1. Write a program that defines a function count\_lower\_upper() that accepts a string and calculates the number of uppercase and lowercase alphabets in it. It should return these values as a dictionary. Call this function for some sample strings.

def count\_lower\_upper(input\_string):

lower\_count = sum(1 for char in input\_string if char.islower())

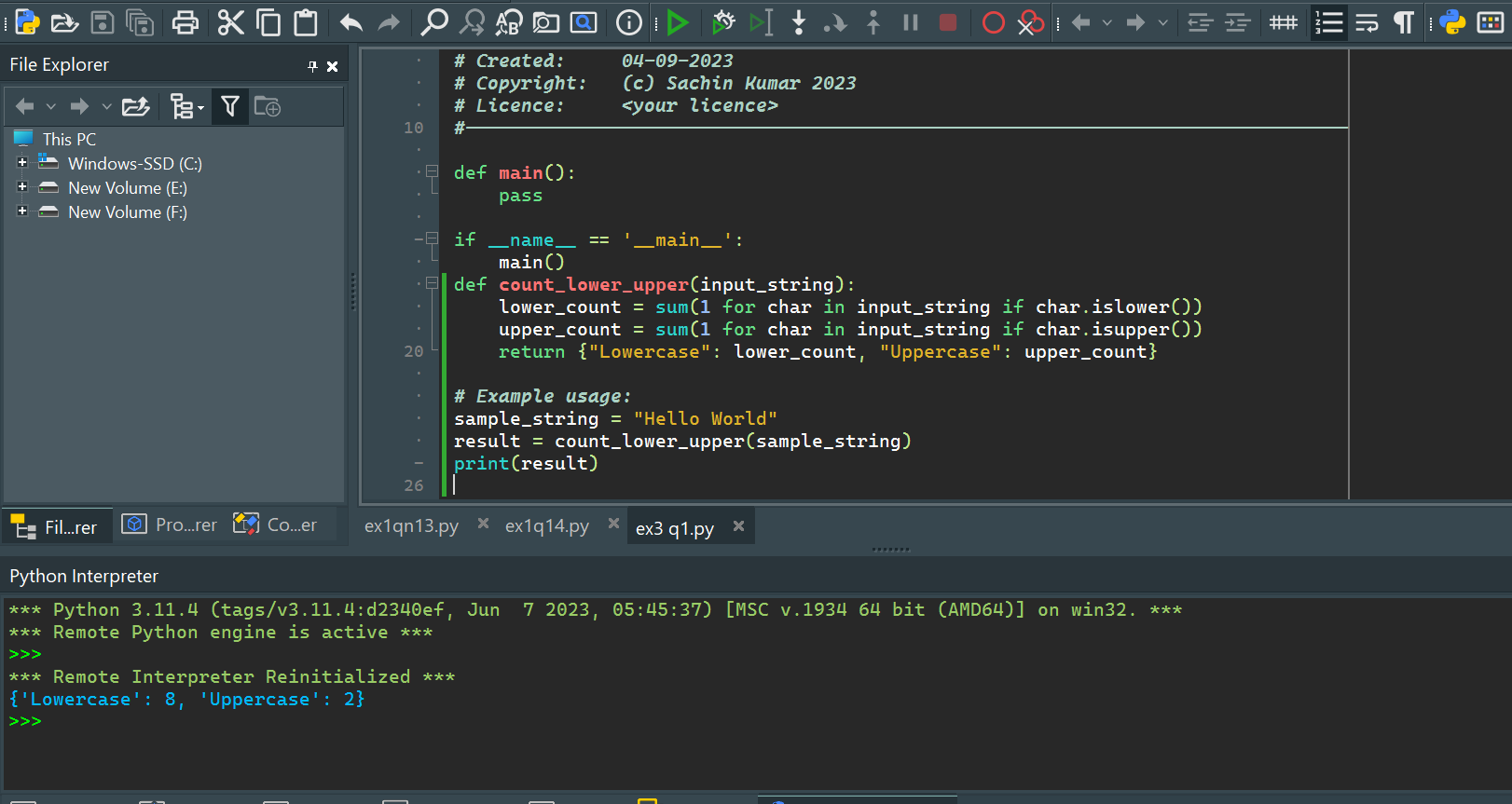
upper\_count = sum(1 for char in input\_string if char.isupper())

return {"Lowercase": lower\_count, "Uppercase": upper\_count}

# Example usage:

sample\_string = "Hello World"

result = count\_lower\_upper(sample\_string)

print(result)  


1. Write a program that defines a function compute() that calculates the value of n + nn + nnn + nnnn, where n is digit received by the function. Test the function for digits 4 and 7.

def compute(n):

return n + int(str(n) \* 2) + int(str(n) \* 3) + int(str(n) \* 4)

# Test the function for digits 4 and 7

digit1 = 4

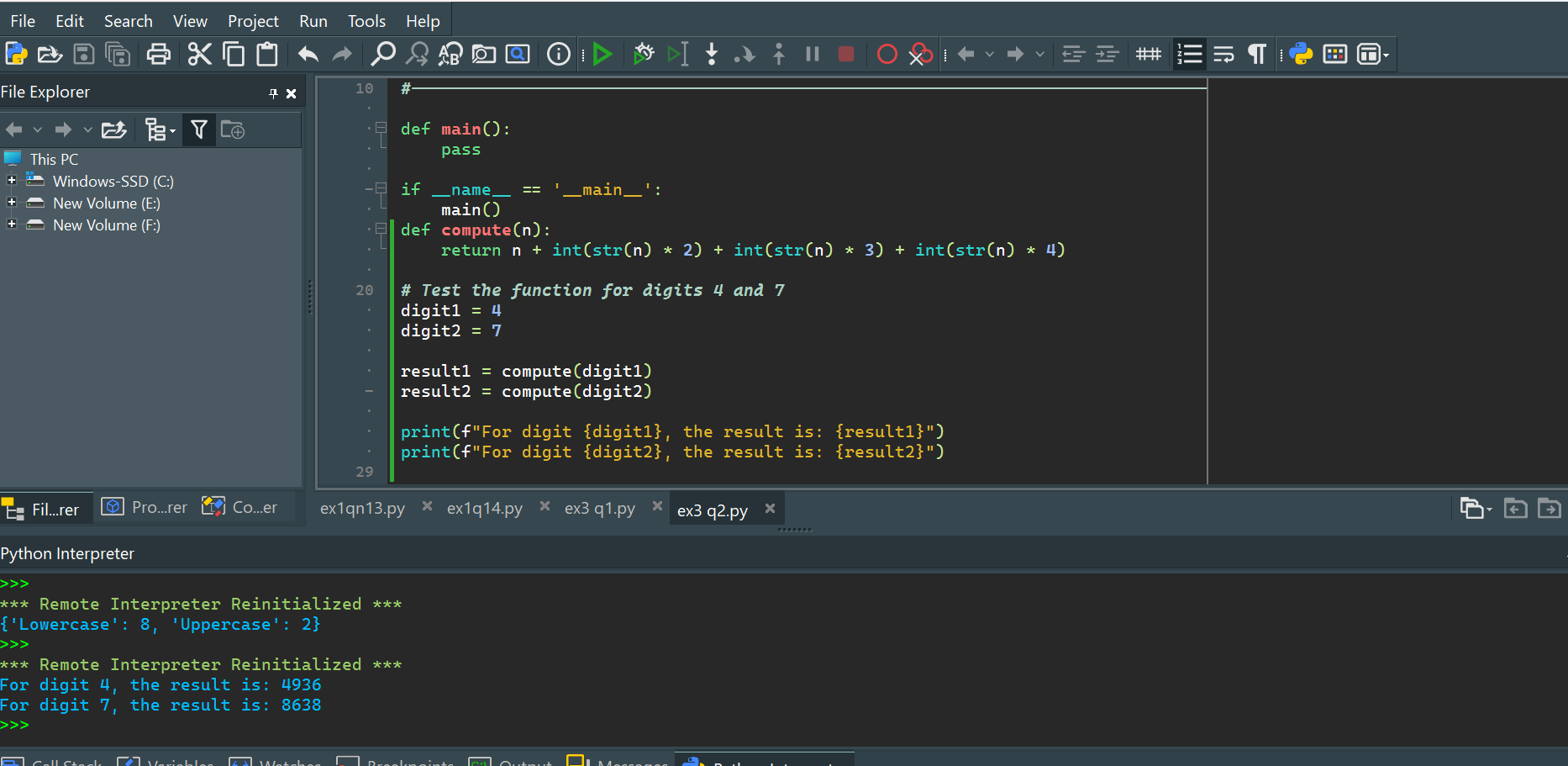
digit2 = 7

result1 = compute(digit1)

result2 = compute(digit2)

print(f"For digit {digit1}, the result is: {result1}")

print(f"For digit {digit2}, the result is: {result2}")



1. Write a program that defines a function create\_array() to create and return a 3D array whose dimensions are passed to the function. Also initialize each element of this array to a value passed to the function.

def create\_array(x, y, z, initial\_value):

# Create a 3D array filled with the initial value

arr = [[[initial\_value for \_ in range(z)] for \_ in range(y)] for \_ in range(x)]

return arr

# Example usage:

x\_dim = 3

y\_dim = 4

z\_dim = 2

initial\_value = 0

result\_array = create\_array(x\_dim, y\_dim, z\_dim, initial\_value)

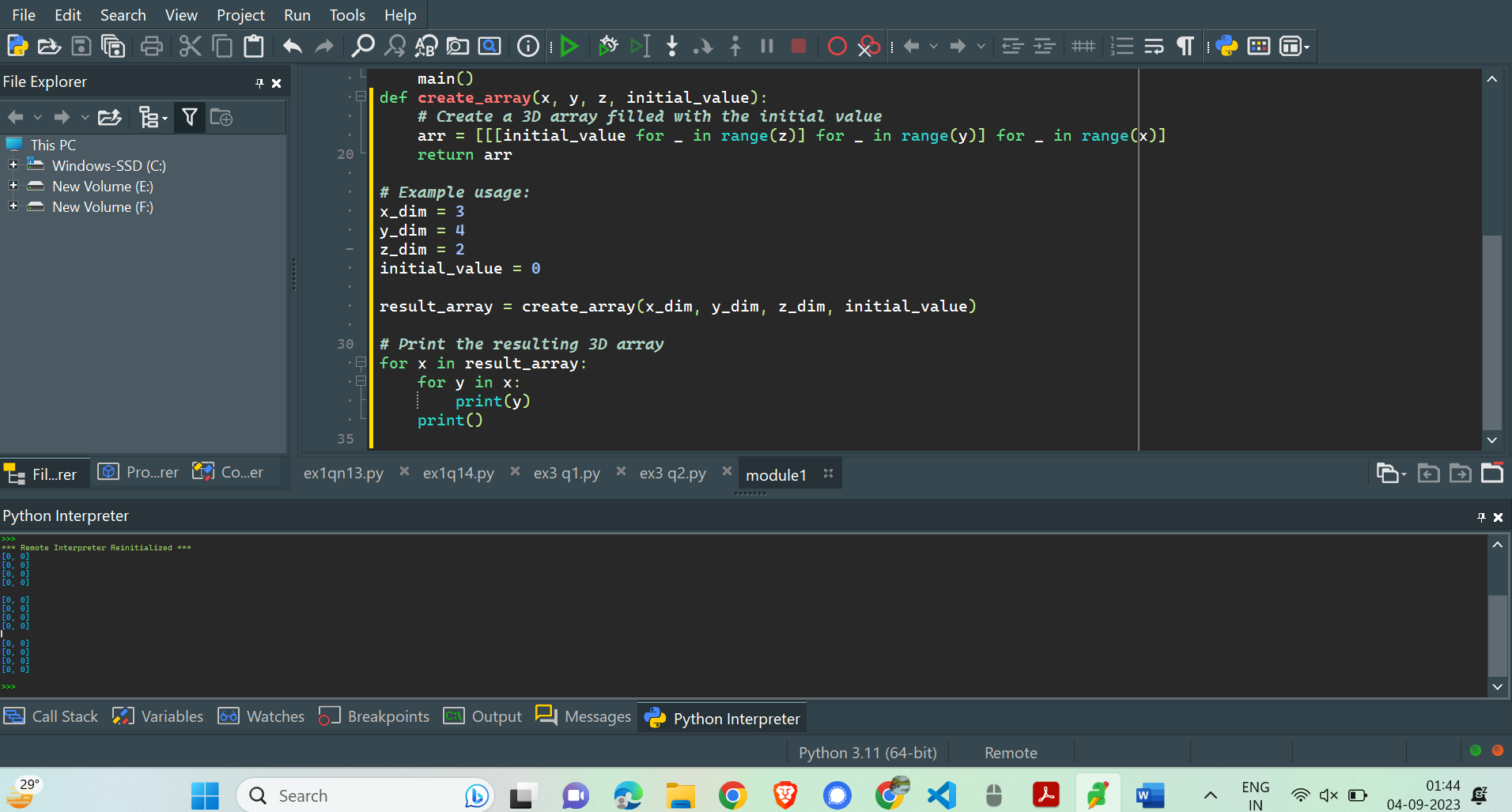
# Print the resulting 3D array

for x in result\_array:

for y in x:

print(y)

print()



1. Write a program that defines a function create\_list() to create and return a list which is an intersection of two lists passed to it.

def create\_list(list1, list2):

# Use set intersection to find the common elements

intersection = list(set(list1) & set(list2))

return intersection

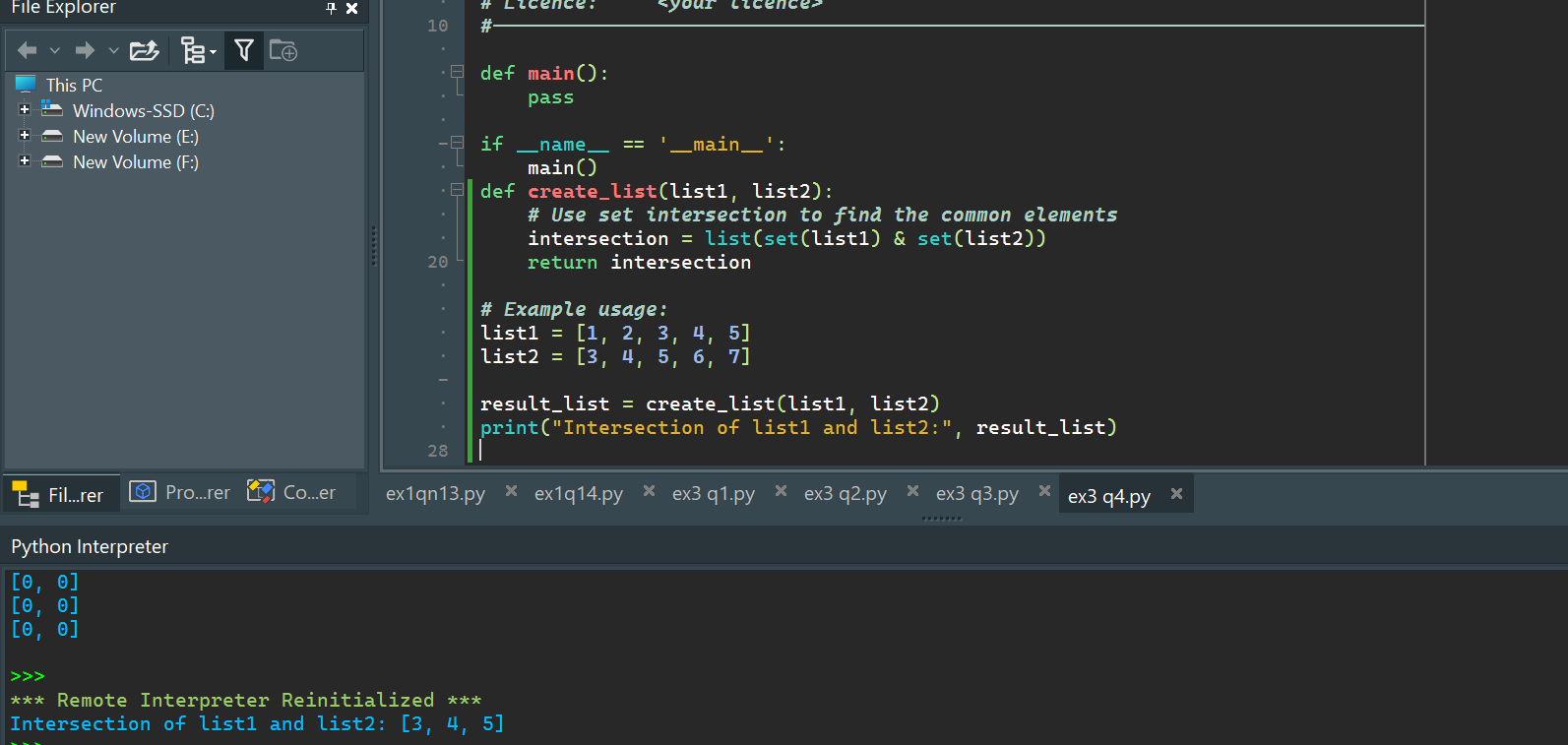
# Example usage:

list1 = [1, 2, 3, 4, 5]

list2 = [3, 4, 5, 6, 7]

result\_list = create\_list(list1, list2)

print("Intersection of list1 and list2:", result\_list)



1. Write a program that defines a function sanitize\_list() to remove all duplicate entries from the list that it receives.

def sanitize\_list(input\_list):

# Use set to remove duplicates, then convert back to list

sanitized\_list = list(set(input\_list))

return sanitized\_list

# Example usage:

original\_list = [1, 2, 2, 3, 4, 4, 5]

sanitized\_result = sanitize\_list(original\_list)

print("Original List:", original\_list)

print("Sanitized List (No Duplicates):", sanitized\_result)

