ASSIGNMENT-4

21.# Number of rows for the pattern

rows = 5

# Loop to iterate through each row

for i in range(1, rows + 1):

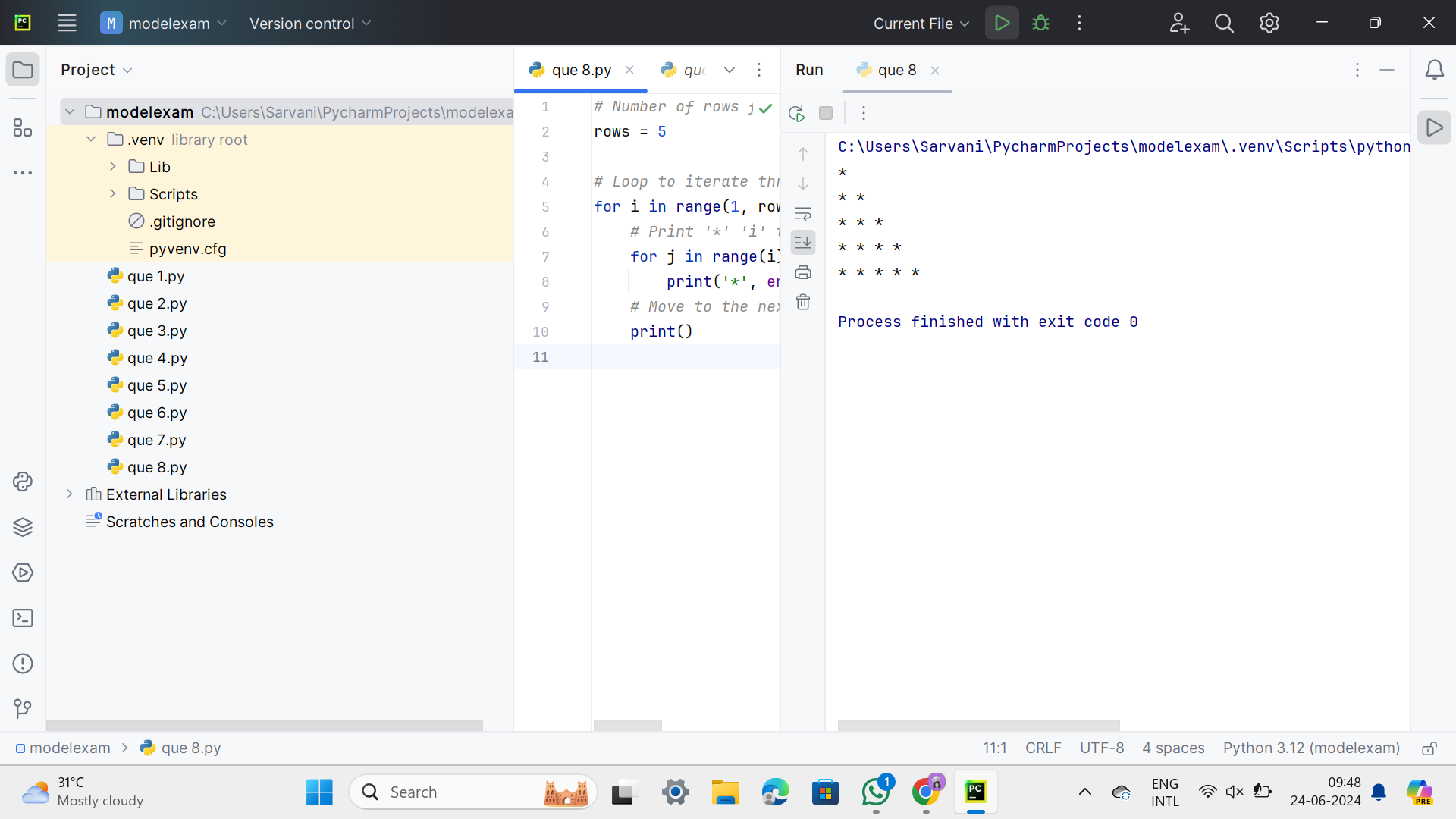
# Print '\*' 'i' times followed by a space

for j in range(i):

print('\*', end=' ')

# Move to the next line after each row

print()



22. def count\_vowels\_and\_consonants(statement):

vowels = "aeiouAEIOU"

num\_vowels = 0

num\_consonants = 0

for char in statement:

if char.isalpha():

if char in vowels:

num\_vowels += 1

else:

num\_consonants += 1

return num\_vowels, num\_consonants

def main():

statement = input("Enter a statement: ")

vowels\_count, consonants\_count = count\_vowels\_and\_consonants(statement)

print(f"Number of vowels: {vowels\_count}")

print(f"Number of consonants: {consonants\_count}")

if vowels\_count > consonants\_count:

print("Vowels are more than consonants.")

elif consonants\_count > vowels\_count:

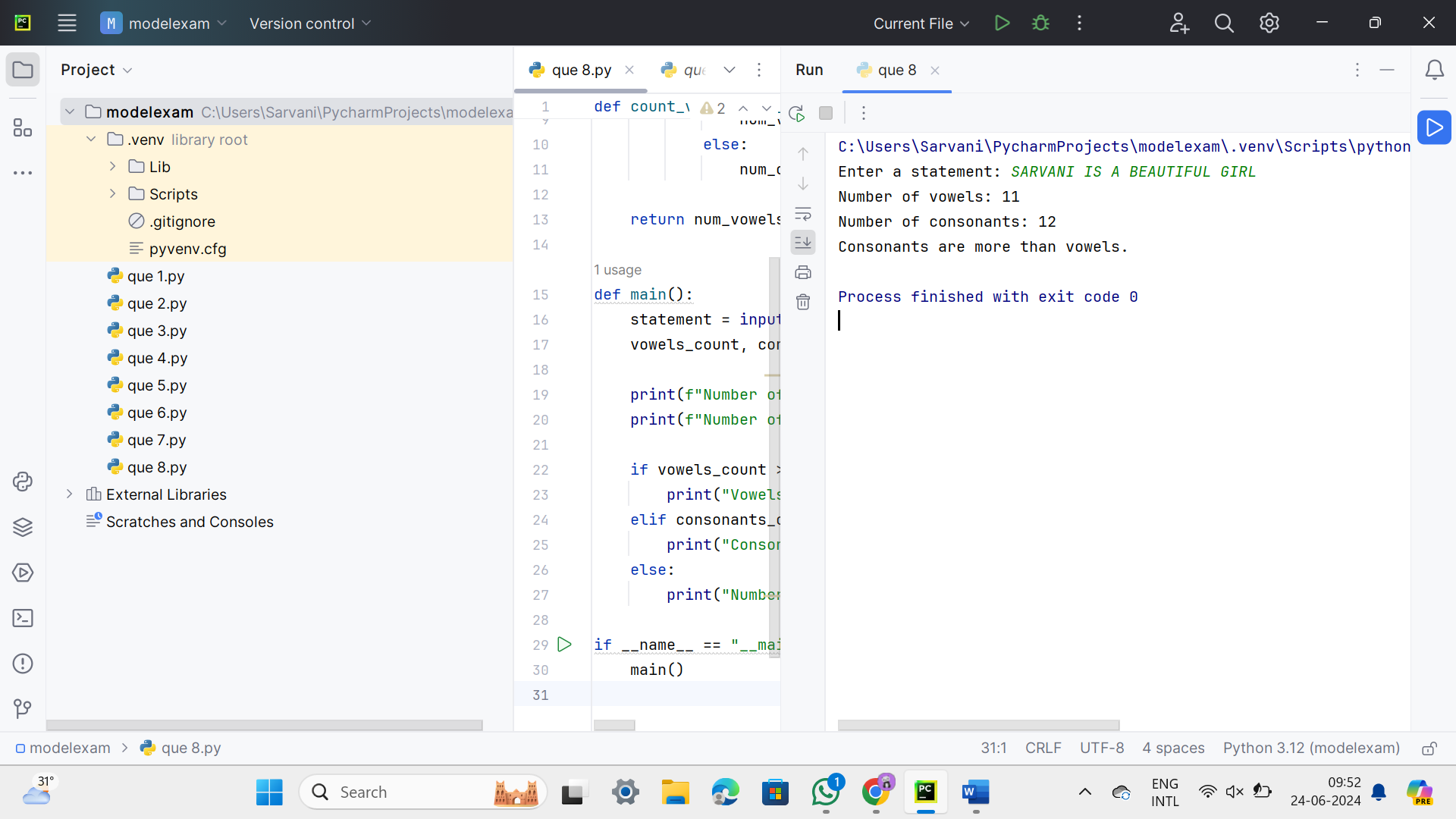
print("Consonants are more than vowels.")

else:

print("Number of vowels and consonants are equal.")

if \_\_name\_\_ == "\_\_main\_\_":

main()



23. def add(x, n):

return x + n

def main():

try:

# Get the input values from the user

x = int(input("Enter the value of X: "))

n = int(input("Enter the value of N: "))

# Get the choice from the user

choice = int(input("Enter your choice (1 for addition): "))

if choice == 1:

result = add(x, n)

print(f"Add(X, N) = {result}")

else:

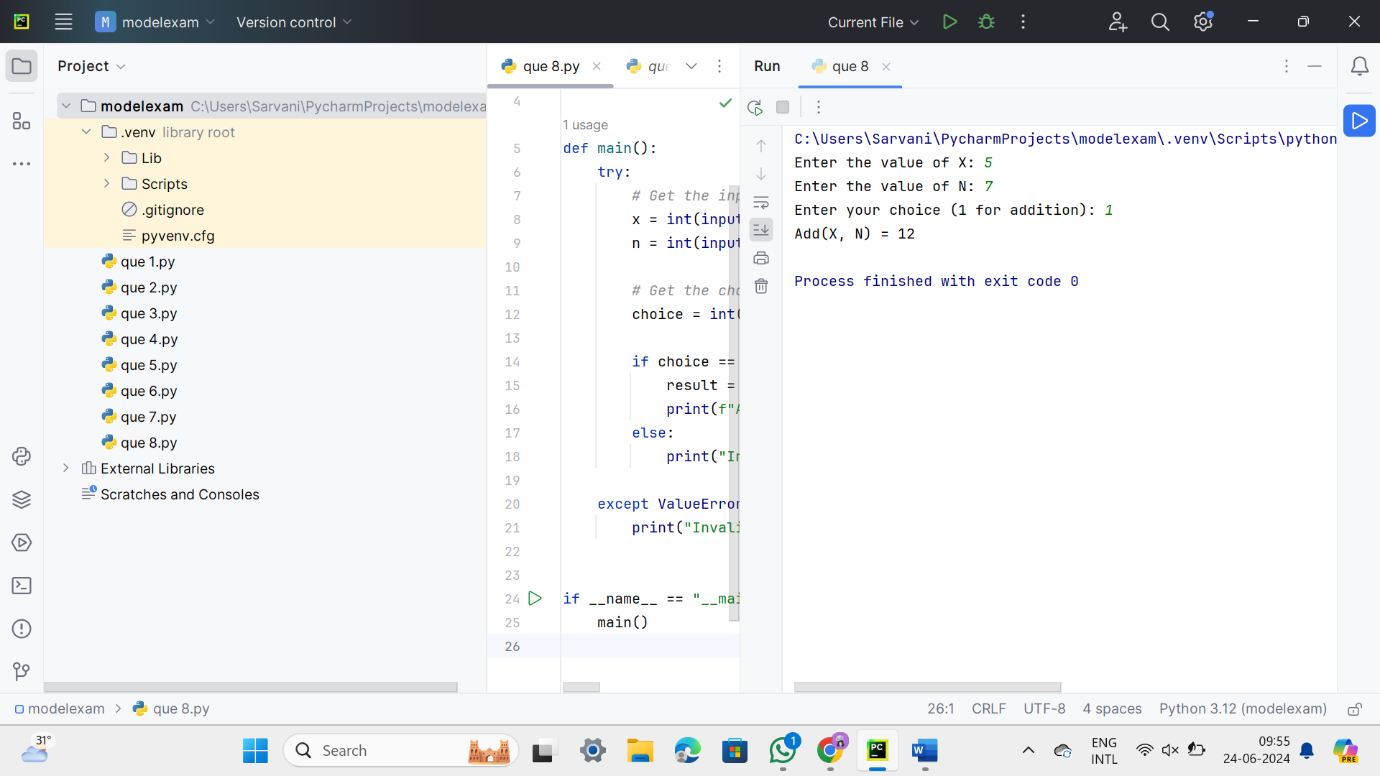
print("Invalid choice. Please enter 1 for addition.")

except ValueError:

print("Invalid input. Please enter integer values for X, N, and choice.")

if \_\_name\_\_ == "\_\_main\_\_":

main()



24. def calculate\_averages():

positive\_numbers = []

negative\_numbers = []

while True:

try:

number = float(input("Enter a number (-1 to stop): "))

if number == -1:

break

elif number > 0:

positive\_numbers.append(number)

elif number < 0:

negative\_numbers.append(number)

except ValueError:

print("Invalid input. Please enter a valid number.")

if positive\_numbers:

average\_positive = sum(positive\_numbers) / len(positive\_numbers)

else:

average\_positive = 0

if negative\_numbers:

average\_negative = sum(negative\_numbers) / len(negative\_numbers)

else:

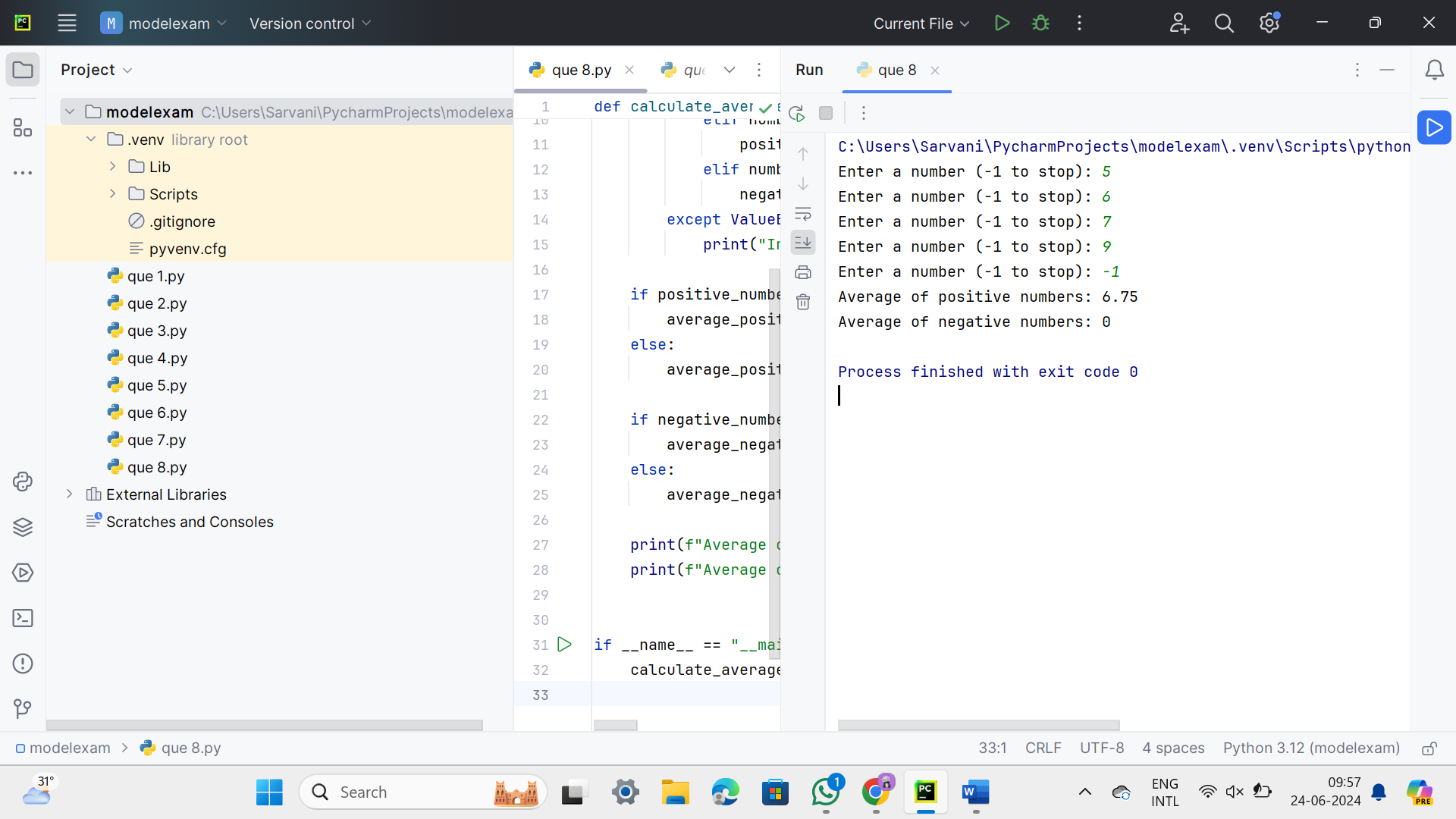
average\_negative = 0

print(f"Average of positive numbers: {average\_positive}")

print(f"Average of negative numbers: {average\_negative}")

if \_\_name\_\_ == "\_\_main\_\_":

calculate\_averages()



25. def reverse\_words\_in\_string(s):

# Split the string into words

words = s.split()

# Reverse the list of words

reversed\_words = words[::-1]

# Join the reversed words back into a string

reversed\_string = ' '.join(reversed\_words)

return reversed\_string

def main():

# Input the string from the user

input\_string = input("Enter a string: ")

# Call the function to reverse words in the string

reversed\_string = reverse\_words\_in\_string(input\_string)

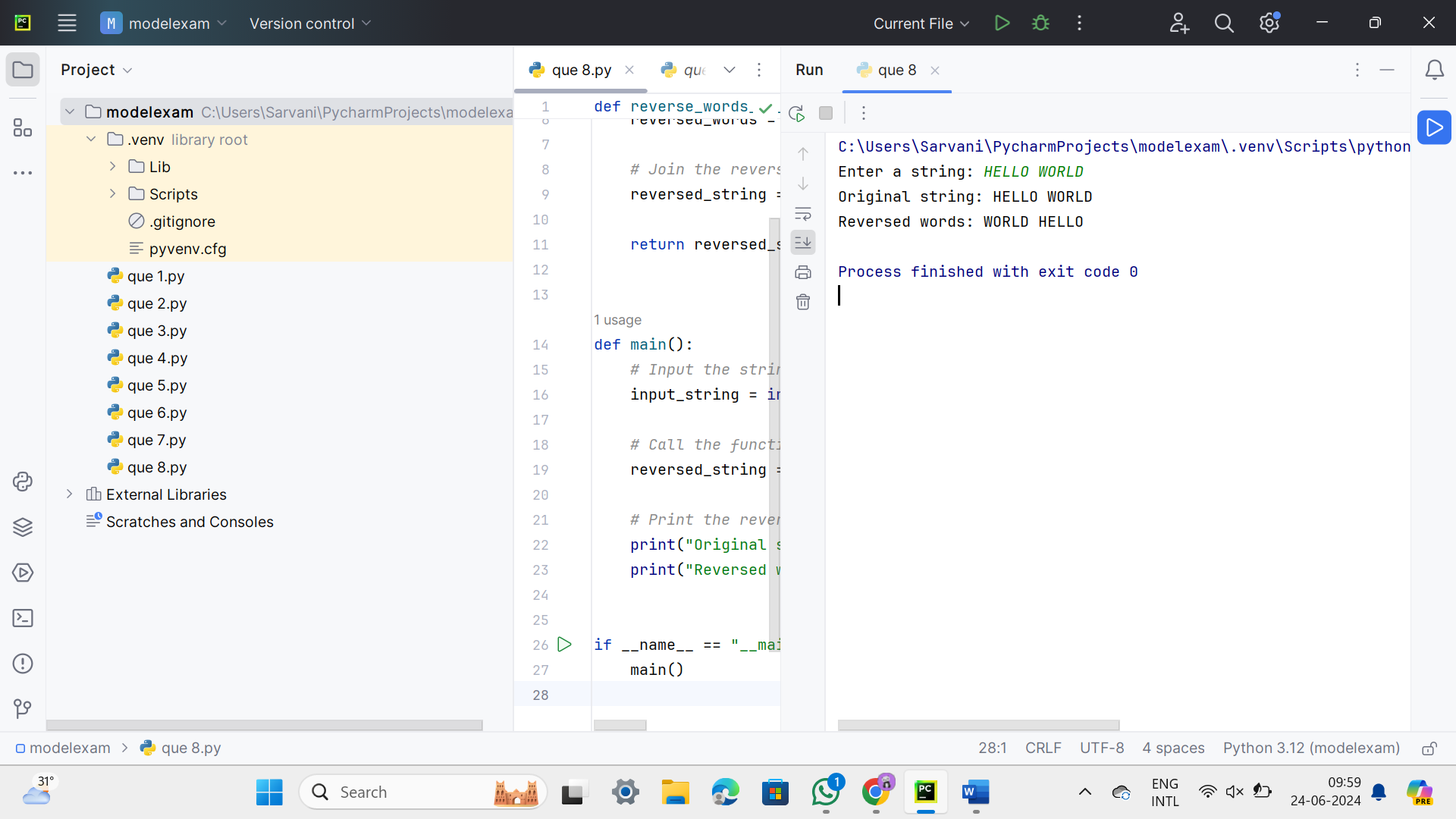
# Print the reversed string

print("Original string:", input\_string)

print("Reversed words:", reversed\_string)

if \_\_name\_\_ == "\_\_main\_\_":

main()



26. def count\_vowels\_and\_consonants(statement):

vowels = "aeiouAEIOU"

num\_vowels = 0

num\_consonants = 0

for char in statement:

if char.isalpha():

if char in vowels:

num\_vowels += 1

else:

num\_consonants += 1

return num\_vowels, num\_consonants

def main():

statement = input("Enter a statement: ")

vowels\_count, consonants\_count = count\_vowels\_and\_consonants(statement)

print(f"Number of vowels: {vowels\_count}")

print(f"Number of consonants: {consonants\_count}")

if vowels\_count > consonants\_count:

print("Vowels are more than consonants.")

elif consonants\_count > vowels\_count:

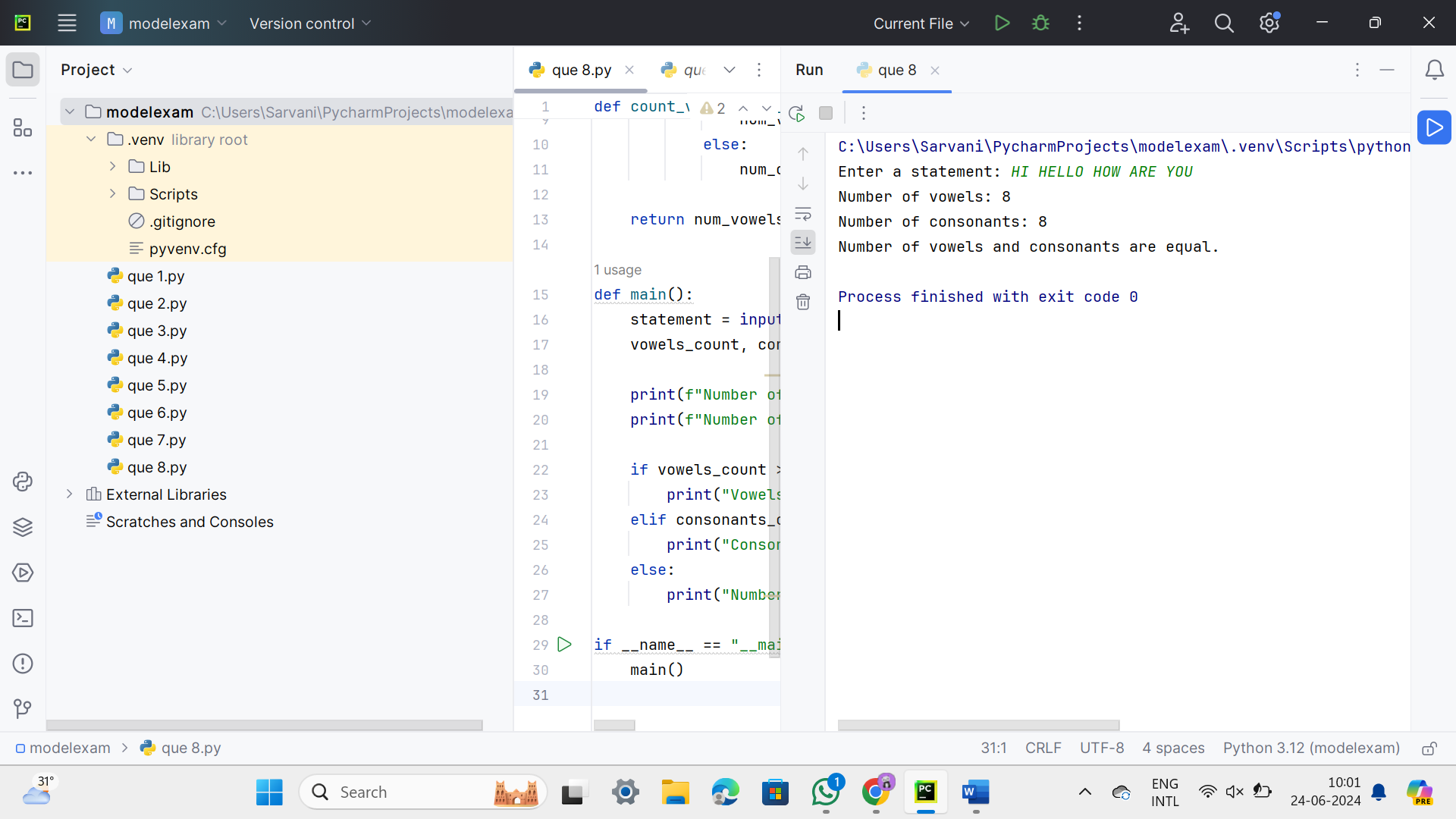
print("Consonants are more than vowels.")

else:

print("Number of vowels and consonants are equal.")

if \_\_name\_\_ == "\_\_main\_\_":

main()



27. def calculate\_tax(income):

if income <= 150000:

tax = 0

elif 150001 <= income <= 300000:

tax = (income - 150000) \* 0.1

elif 300001 <= income <= 500000:

tax = 15000 + (income - 300000) \* 0.2

else:

tax = 45000 + (income - 500000) \* 0.3

return tax

def main():

try:

income = float(input("Enter the income: "))

if income < 0:

print("Income should be a positive number.")

else:

tax = calculate\_tax(income)

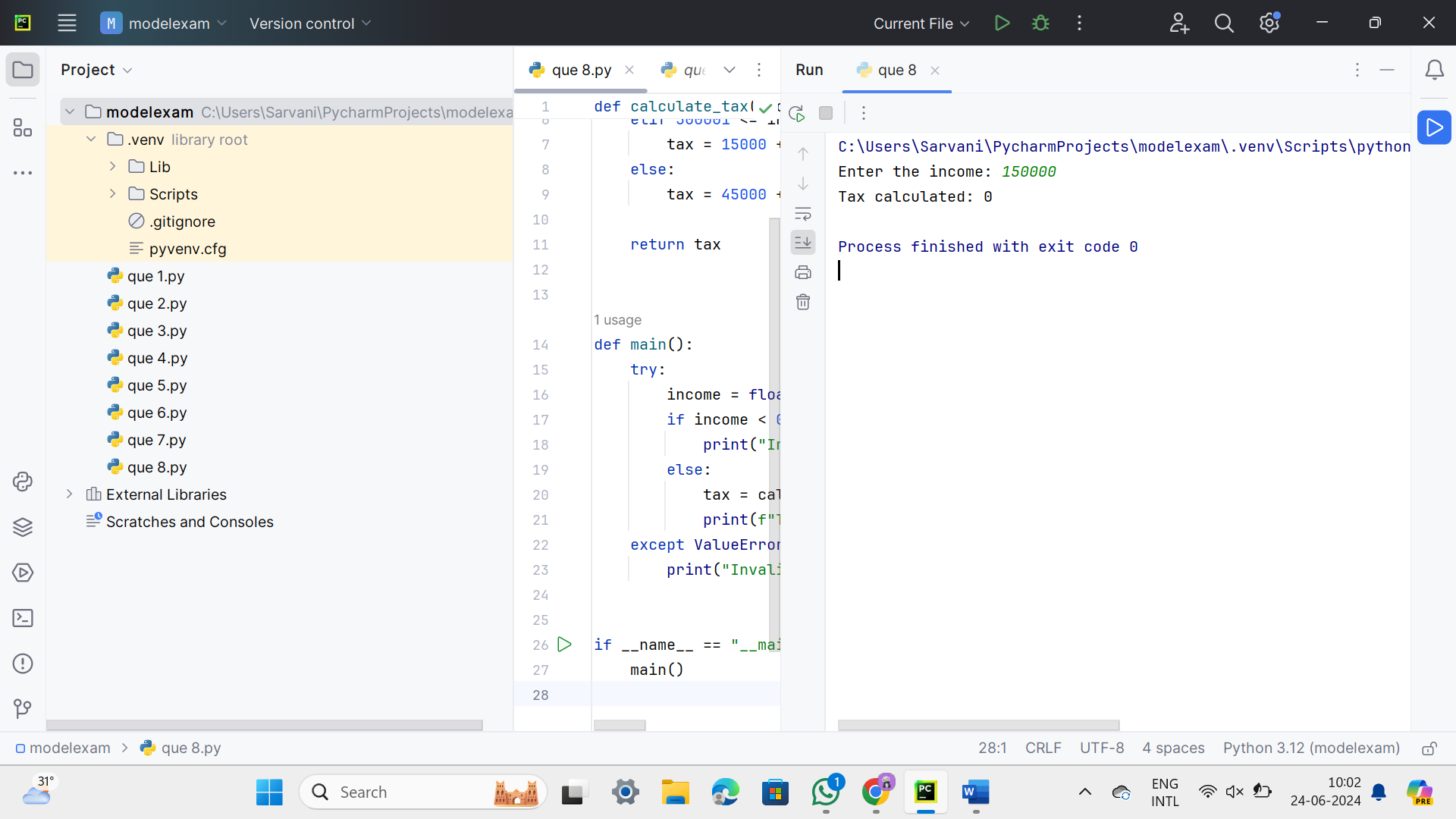
print(f"Tax calculated: {tax}")

except ValueError:

print("Invalid input. Please enter a valid number.")

if \_\_name\_\_ == "\_\_main\_\_":

main()



28.

def count\_spaces\_lines\_vowels\_consonants(filename):

vowels = "aeiouAEIOU"

space\_count = 0

line\_count = 0

vowel\_count = 0

consonant\_count = 0

try:

with open(filename, 'r') as file:

for line in file:

line\_count += 1

space\_count += line.count(' ')

for char in line:

if char.isalpha():

if char.lower() in vowels:

vowel\_count += 1

else:

consonant\_count += 1

elif char == ' ':

space\_count += 1

except FileNotFoundError:

print(f"File '{filename}' not found.")

return None

return space\_count, line\_count, vowel\_count, consonant\_count

def main():

filename = input("Enter the filename: ")

result = count\_spaces\_lines\_vowels\_consonants(filename)

if result:

space\_count, line\_count, vowel\_count, consonant\_count = result

print(f"Number of spaces: {space\_count}")

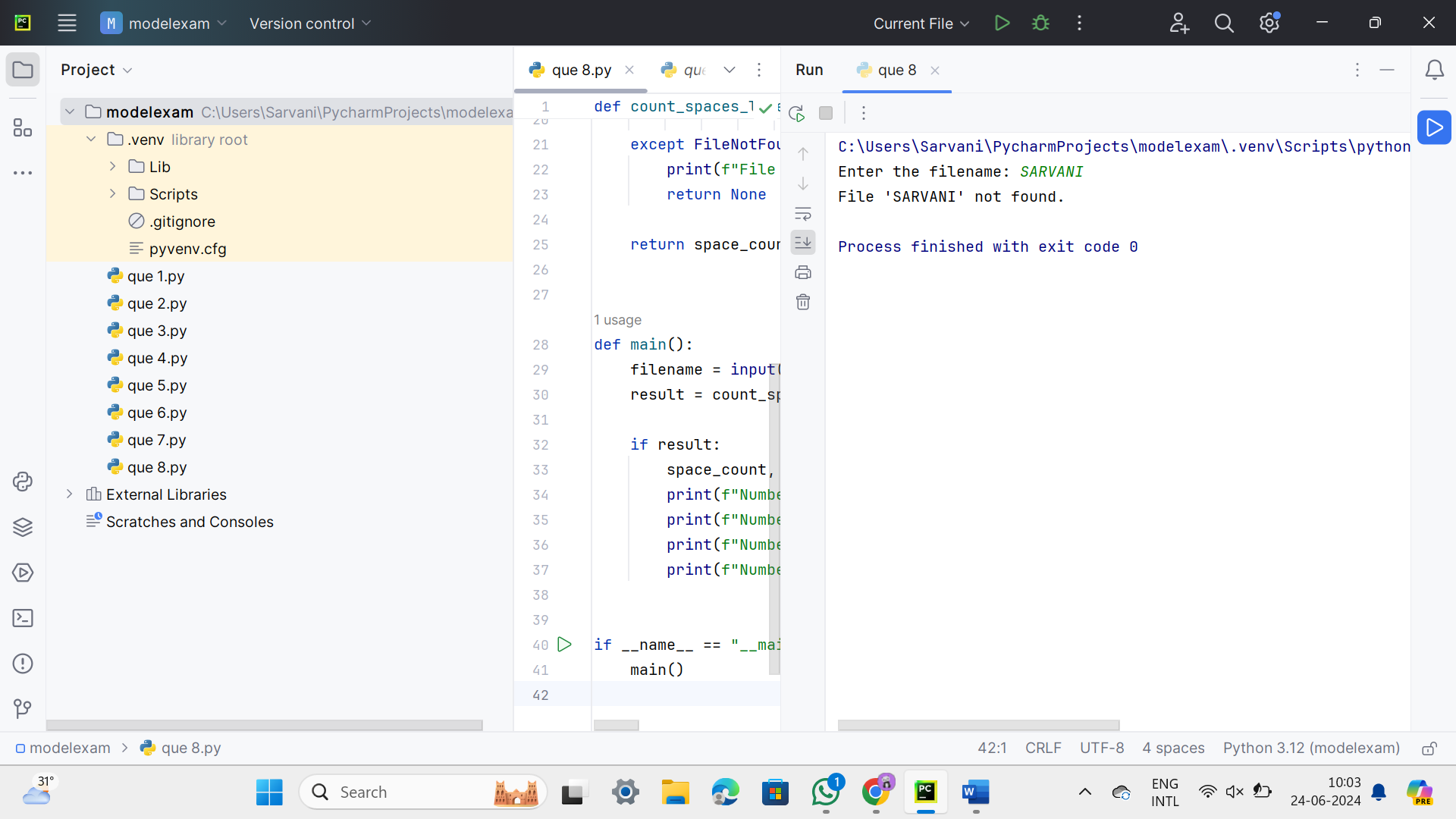
print(f"Number of lines: {line\_count}")

print(f"Number of vowels: {vowel\_count}")

print(f"Number of consonants: {consonant\_count}")

if \_\_name\_\_ == "\_\_main\_\_":

main()



29.

def print\_pattern(n):

for i in range(1, n + 1):

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

print(f"{j / 10:.1f}", end=" ")

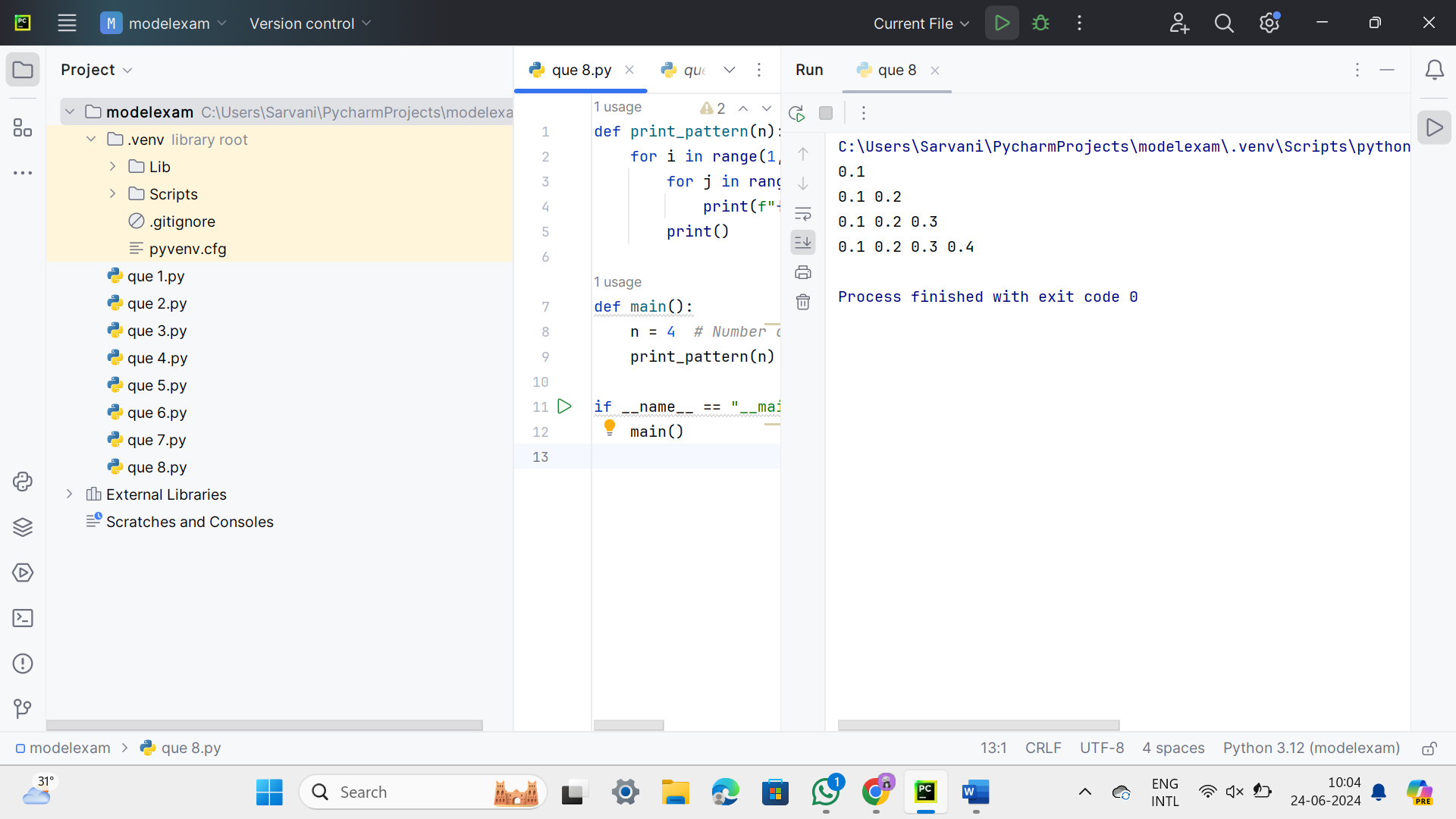
print()

def main():

n = 4 # Number of rows in the pattern

print\_pattern(n)

if \_\_name\_\_ == "\_\_main\_\_":

main()

30.

def matrix\_row\_sum(matrix):

row\_sums = []

for row in matrix:

row\_sum = sum(row)

row\_sums.append(row\_sum)

return row\_sums

def matrix\_column\_sum(matrix):

num\_rows = len(matrix)

num\_cols = len(matrix[0]) if matrix else 0

col\_sums = [0] \* num\_cols

for j in range(num\_cols):

for i in range(num\_rows):

col\_sums[j] += matrix[i][j]

return col\_sums

def matrix\_diagonal\_sum(matrix):

num\_rows = len(matrix)

num\_cols = len(matrix[0]) if matrix else 0

if num\_rows != num\_cols:

return None # Not a square matrix

primary\_diagonal\_sum = sum(matrix[i][i] for i in range(num\_rows))

secondary\_diagonal\_sum = sum(matrix[i][num\_rows - 1 - i] for i in range(num\_rows))

return primary\_diagonal\_sum, secondary\_diagonal\_sum

def main():

# Example matrix

matrix = [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9]

]

# Calculate row sums

row\_sums = matrix\_row\_sum(matrix)

print("Row sums:", row\_sums)

# Calculate column sums

col\_sums = matrix\_column\_sum(matrix)

print("Column sums:", col\_sums)

# Calculate diagonal sums

diagonal\_sums = matrix\_diagonal\_sum(matrix)

print("Primary diagonal sum:", diagonal\_sums[0])

print("Secondary diagonal sum:", diagonal\_sums[1])

if \_\_name\_\_ == "\_\_main\_\_":

main()

