NAME : BRIJIT ADAK

ROLL NO : 21052410

SEC : CSE 31

Q1. Guess a number game The computer picks a random number from 1 to 5, the player tries to guess. The player may have the flexibility to enter the number in an expression format.

import random

import re

def evaluate\_expression(expression):

try:

result = eval(expression)

return result

except:

return None

def main():

print("Guess the nummber between 1 to 5 game ")

secret\_number = random.randint(1,5)

player\_guess\_expression = input("Your guess (may be expression format): ")

# Check if the input is a valid expression

if re.match(r'^[1-5+\-\*/()\s]+$', player\_guess\_expression):

# Evaluate the expression

player\_guess = evaluate\_expression(player\_guess\_expression)

if player\_guess is not None:

if player\_guess == secret\_number:

print("Congratulations! You guessed the correct number.")

else:

print(f'Incorrect guess, correct -> {secret\_number}')

else:

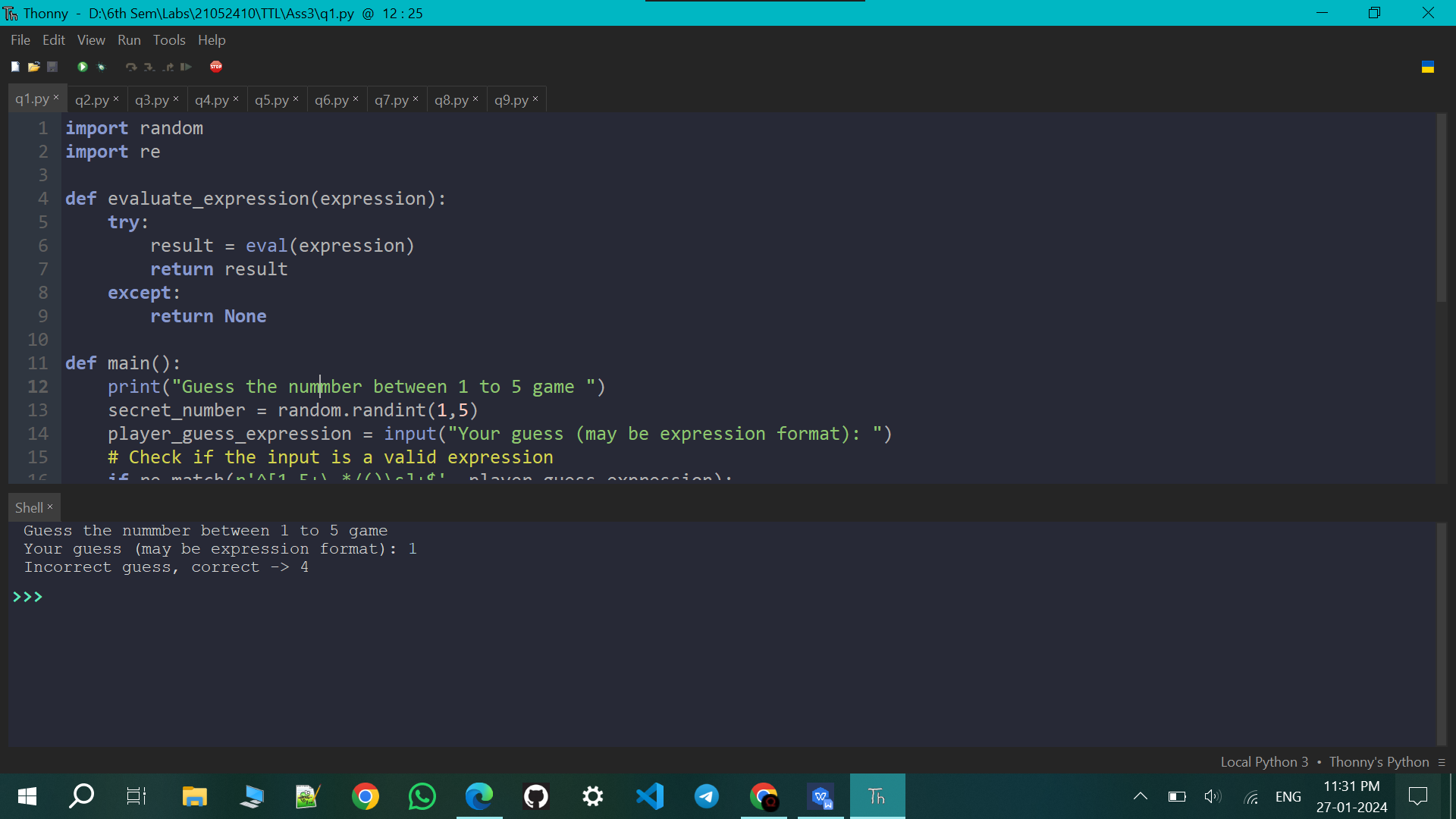
print("Invalid expression. Please enter a valid mathematical expression.")

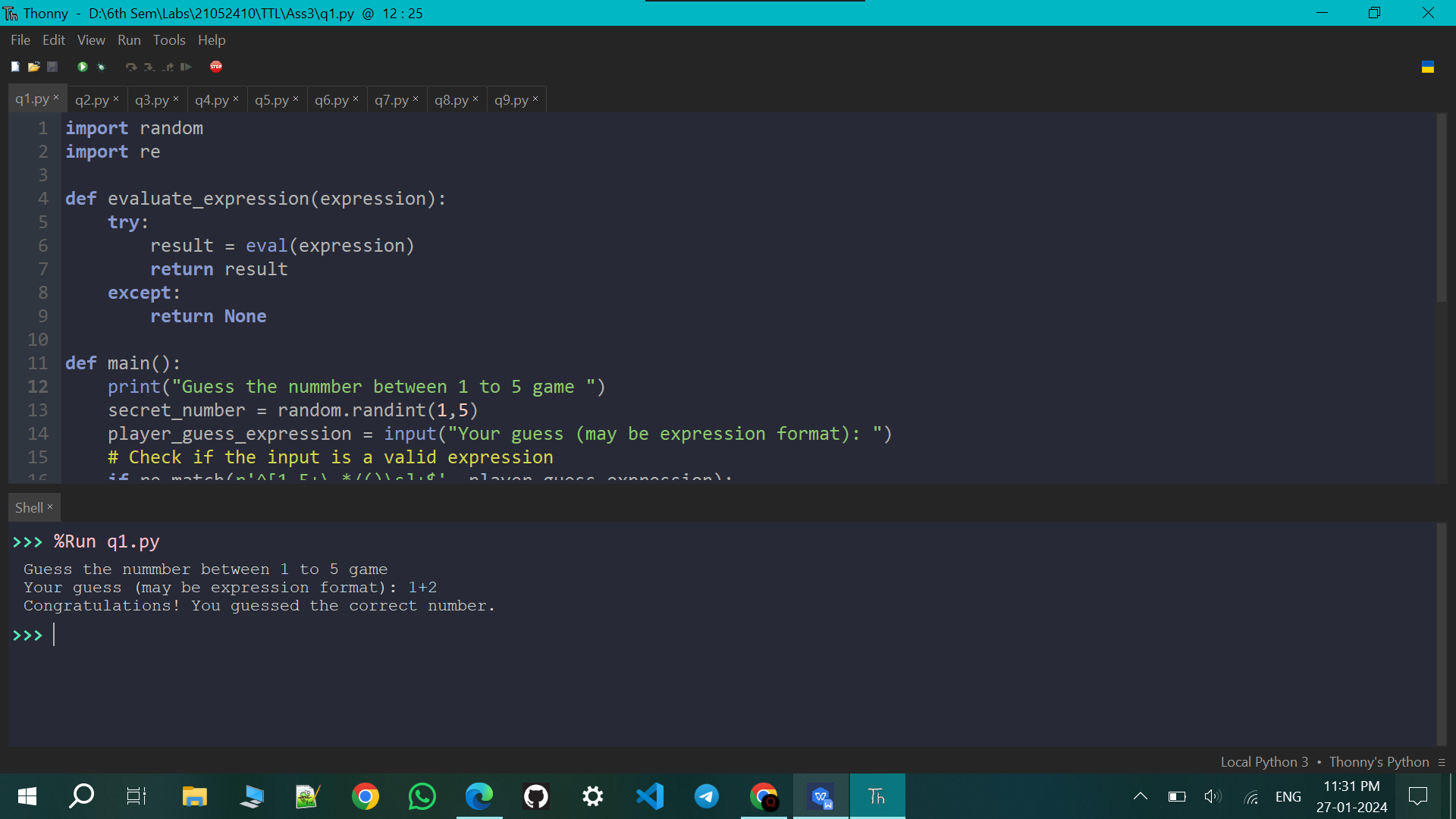
else:

print("Invalid input. Please enter a valid expression using numbers 1 to 5 and basic operators (+, -, \*, /).")

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

main()





Q2. Kids Multiplication Table Write a multiplication game program for kids. The program should give the player ten randomly generated multiplication questions to do. After each, the program should tell them whether they got it right or wrong and what the correct answer is.

import random

def generate\_question():

num1 = random.randint(1, 10)

num2 = random.randint(1, 10)

return num1, num2

def ask\_question(num1, num2):

user\_answer = int(input(f" {num1} times {num2}? "))

return user\_answer == num1 \* num2

def main():

print("Welcome to the Multiplication Game!")

correct\_count = 0

total\_questions = 10

for \_ in range(total\_questions):

num1, num2 = generate\_question()

is\_correct = ask\_question(num1, num2)

if is\_correct:

print("Correct! Great job!\n")

correct\_count += 1

else:

print(f"That's incorrect, the correct answer is {num1 \* num2}.\n")

if correct\_count==10:

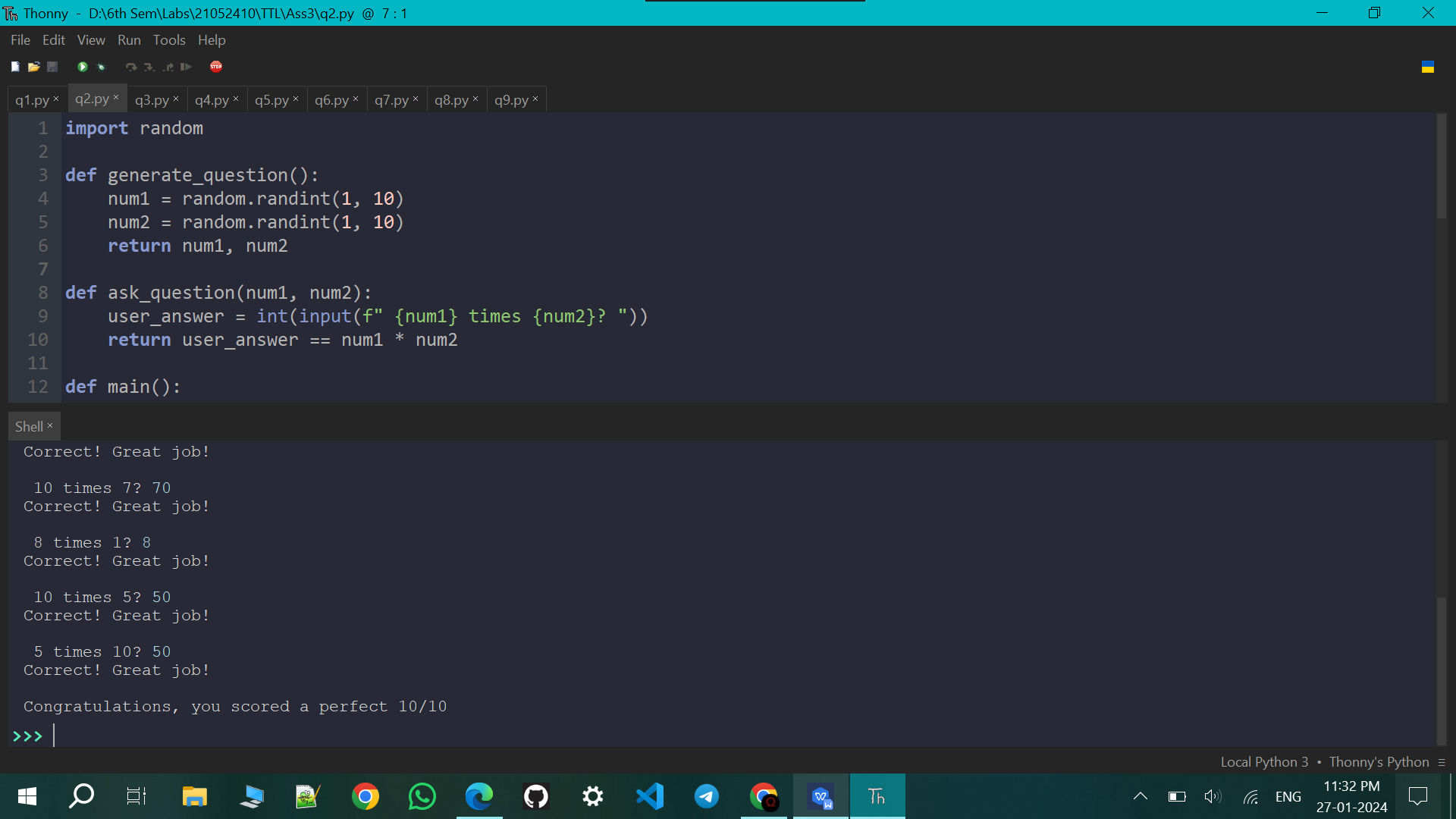
print('Congratulations, you scored a perfect 10/10')

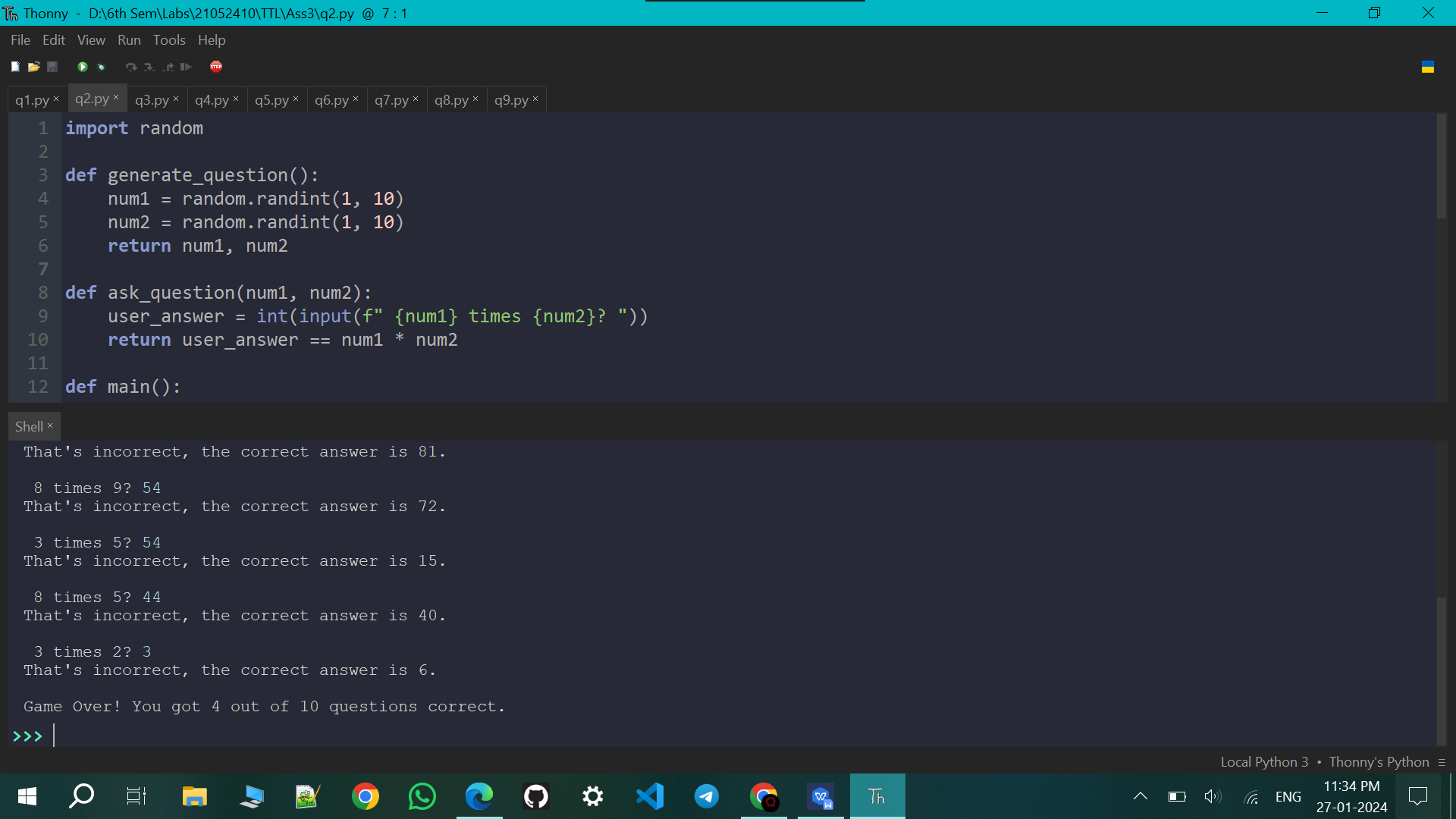
else:

print(f"Game Over! You got {correct\_count} out of {total\_questions} questions correct.")

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

main()





Q3. Check a list contains even number or not

numbers = [11,33,55,39,55,75,37,21,23,41,13]

def contains\_even(numbers):

for num in numbers:

if num % 2 == 0:

return True

return False

n=eval(input('Enter the total no in list: '))

my\_list=[]

for \_ in range(n):

x=int(input(f'{\_}: '))

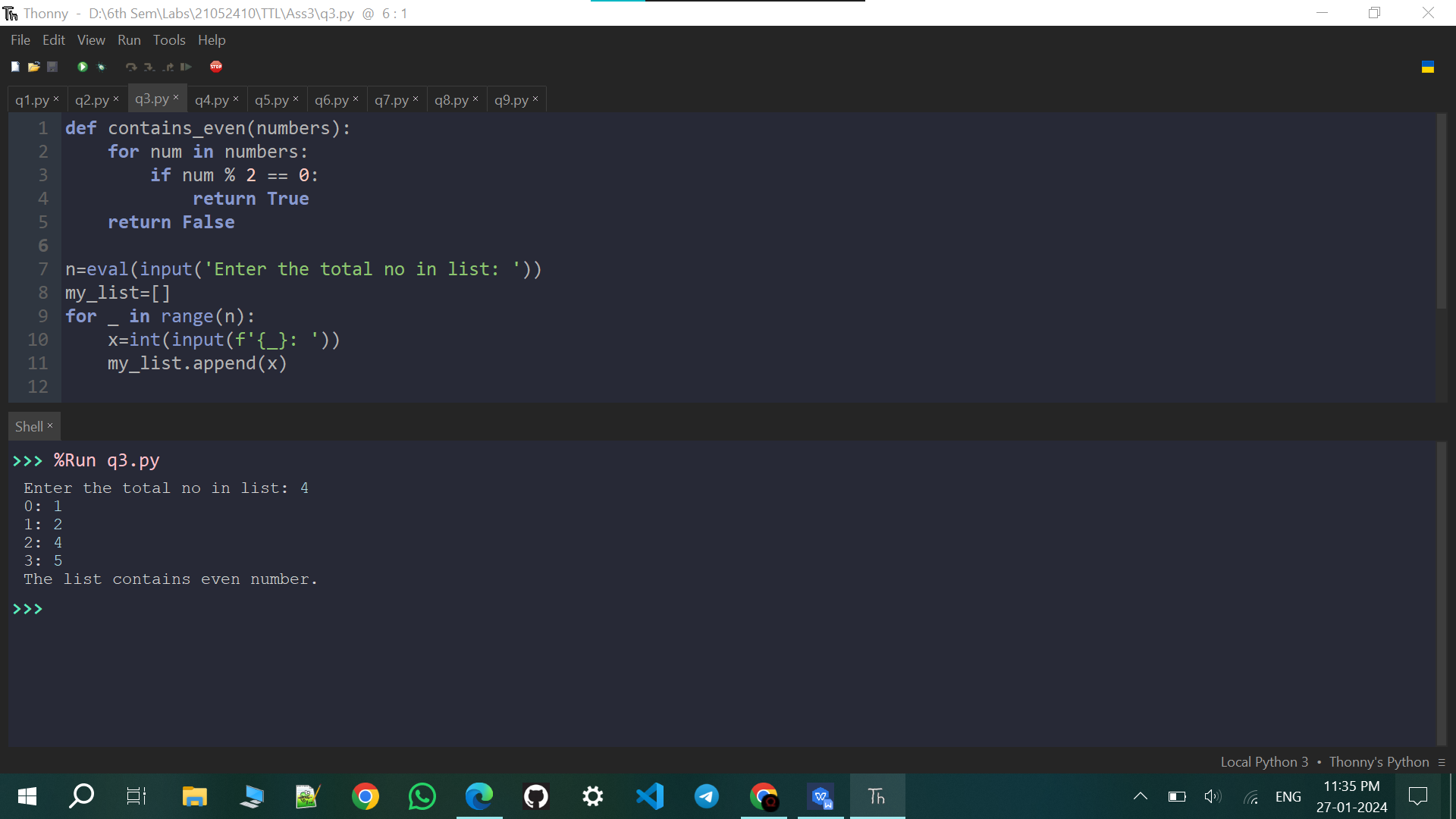
my\_list.append(x)

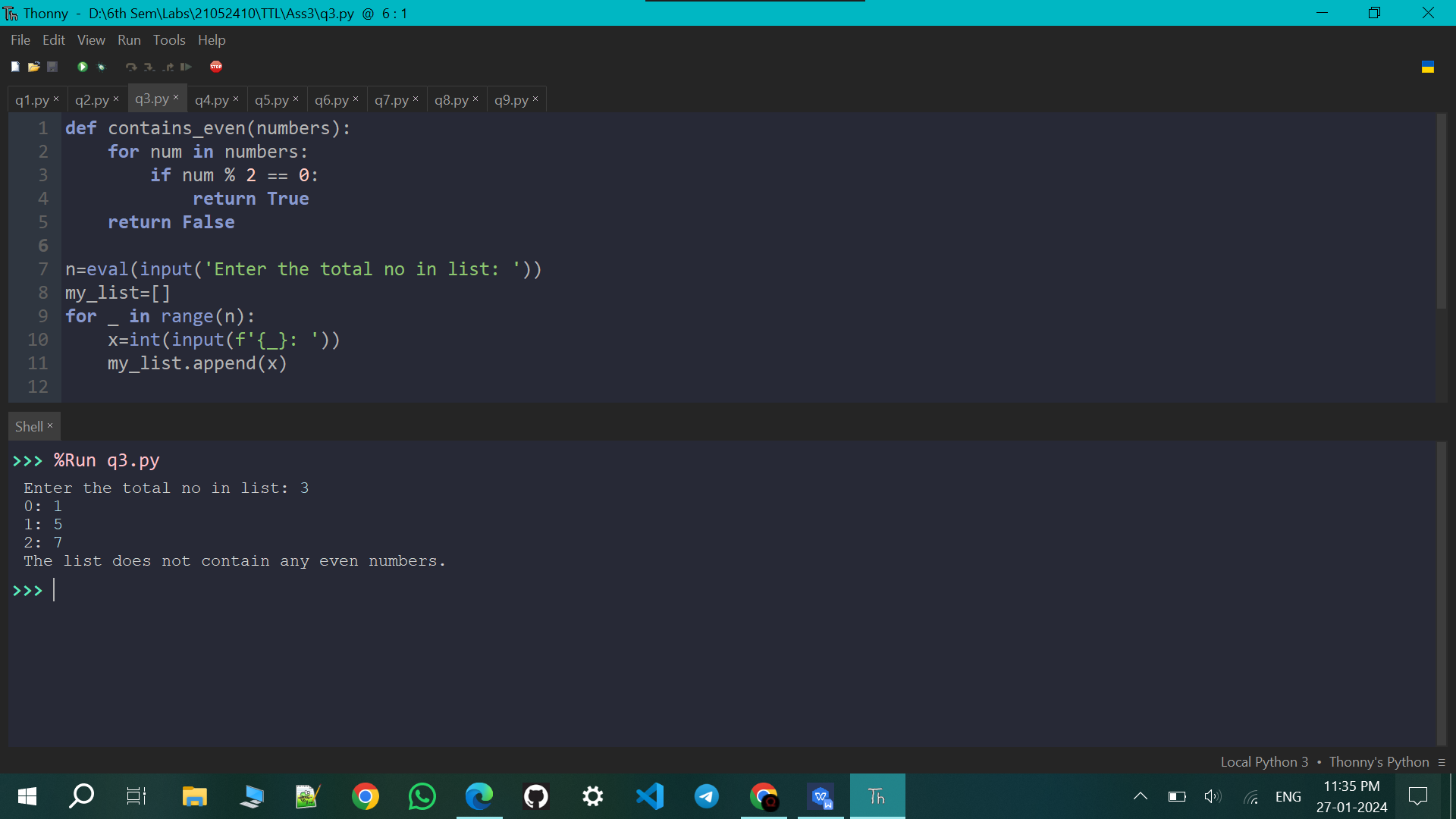
if contains\_even(my\_list):

print("The list contains even number.")

else:

print("The list does not contain any even numbers.")





Q4. Read N and generate the Fibonacci sequence upto N.

def generate\_fibonacci(n):

fibonacci\_sequence = [0, 1]

while len(fibonacci\_sequence) < n:

next\_number = fibonacci\_sequence[-1] + fibonacci\_sequence[-2]

fibonacci\_sequence.append(next\_number)

return fibonacci\_sequence[:n]

# Read N from the user

try:

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

if n <= 0:

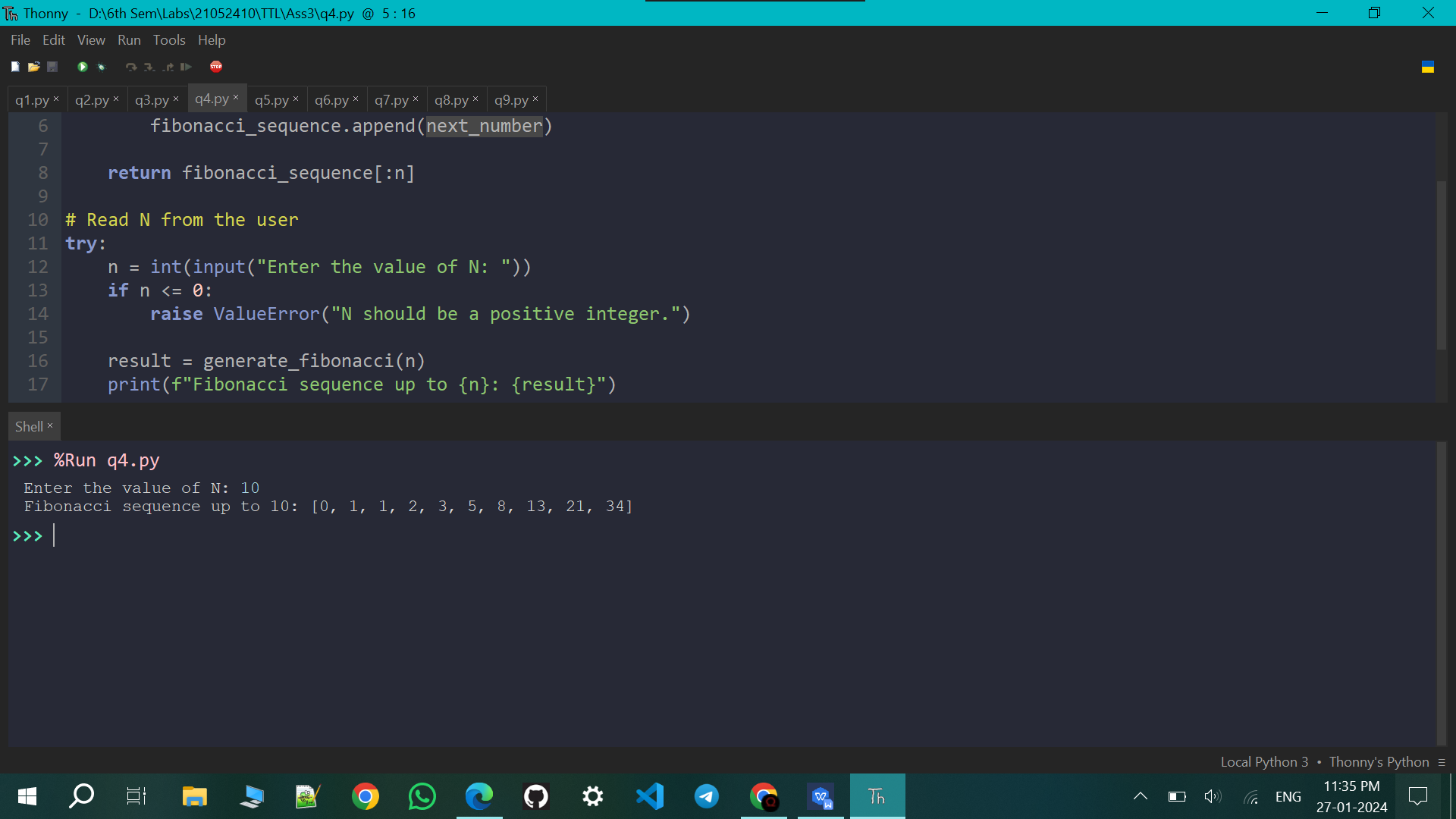
raise ValueError("N should be a positive integer.")

result = generate\_fibonacci(n)

print(f"Fibonacci sequence up to {n}: {result}")

except ValueError as ve:

print(f"Error: {ve}")



Q5. Use a for loop to print a box like the one below. Allow the user to specify how wide and how high the box should be. [Hint: print('\*'\*10) prints ten asterisks.]

print('Enter row and column to make a filled box ')

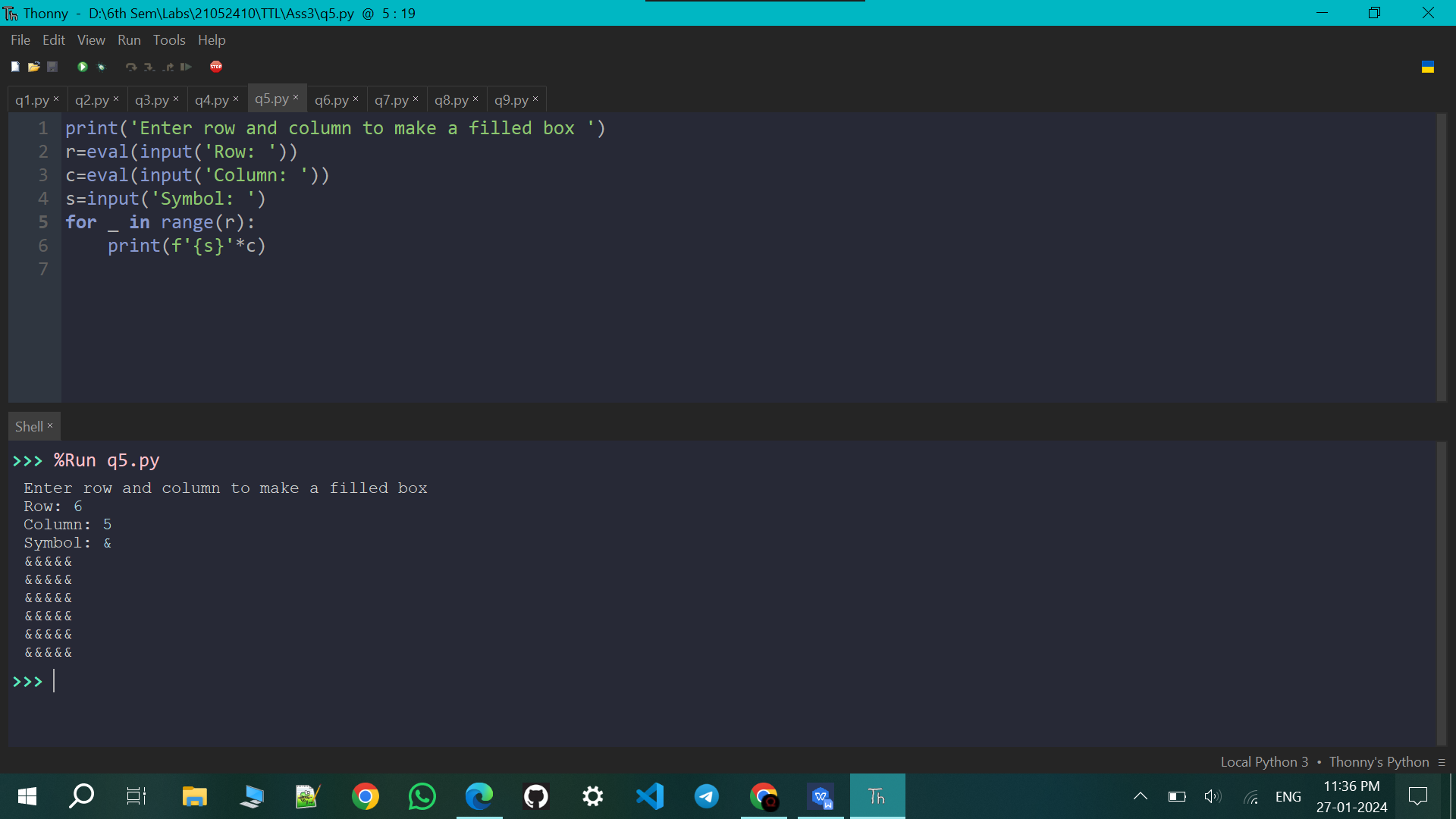
r=eval(input('Row: '))

c=eval(input('Column: '))

s=input('Symbol: ')

for \_ in range(r):

print(f'{s}'\*c)



Q6. Use loop to print a box like the one below. Allow the user to specify how wide and how high the box should be.

print('Enter row and column to make a hollow box ')

r=eval(input('Row: '))

c=eval(input('Column: '))

s=input('Symbol: ')

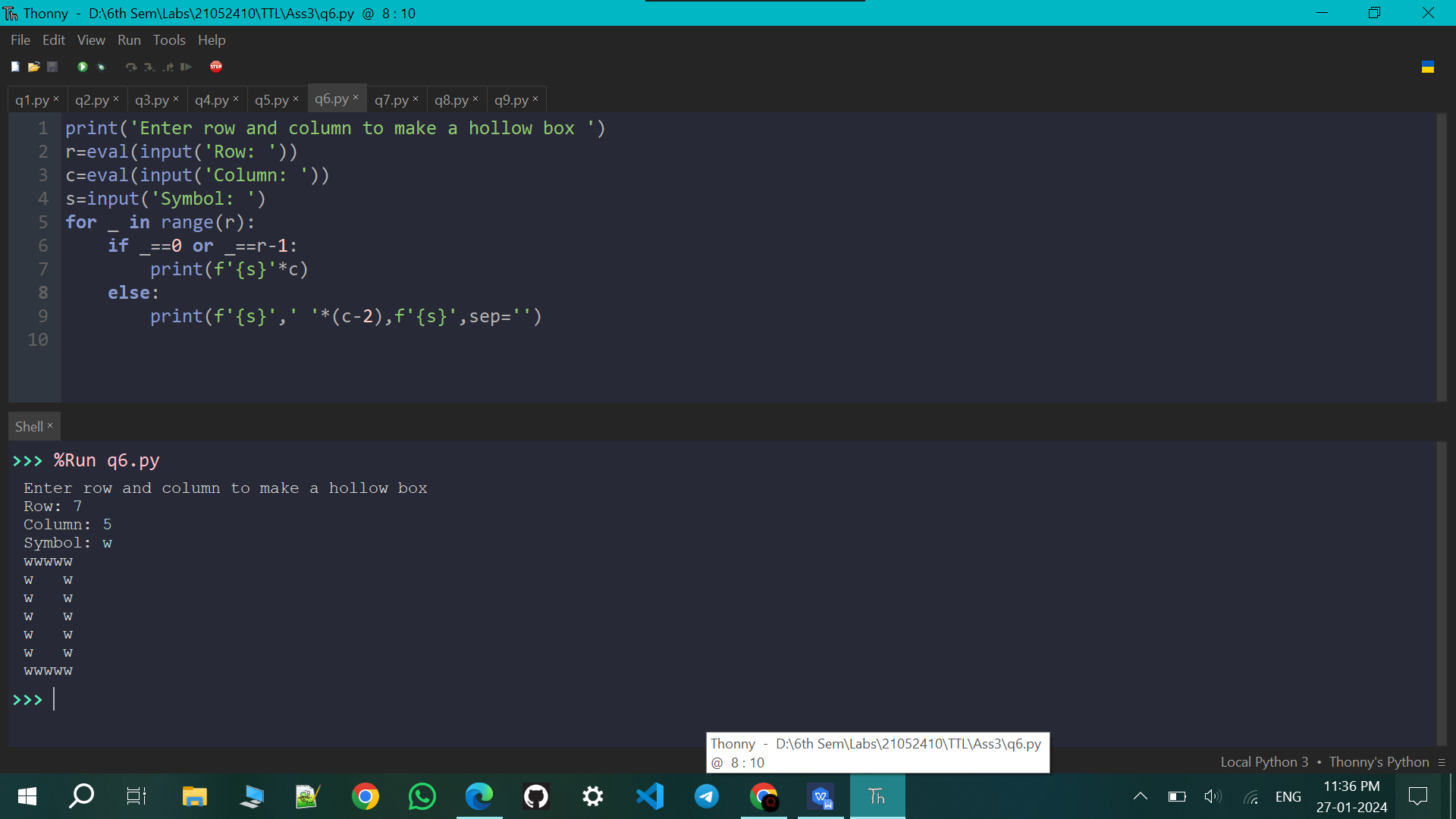
for \_ in range(r):

if \_==0 or \_==r-1:

print(f'{s}'\*c)

else:

print(f'{s}',' '\*(c-2),f'{s}',sep='')



Q7. Use for loops to print a diamond like the one below. Allow the user to specify how high the diamond should be.

def print\_diamond(height,s):

# Upper half of the diamond

for i in range(1, height // 2 + 2):

spaces = " " \* (height // 2 + 1 - i)

stars = s \* (2 \* i - 1)

print(spaces + stars)

# Lower half of the diamond

for i in range(height // 2, 0, -1):

spaces = " " \* (height // 2 + 1 - i)

stars = s \* (2 \* i - 1)

print(spaces + stars)

# Get user input for the height of the diamond

print('Enter the height and symbol to make a filled diamond')

user\_height = int(input("Height (must be an odd number): "))

s=input('Symbol: ')

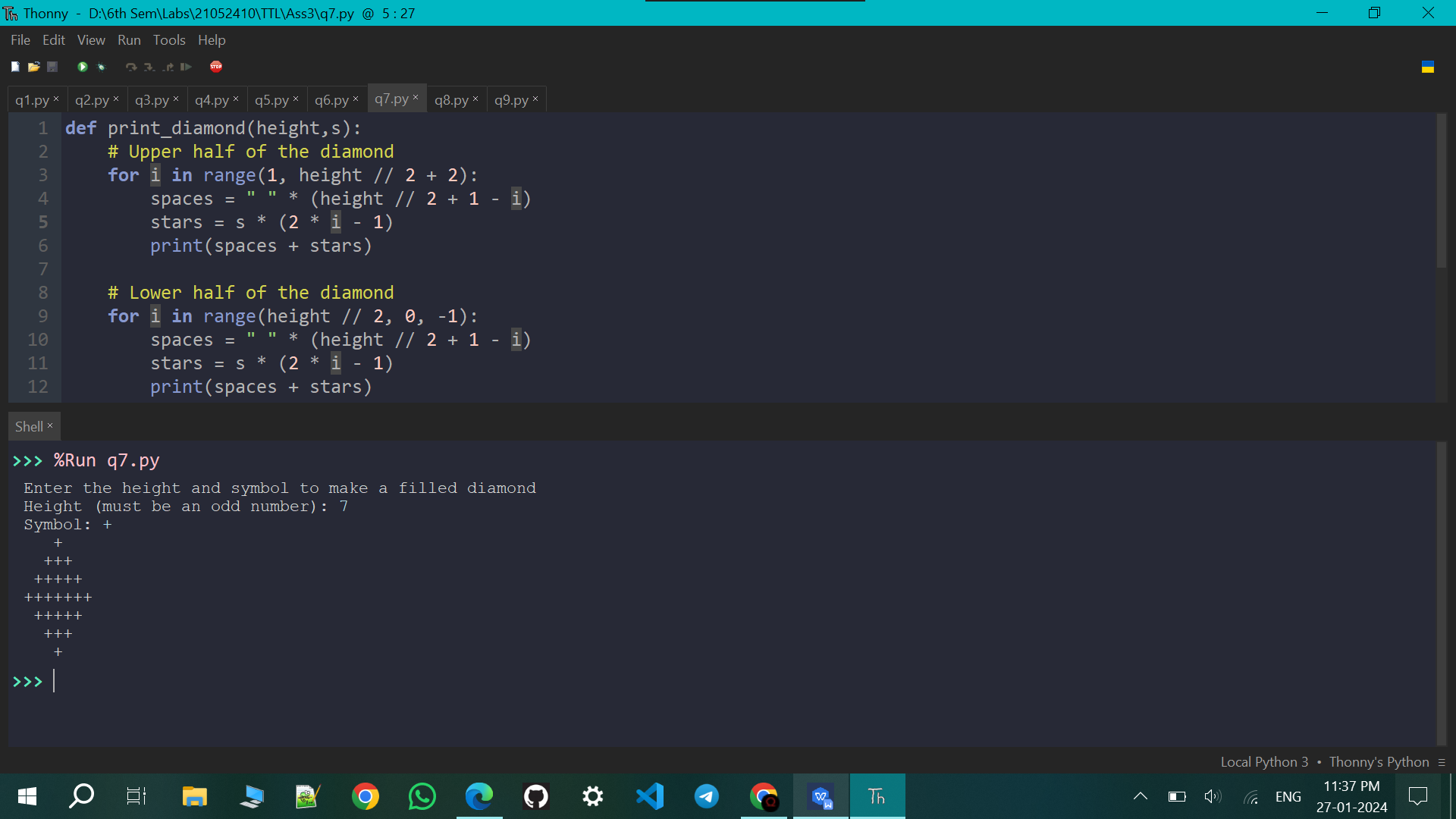
# Check if the entered height is odd

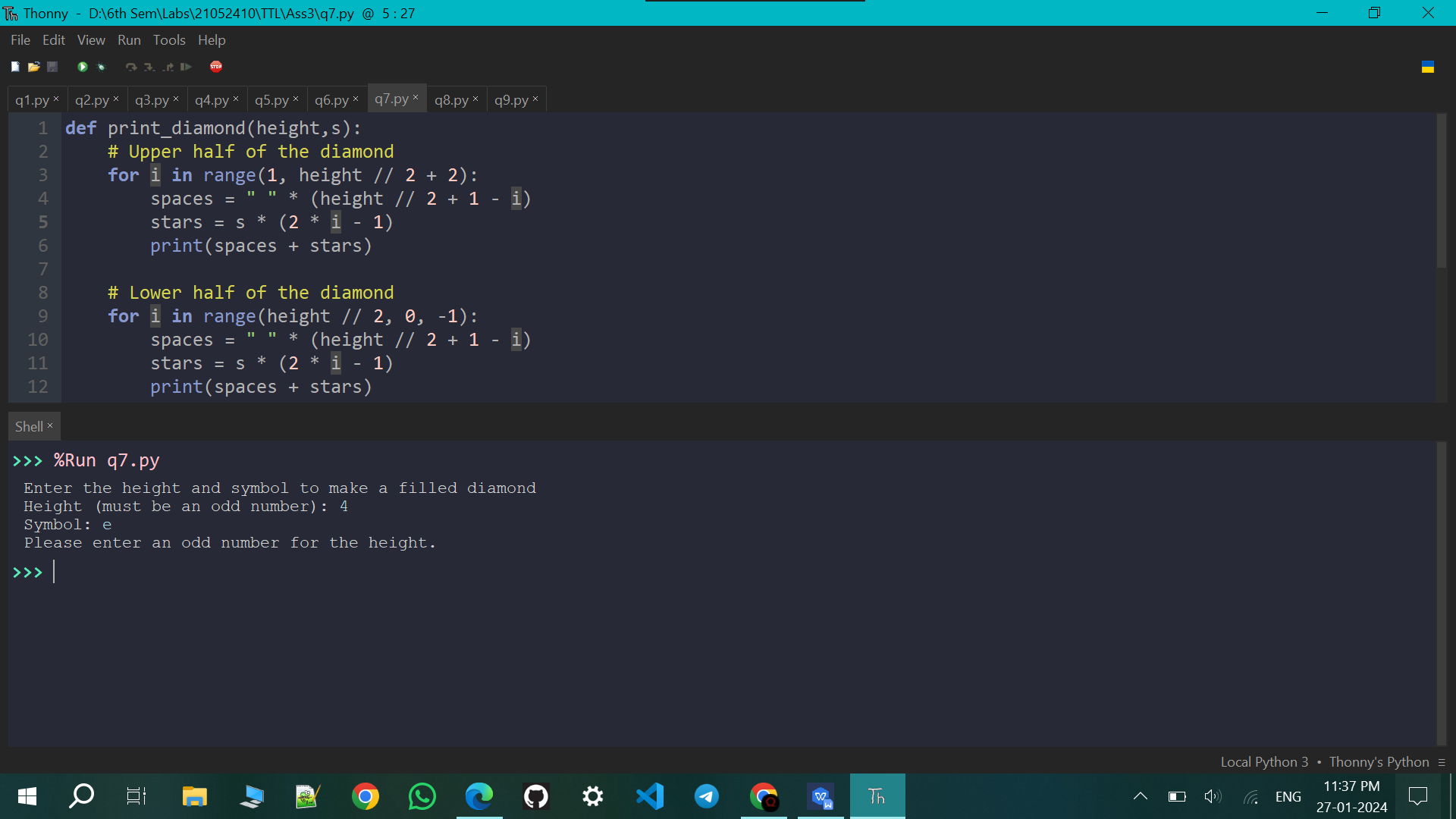
if user\_height % 2 != 0:

print\_diamond(user\_height,s)

else:

print("Please enter an odd number for the height.")





Q8. Write a program that lets the user play Rock-Paper-Scissors against the computer. There should be five rounds, and after those five rounds, your program should print out who won and lost or that there is a tie.

import random

def get\_user\_choice():

user\_choice = input("Enter your choice (rock, paper, or scissors): ").lower()

while user\_choice not in ["rock", "paper", "scissors"]:

print("Invalid choice. Please enter rock, paper, or scissors.")

user\_choice = input("Enter your choice: ").lower()

return user\_choice

def get\_computer\_choice():

return random.choice(["rock", "paper", "scissors"])

def determine\_winner(user\_choice, computer\_choice):

if user\_choice == computer\_choice:

return "Tie"

elif (user\_choice == "rock" and computer\_choice == "scissors") or \

(user\_choice == "paper" and computer\_choice == "rock") or \

(user\_choice == "scissors" and computer\_choice == "paper"):

return "You win!"

else:

return "Computer wins!"

def play\_game():

user\_wins = 0

computer\_wins = 0

for round in range(1, 6):

print(f"\nRound {round}:")

user\_choice = get\_user\_choice()

computer\_choice = get\_computer\_choice()

print(f"You chose {user\_choice}.")

print(f"Computer chose {computer\_choice}.")

result = determine\_winner(user\_choice, computer\_choice)

print(result)

if "you" in result.lower():

user\_wins += 1

elif "computer" in result.lower():

computer\_wins += 1

print(f"You won {user\_wins} rounds.")

print(f"Computer won {computer\_wins} rounds.")

if user\_wins > computer\_wins:

print("Congratulations! You are the overall winner!")

elif user\_wins < computer\_wins:

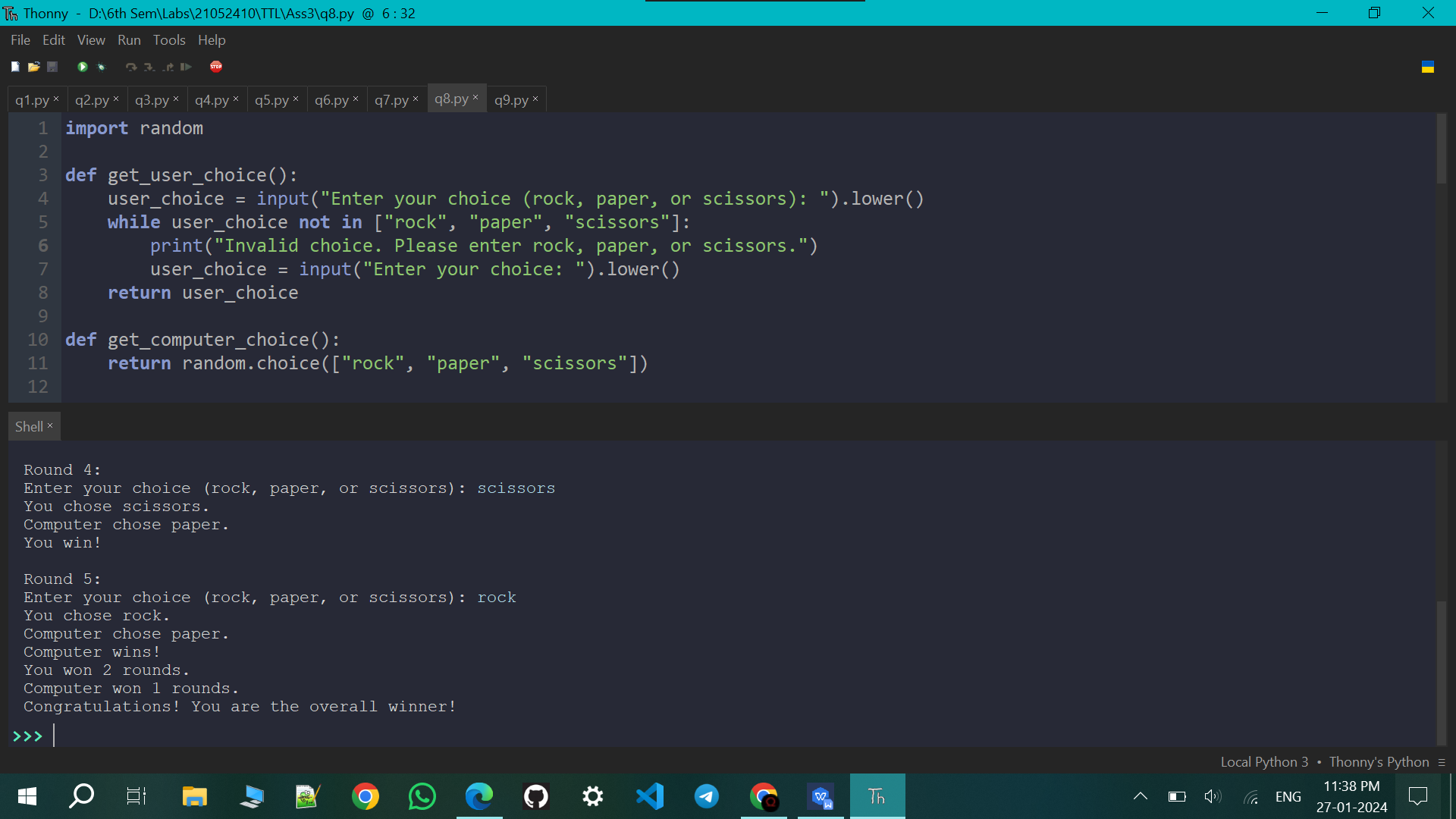
print("Game Over!, the computer is the overall winner!")

else:

print("It's a tie!")

# Start the game

play\_game()



Q9. Playing with Magic Words Here a word ‘S’ of length ‘n’ is said to be magic word if it satisfies the following conditions: All letters of S are lowercase letters of the English alphabets. Si, the character in the ith position, is lexicographically smaller than Sn-1-i for all even i from 0 to n/2 Si is lexicographically greater than Sn-1-i for all odd i from 0 to n/2 For example, the word “difference” is a magic word, while “similar” is not. Given a word, write python code to check whether the word is magic or not.

