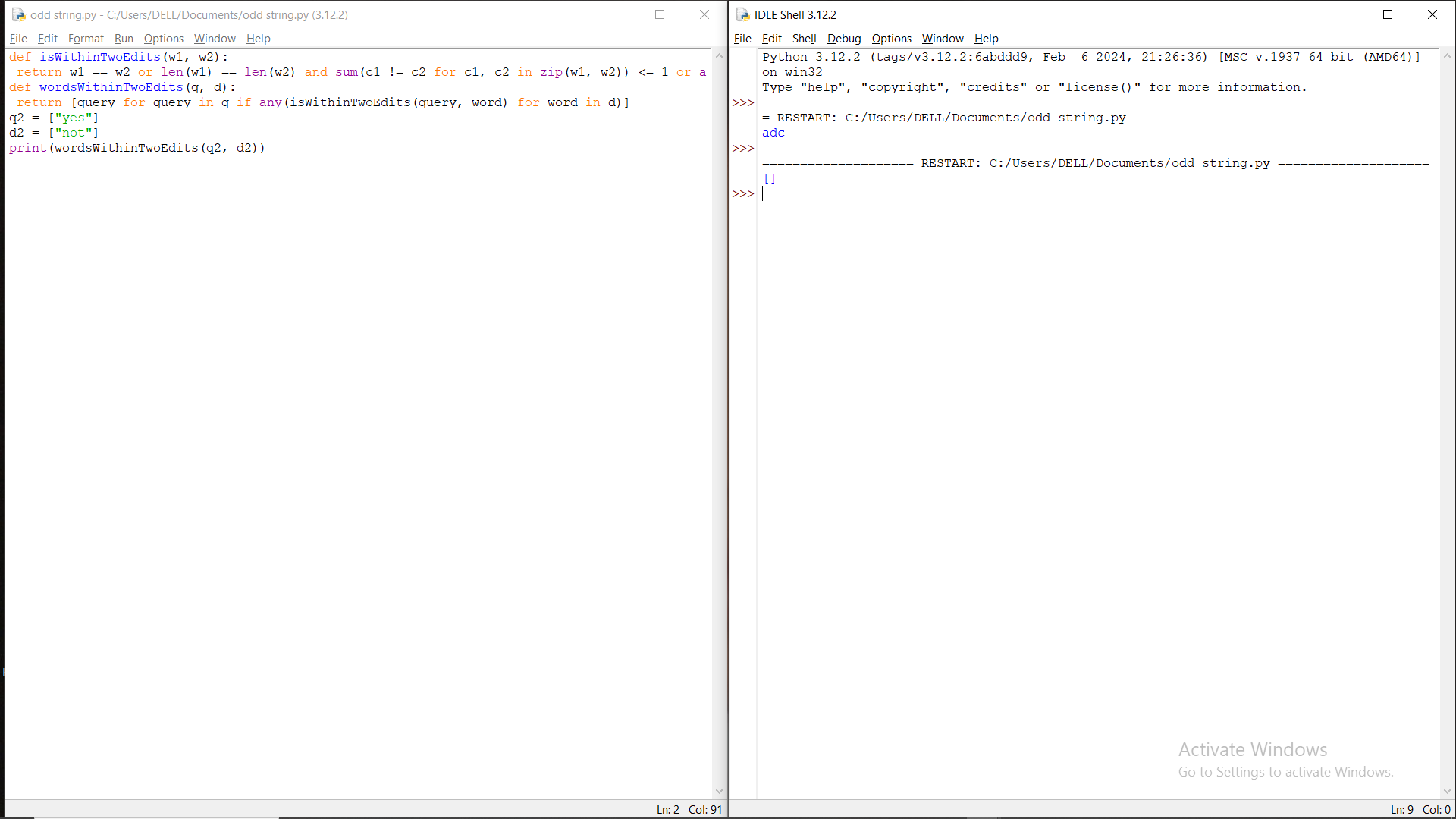
def wordsWithinTwoEdits(q, d):

return [query for query in q if any(isWithinTwoEdits(query, word) for word in d)]

q2 = ["yes"]

d2 = ["not"]

print(wordsWithinTwoEdits(q2, d2))



1. Next Greater Element IV

def printNGE(arr):

for i in range(0, len(arr), 1):

next = -1

for j in range(i+1, len(arr), 1):

if arr[i] < arr[j]:

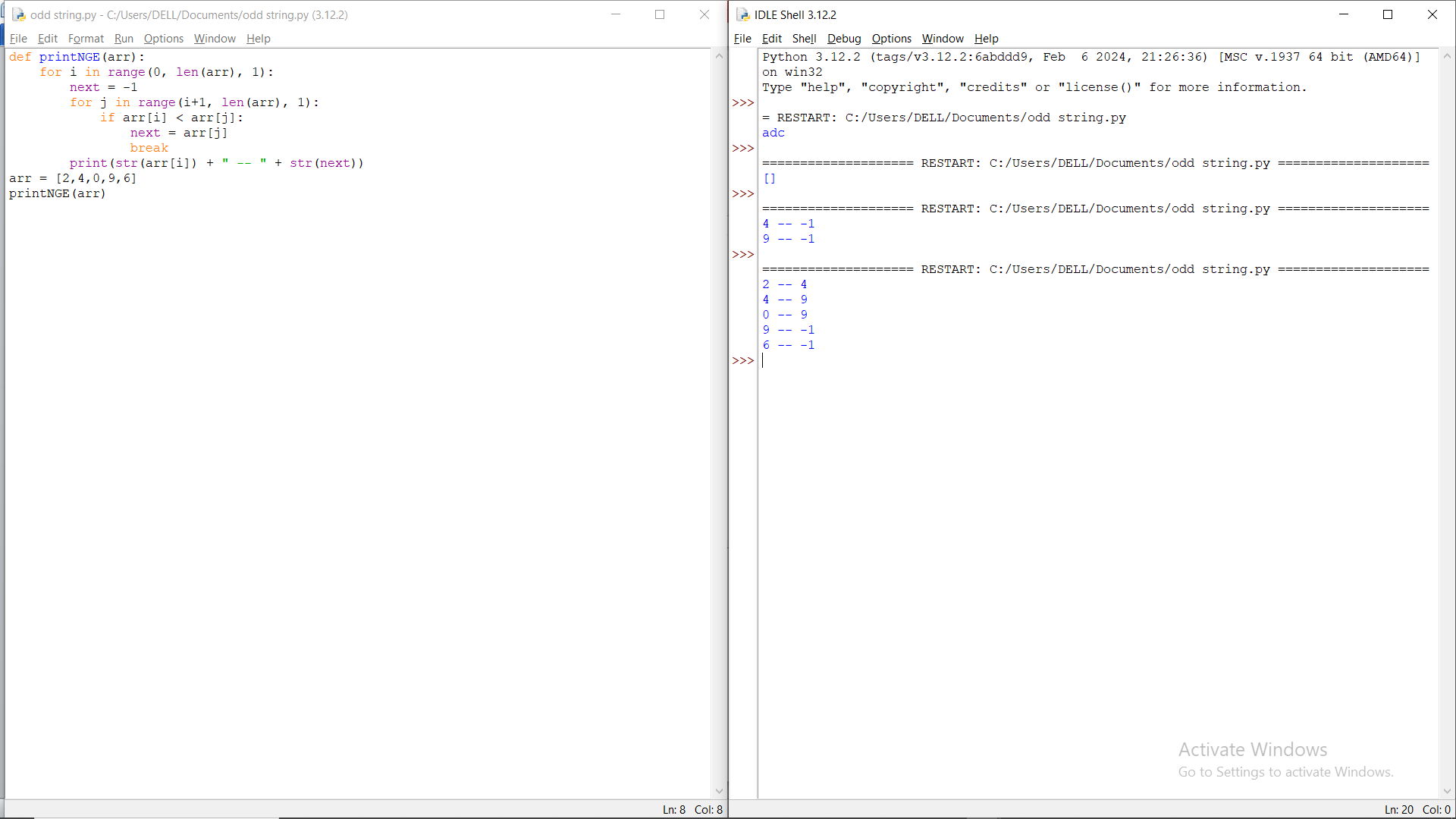
next = arr[j]

break

print(str(arr[i]) + " -- " + str(next))

arr = [2,4,0,9,6]

printNGE(arr)



1. Minimum Addition to Make Integer Beautiful

def digit\_sum(num):

return sum(int(digit) for digit in str(num))

def min\_addition\_to\_make\_beautiful(n, target):

x = 0

while digit\_sum(n + x) > target:

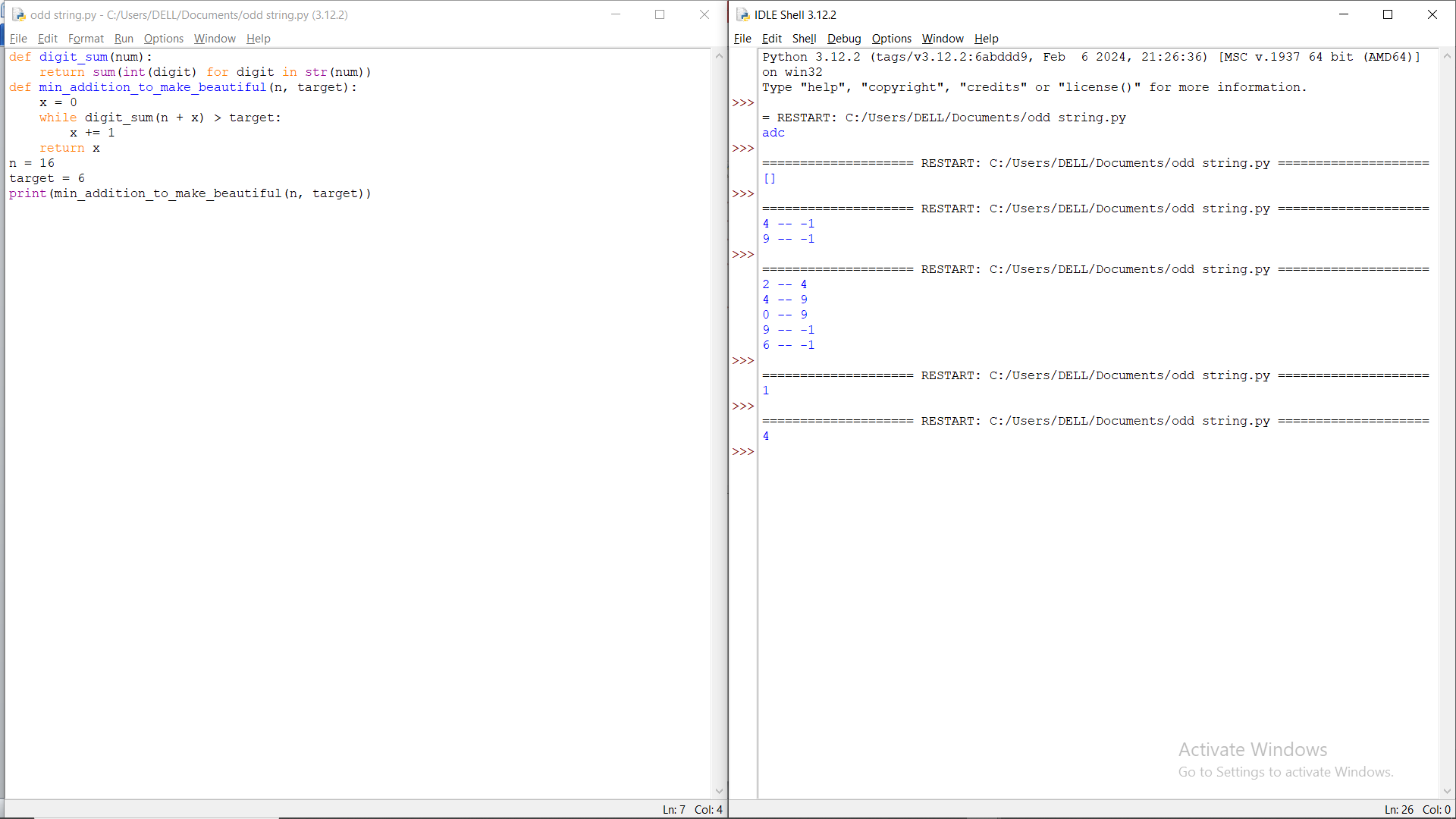
x += 1

return x

n = 16

target = 6

print(min\_addition\_to\_make\_beautiful(n, target))



1. Sort Array by Moving Items to Empty Space

def minOperationsToSort(nums):

n = len(nums)

out\_of\_place = 0

for i in range(1, n):

if nums[i] != i:

out\_of\_place += 1

return max(out\_of\_place - 1, 0)

nums = [1,0,2,4,3]

print(minOperationsToSort(nums))

