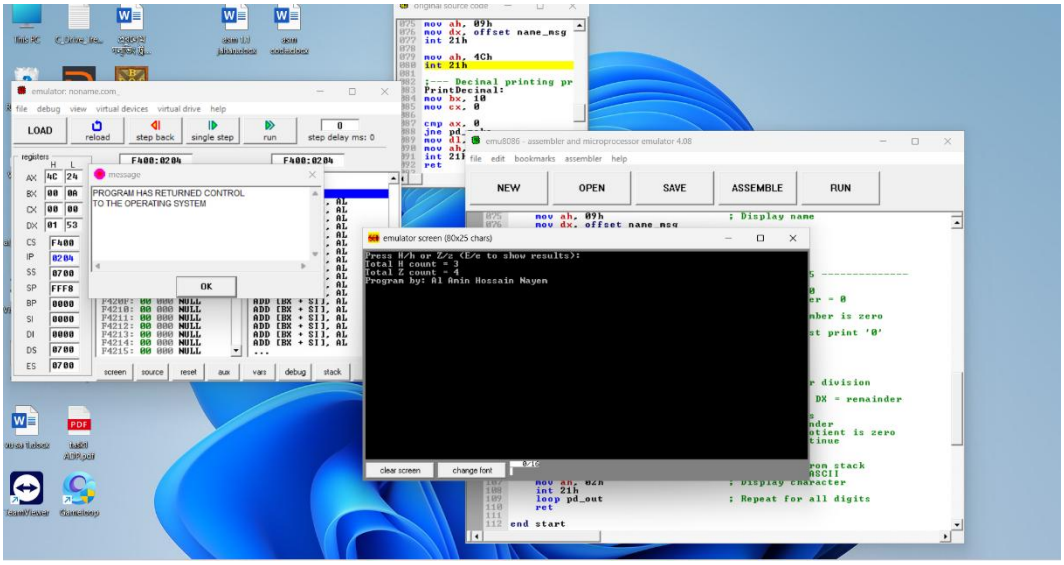
Output Explanation

In a test run the keys were pressed in the order: H H Z Z Z H E. The program displayed:
Total H count = 3
Total Z count = 4
Program by: Al Amin Hossain Nayem

This means three H/h presses and four Z/z presses were recorded before exiting. The screenshot below shows the program running inside EMU8086 and confirming these results.



Program Explanation

This 8086 assembly program keeps track of how many times you press the H/h and Z/z keys. It is built as a TINY COM executable so it fits in a single memory segment and runs directly under DOS.

• Setup: At startup the program prints a prompt and waits for key presses using DOS interrupt 21h function 7 so keys are read without automatic echo.
• Counting Logic: If you press H or h, the H counter increases. Pressing Z or z increases the Z counter. Any other key is ignored. Pressing E or e stops the loop and shows the final totals.
• Displaying Results: After E/e is pressed, the program prints totals using the PrintDecimal procedure, which converts the numbers to ASCII by dividing by 10 and printing each digit.
• Environment: Assembler model – TINY (COM file). Emulator/Interpreter – EMU8086.

Code Explanation

1. Initialization

asm

```
start:
  mov ax, @data          ; Initialize data segment
  mov ds, ax             ; Set DS to point to data segment

  mov ah, 09h            ; Display prompt
  mov dx, offset prompt
  int 21h
```

- Sets up the data segment so the program can access variables
- Displays the initial prompt to guide the user

2. Main Loop - Key Reading

asm

```
main_loop:
  mov ah, 7              ; Read key without echo
  int 21h
  mov bl, al             ; Store character in BL
```

- Uses DOS interrupt 21h function 07h to read keyboard input silently
- Stores the character in BL register for comparison

3. Conditional Checks

asm

```
  cmp bl, 'E'            ; Check for exit condition (E)
  je show_results
  cmp bl, 'e'            ; Check for exit condition (e)
  je show_results
```

```
cmp bl, 'H'              ; Check for H (uppercase)
je inc_H
cmp bl, 'h'              ; Check for h (lowercase)
je inc_H

cmp bl, 'Z'              ; Check for Z (uppercase)
je inc_Z
cmp bl, 'z'              ; Check for z (lowercase)
je inc_Z
```

- First checks for exit conditions (E/e)
- Then checks for H/h and Z/z keys
- Uses conditional jumps to handle each case

4. Counter Increment Routines

asm

```
inc_H:
  inc Hcount             ; Increment H counter
  jmp main_loop          ; Continue loop

inc_Z:
  inc Zcount             ; Increment Z counter
  jmp main_loop          ; Continue loop
```

- Simple routines that increment the appropriate counter
- Immediately return to the main loop

5. Results Display

asm

```
show_results:
  mov ah, 09h              ; Display H count message
  mov dx, offset finalH
  int 21h

  xor ax, ax               ; Clear AX
  mov al, Hcount           ; Load H count into AL
  call PrintDecimal        ; Display H count as decimal
  ; ... similar for Z count ...
```

- Displays formatted results with proper decimal conversion
- Shows both H and Z counts separately

6. Decimal Conversion Algorithm

asm

```
PrintDecimal:
  ; Converts binary number to ASCII decimal
  ; Handles numbers 0-255 correctly
  ; Uses stack to reverse digit order
```

Conclusion

This project successfully demonstrates a simple but complete 8086 assembly application that interacts with the keyboard, tracks specific key presses, and displays results in a user-friendly way.
By using the **TINY COM model** and running in the **EMU8086 emulator**, the program efficiently counts presses of the **H/h** and **Z/z** keys and outputs accurate totals when the user finishes by pressing **E/e**.

The task highlights core assembly concepts such as:

- Handling keyboard input with DOS interrupts

- Using conditional jumps and loops for program flow
- Converting numeric data to human-readable decimal format

Overall, it shows how low-level programming can directly manage hardware input and produce clear, reliable results with minimal resources.