# **Name: Abdurrahman Qureshi**

# **Roll No: 242466**

Practical No: 4

**1) Remove all spaces from a sentence**

CODE:

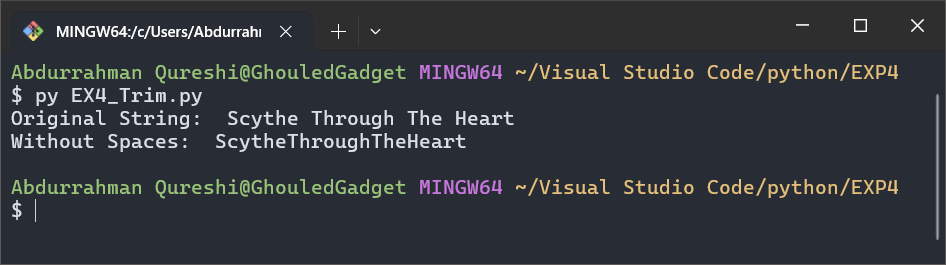
s = "Scythe Through The Heart"

print("Original String: " , s)

x = s.replace(" ","")

print("Without Spaces: ",x)

OUTPUT:



**2) Find the duplicate characters in a string**

CODE:

x = "You are a good man Arthur Morgan."

print("Original String:", x)

c = []

for i in x:

    if i not in c:

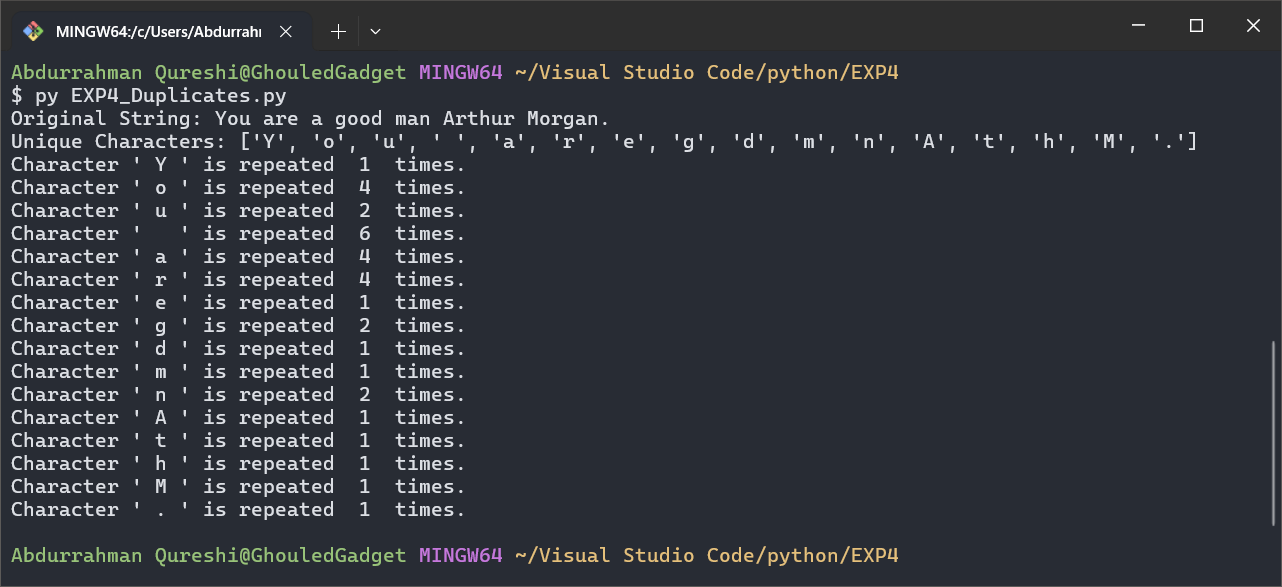
        c.append(i)

print("Unique Characters:", c)

for i in c:

    print("Character '" , i , "' is repeated " , x.count(i) , " times.")

OUTPUT:



**3) Check if a string is a palindrome**

CODE:

def isPalindrome(string\_x):

        if string\_x.lower() != string\_x[::-1].lower():

            return False

        return True

if (isPalindrome("Death is an old friend.")):

    print("The string 'Death is an old friend.' is a palindrome.")

else:

    print("The string 'Death is an old friend.' is not a palindrome")

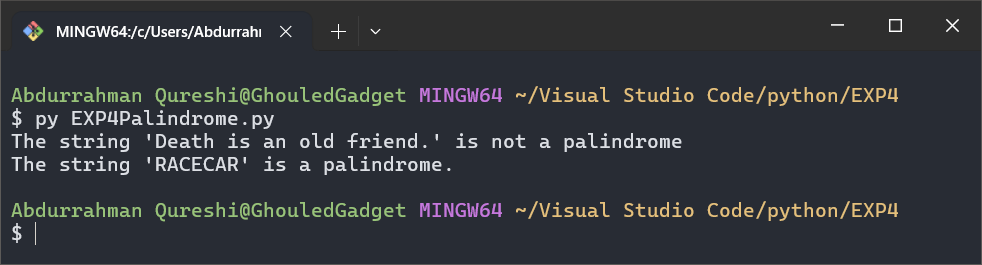
if (isPalindrome("RACECAR")):

    print("The string 'RACECAR' is a palindrome.")

else:

    print("The string 'RACECAR' is not a palindrome")

OUTPUT:



**4) Capitalize the first character of each word in a sentence.**

CODE:

x = "Welcome to the family son!"

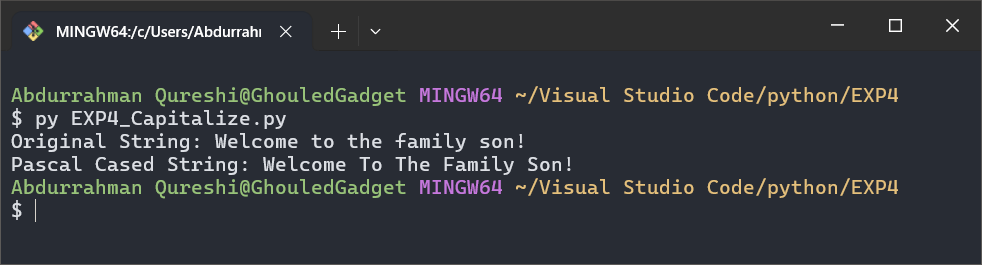
print("Original String:", x)

print("Pascal Cased String:", end=" ")

for i in x.split(" "):

    print(i.capitalize() , end=" ")

OUTPUT:



**5) Sort words alphabetically in a sentence**

CODE:

x = "Lycans and gentlemen, we thank you for waiting! And now let the games begin!"

y = x.split(" ")

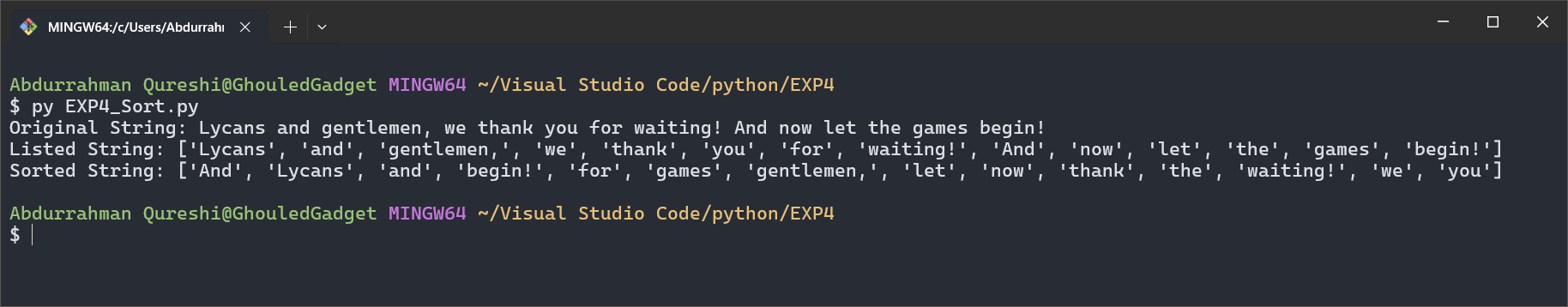
z = y.sort()

print("Original String:", x)

print("Listed String:", x.split(" "))

print("Sorted String:", y)

OUTPUT:



**6) Demonstrate 10 string functions**

CODE:

x = "50,000 people used to live here. Now it's a ghost town."

print("Original Sting: " , x)

print("Lowercase: " , x.lower())

print("Uppercase: " , x.upper())

print("Striping: " , x.strip())

print("Replacing ' ' with '\_': " , x.replace(" ", "\_"))

print("Splitting: " , x.split())

print("Joining '\_': ".join(x.split()))

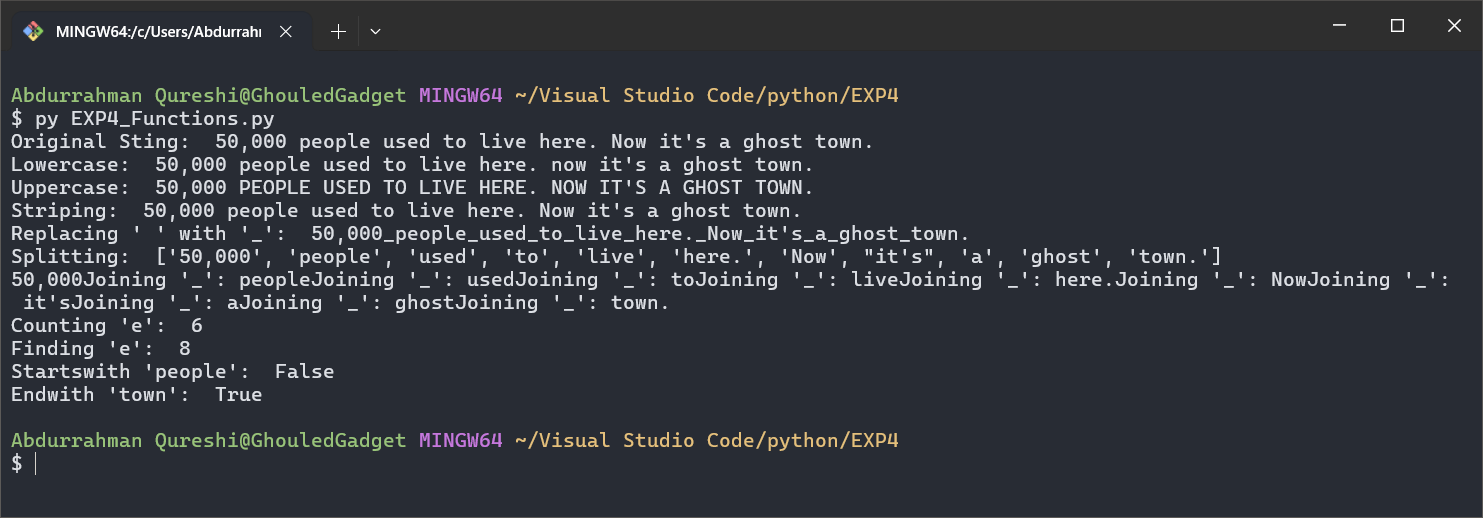
print("Counting 'e': " , x.count("e"))

print("Finding 'e': " , x.find("e"))

print("Startswith 'people': " , x.startswith("people"))

print("Endwith 'town': " , x.endswith("town."))

OUTPUT:



**7) Count the number of vowels in a string**

CODE:

x = "The Numbers Mason, What Do They Mean!"

print("Original String:", x)

print("Number of 'a' in the original string: ", x.count("a"))

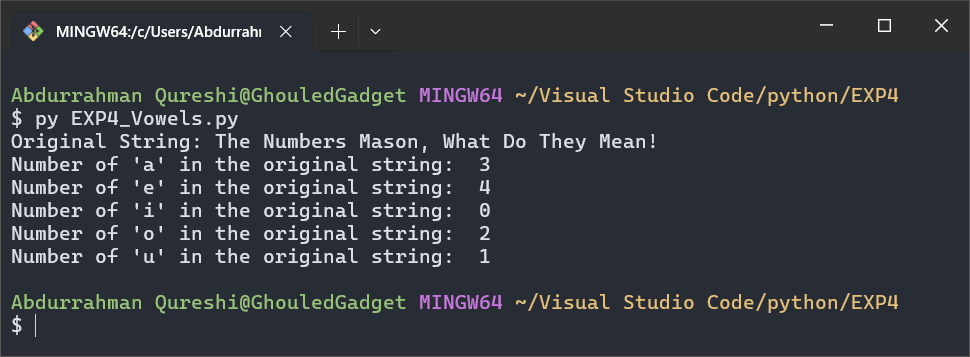
print("Number of 'e' in the original string: ", x.count("e"))

print("Number of 'i' in the original string: ", x.count("i"))

print("Number of 'o' in the original string: ", x.count("o"))

print("Number of 'u' in the original string: ", x.count("u"))

OUTPUT:



**8) Check if two strings are anagrams**

CODE:

x = "Captain Price"

y = "Ethan winters"

a = "listen"

b = "silent"

z = sorted(x.replace(" ","").lower()) == sorted(y.replace(" ","").lower())

c = sorted(a.replace(" ","").lower()) == sorted(b.replace(" ","").lower())

print("String X: ", x)

print("String Y: ", y)

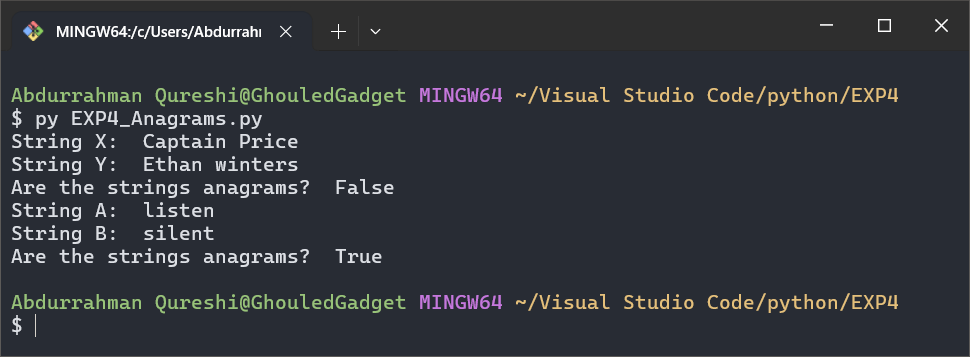
print("Are the strings anagrams? ", z)

print("String A: ", a)

print("String B: ", b)

print("Are the strings anagrams? ", c)

OUTPUT:



9) Write a Python program that demonstrates string slicing by extracting the first 5 characters, the last 5 characters, and reversing a string using slicing

CODE:

x = "Bishop Takes Rook"

a = x[:5]

b = x[-5:]

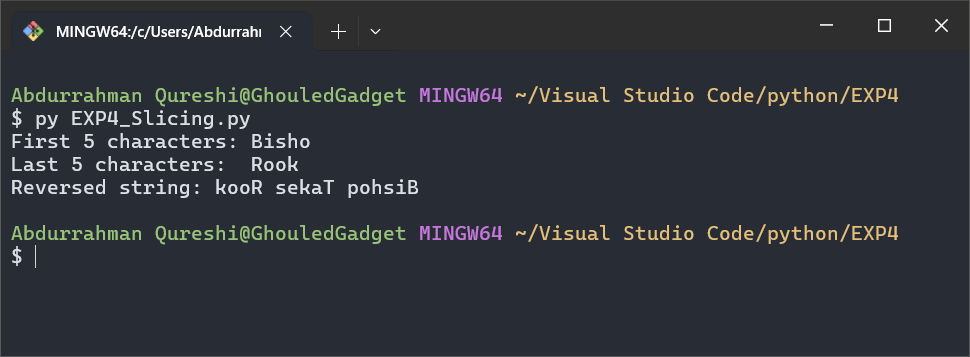
c = x[::-1]

print("First 5 characters:", a)

print("Last 5 characters:", b)

print("Reversed string:", c)

OUTPUT:



**10) Check if a substring is present in a string**

CODE:

x = "If it isn't Aloy, the Savior of Meridian, Anointed of the Nora!"

print("Original String:", x)

if x.find("Aloy") > 0:

    print("Does 'Aloy' exists in the string? True")

else:

    print("Does 'Aloy' exists in the string? False")

if x.find("Spectre") > 0:

    print("Does 'Spectre' exists in the string? True")

else:

    print("Does 'Spectre' exists in the string? False")

OUTPUT:



**11) Convert a string to uppercase or lowercase**

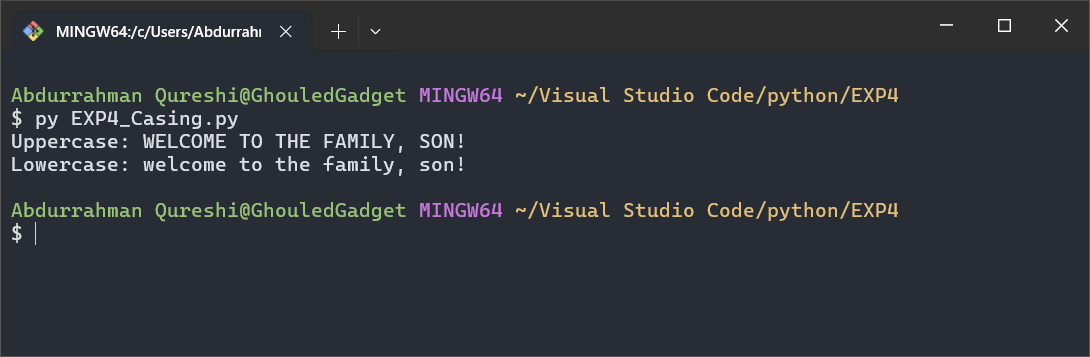
CODE:

x = "Welcome to the Family, Son!"

print("Uppercase:", x.upper())

print("Lowercase:", x.lower())

OUTPUT:



**12) Find the duplicate characters in a string**

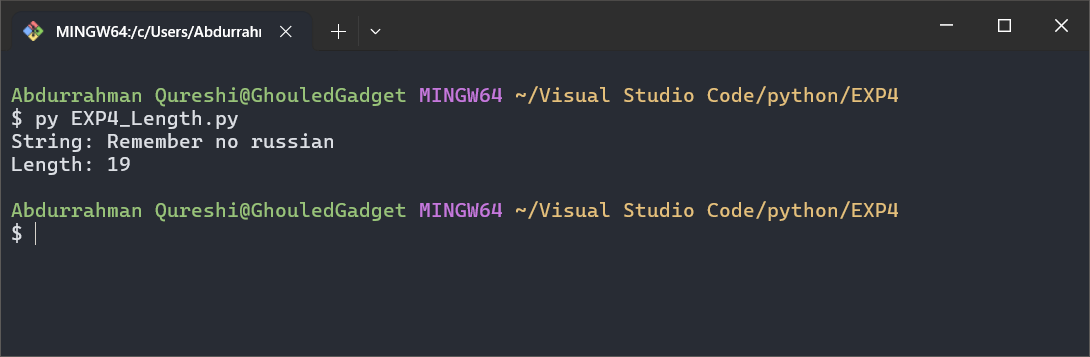
CODE:

x = "Remember no russian"

print("String:", x)

print("Length:", len(x))

OUTPUT:



**13) Check if a string starts or ends with a certain substring**

CODE:

x = "Nothing is true, everything is permitted."

print("Original string:", x)

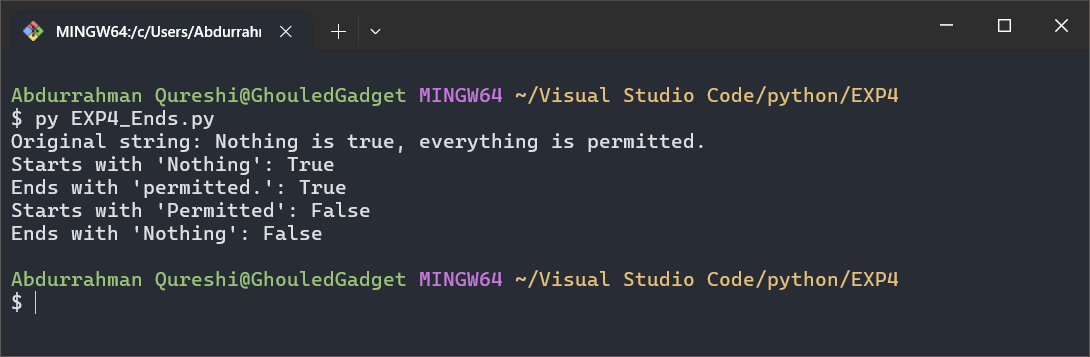
print("Starts with 'Nothing':", x.startswith("Nothing"))

print("Ends with 'permitted.':", x.endswith("permitted."))

print("Starts with 'Permitted':", x.startswith("Permitted"))

print("Ends with 'Nothing':", x.endswith("Nothing"))

OUTPUT:



**14) Count the number of occurrences of a substring.**

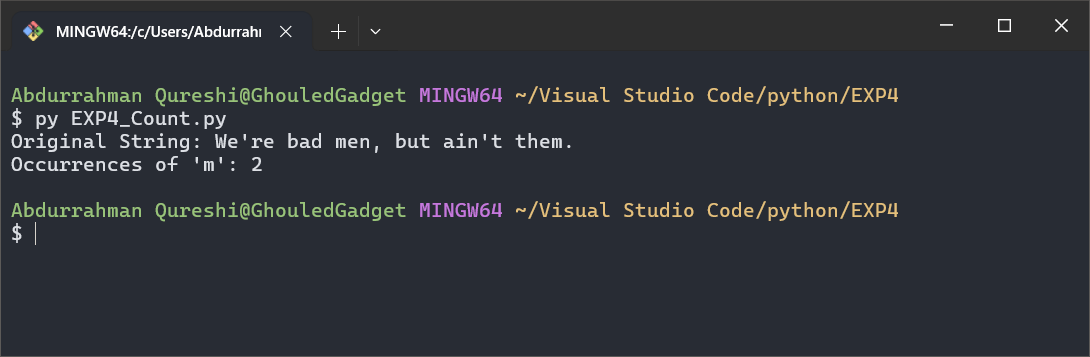
CODE:

x = "We're bad men, but ain't them."

print("Original String: " + x)

print("Occurrences of 'm':", x.count("m"))

OUTPUT:



**15) Sort the words in a sentence**

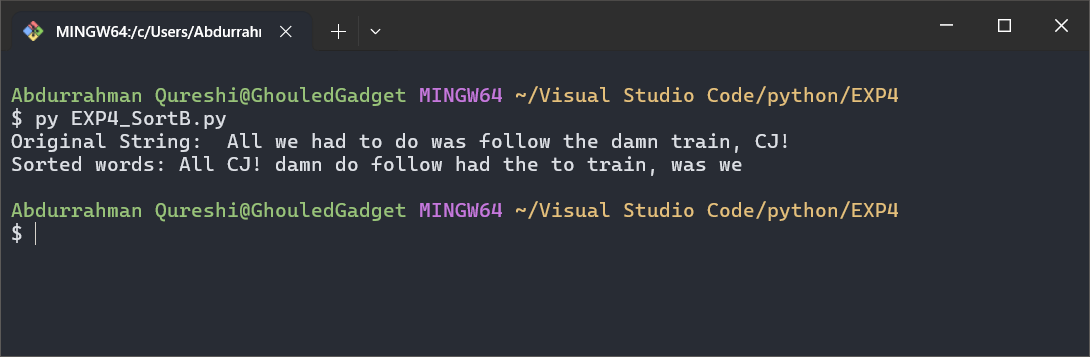
CODE:

x = "All we had to do was follow the damn train, CJ!"

print("Original String: ", x)

print("Sorted words:", " ".join(sorted(x.split(), key=str.lower)))

OUTPUT:



**16) Implement a Caesar cipher**

CODE:

def ceaser\_cipher(input, key, type):

    characters = ["A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K","M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z", "a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k", "l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z", "1", "2", "3", "4", "5", "6", "7", "8", "9", "0", "/", "+", " "]

    base64 = ""

    for char in input:

        if(type == True):

            base64 += characters[(characters.index(char) + key) % len(characters)]

        else:

            base64 += characters[(characters.index(char) - key) % len(characters)]

    return base64

x = input("Enter a string: ")

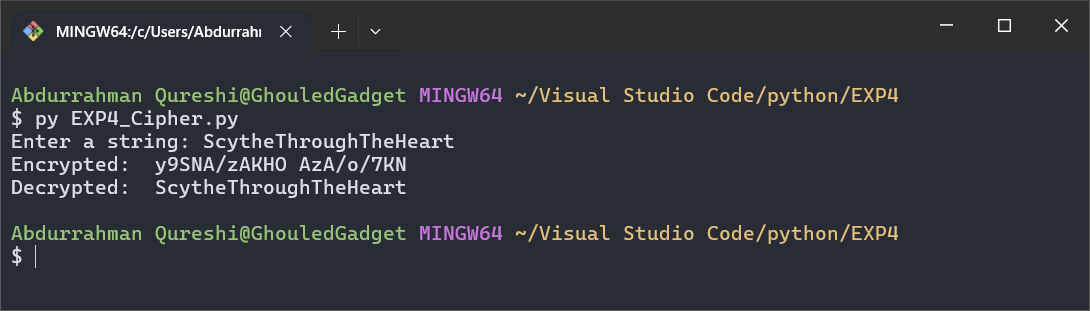
encrypted = ceaser\_cipher(x, 32, True)

print("Encrypted: " ,encrypted)

decrypted = ceaser\_cipher(encrypted, 32, False)

print("Decrypted: " ,decrypted)

OUTPUT:



**17) Write a Python program that randomly selects a word from a list, shuffles its characters, and asks the user to guess the original word**

CODE:

import random

words = ["Captain Price", "Aloy", "Ethan Winters", "Trevor Philips", "Spectre", "Simon Riley", "John Marston", "Arthur Morgan"]

attempts = 0

score = 0

word = random.choice(words)

shuffled\_word = ''.join(random.sample(word, len(word)))

while True:

    attempts += 1

    print("\nAttempts:", attempts)

    print("Current Score:", score)

    print("Shuffled Word:", shuffled\_word)

    guess = input("Enter your guess: ")

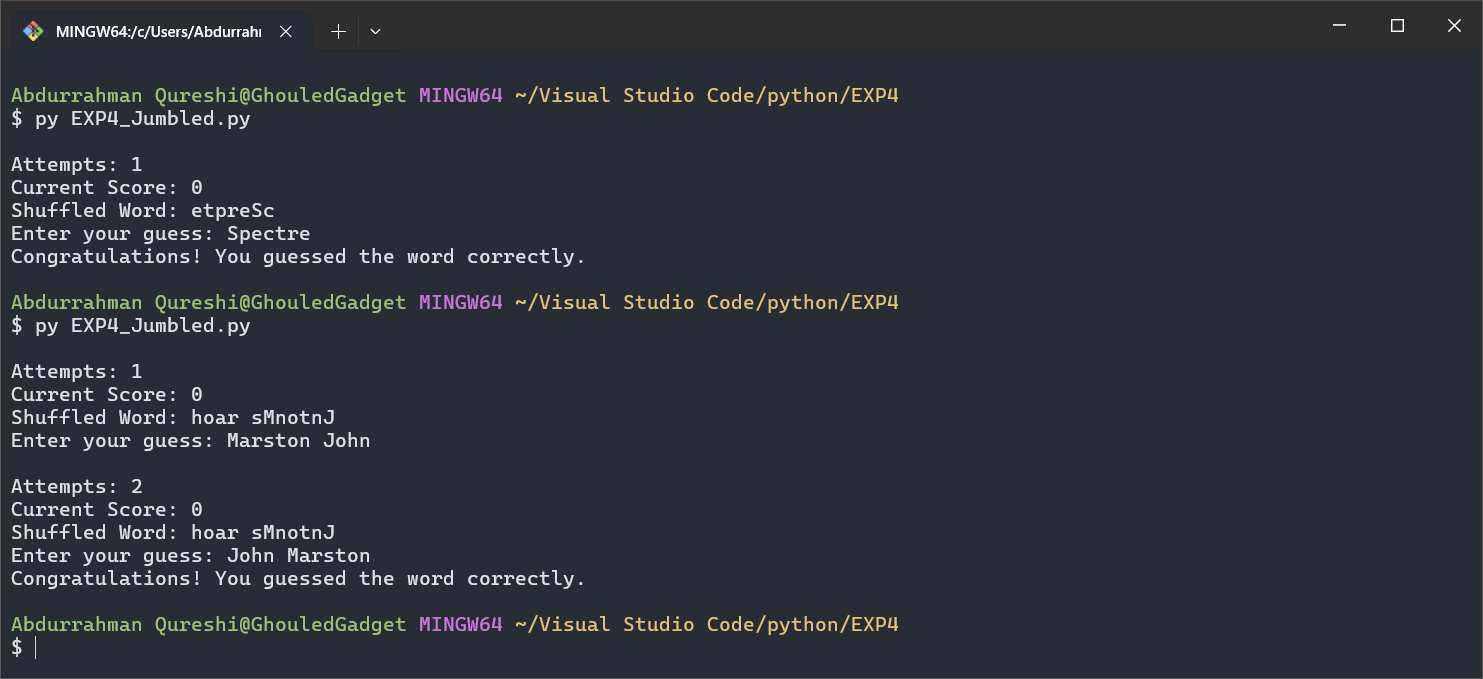
    if guess == word:

        print("Congratulations! You guessed the word correctly.")

        score += 1

        break

OUTPUT:



**18) Count the frequency of each character in a string via (dictionary)**

CODE:

x = "Why did I move here? I guess it was the weather."

print("Original String:", x)

d = {}

for i in x:

    if i not in d:

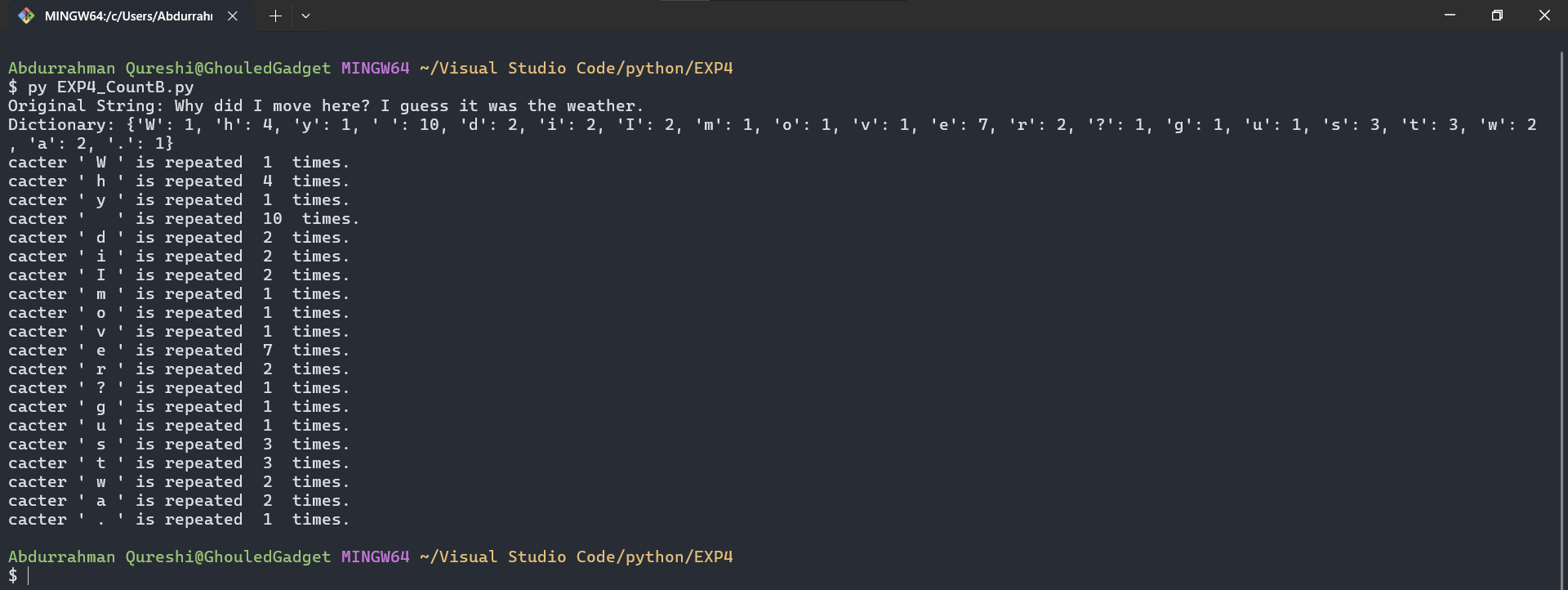
        d[i] = x.count(i)

print("Dictionary:", d)

for c in d:

    print("cacter '", c, "' is repeated ", d[c], " times.")

OUTPUT:



19) Write a Python program that generates a "word cloud" by counting the frequency of each word in a given sentence or paragraph and then sorting the words based on their frequency. Display the most frequent words and their count. Use built-in string methods and split() to process the input text.

CODE:

x = "History Is Written By The Victor. History Is Filled With Liars. If He Lives, And We Die, His Truth Becomes Written And Ours Is Lost."

y = x.split(" ")

z = []

a = []

print(y)

for i in y:

    if i not in z:

        z.append(i)

print(z)

for i in z:

    count = y.count(i)

    if count > 1:

        a.append(i)

    print("The word '", i, "' appears ", count, " times.")

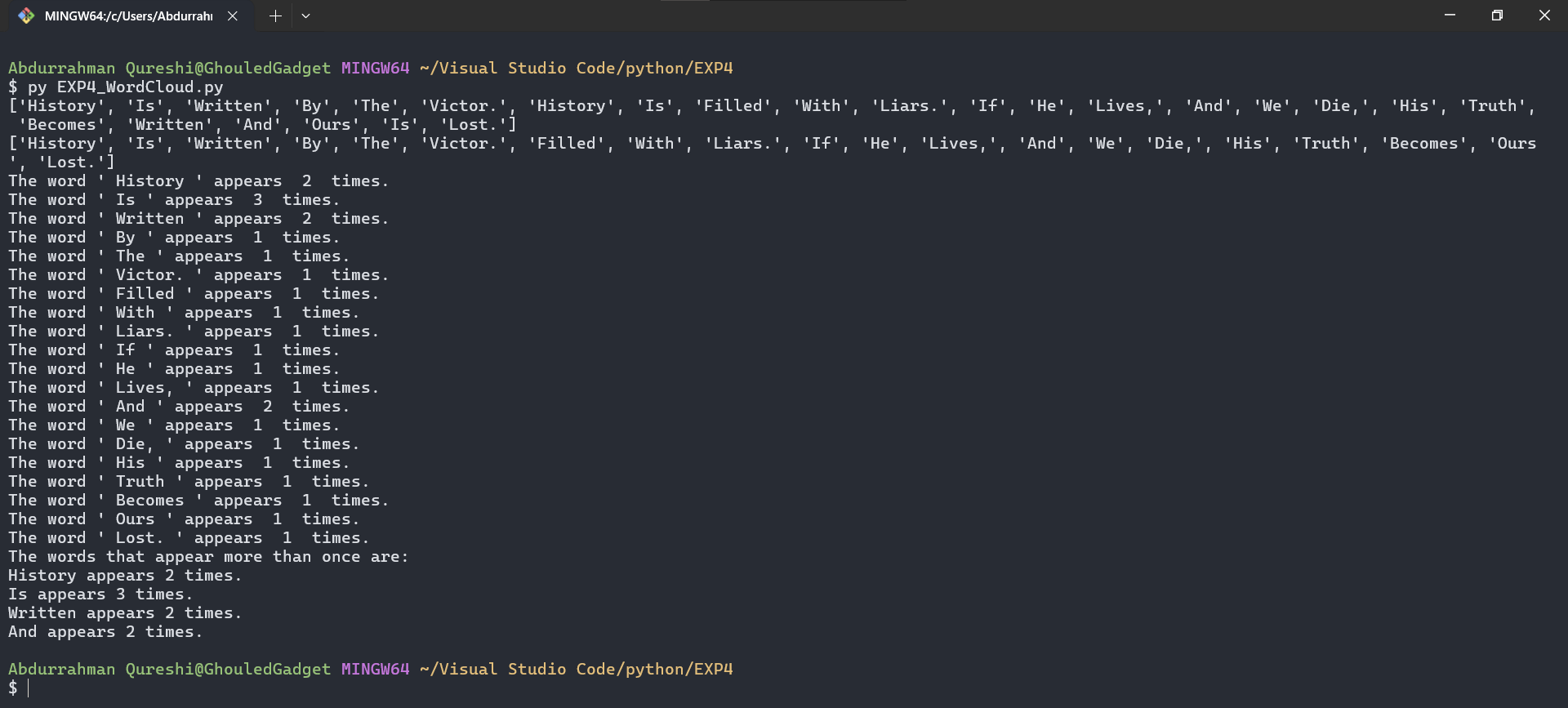
print("The words that appear more than once are:")

for i in a:

    count = y.count(i)

    print(i, "appears", count, "times.")

OUTPUT:



**20) Write a Python program that reverses each word in a given sentence without changing the order of the words. For example, for the input "Hello World", the output should be "olleH dlroW".**

CODE:

x = "Switching To Your Pistol Is Always Faster Than Reloading."

y = x.split(" ")

print("Original String: ", x)

print("Reversed Each word: ", end="")

for i in y:

    print(i[::-1], end=" ")

OUTPUT:

