Name:SOUNG VICHKA

ID:e20240672

Group:I2-GIC2-B

TP11

Computer Architecture (Assembly Language smz32v50)

We are going to create a guessing game. Below is the process:

1. You start running the program
2. Program requests the user to input a number from 0 to 9
3. The program compares the input number with the number of you have set in your program:
    1. If the input number is smaller, display "It is bigger" in VDU and accept new number
    2. If the input number is bigger, display "It is smaller" in VDU and accept new number
    3. If the input number is equal, display "Correct!". The game ends here.
4. Optional: If the user tries 3 times but still incorrect, display "You lose" in VDU

Code:

```
;Correct
JMP Init_v1        ; Skip the text data and jump to the program start

DB "Correct!"      ; Store the message "Correct!"
DB "It is smaller" ; Store the message "It is smaller"
DB "It is bigger"  ; Store the message "It is bigger"
DB "You lose"      ; Store the message "You lose"
DB "6"             ; Store the secret number '6' location 2B

Init_v1:
  MOV AL, 0        ; AL = 0
  PUSH AL          ; Push 0 onto stack (flag)
  PUSH AL          ; Push 0 onto stack (guess counter)

Init:
  POP DL           ; Get current guess counter from stack
  PUSH DL          ; Put it back on stack
  CMP DL,3         ; Compare guess counter with 3
  JZ Lose          ; If counter = 3, jump to Lose
  IN 00            ; Read a key from keyboard into AL
  CMP AL,[2B]      ; Compare input with secret number
  JZ Correct       ; If equal, jump to Correct
  CMP AL,[2B]      ; Compare input with secret number again
  JS Bigger        ; If input is smaller than secret number,
                   ; jump to Bigger
  JMP Smaller      ; Otherwise jump to Smaller
```

```
End_loop:
  POP DL          ; Pop guess counter
  MOV [F0], DL    ; Save it temporarily at F0
  POP DL          ; Pop flag value
  MOV DL, 1       ; DL = 1
  PUSH DL         ; Push flag = 1
  MOV DL,[F0]     ; Restore guess counter
  PUSH DL         ; Push guess counter back
  JMP AllinOne    ; Display the selected message

Reset:
  MOV BL, C0      ; BL points to start of screen memory

Reset_loop:
    MOV CL, [C]   ; Read blank character
    MOV [BL], CL  ; Write blank character to screen
    INC BL        ; Move to next screen location
    CMP BL, CD    ; Reached end of screen?
    JNZ Reset_loop ; No, continue clearing
    JMP Init      ; Restart game

Lose:
  MOV CL, 23      ; CL points to start of "You lose"
  MOV BL, C0      ; Screen output starts at C0
  MOV AL, 2B      ; End address of string
  JMP End_loop    ; Display message

Bigger:
  MOV CL, 17      ; CL points to start of "It is bigger"
  MOV BL, C0      ; Output position
  MOV AL, 23      ; End address of string
  JMP ALLinOne    ; Display message

Smaller:
  MOV CL, A       ; CL points to start of "It is smaller"
  MOV BL, C0      ; Output position
  MOV AL, 17      ; End address of string
  JMP ALLinOne    ; Display message

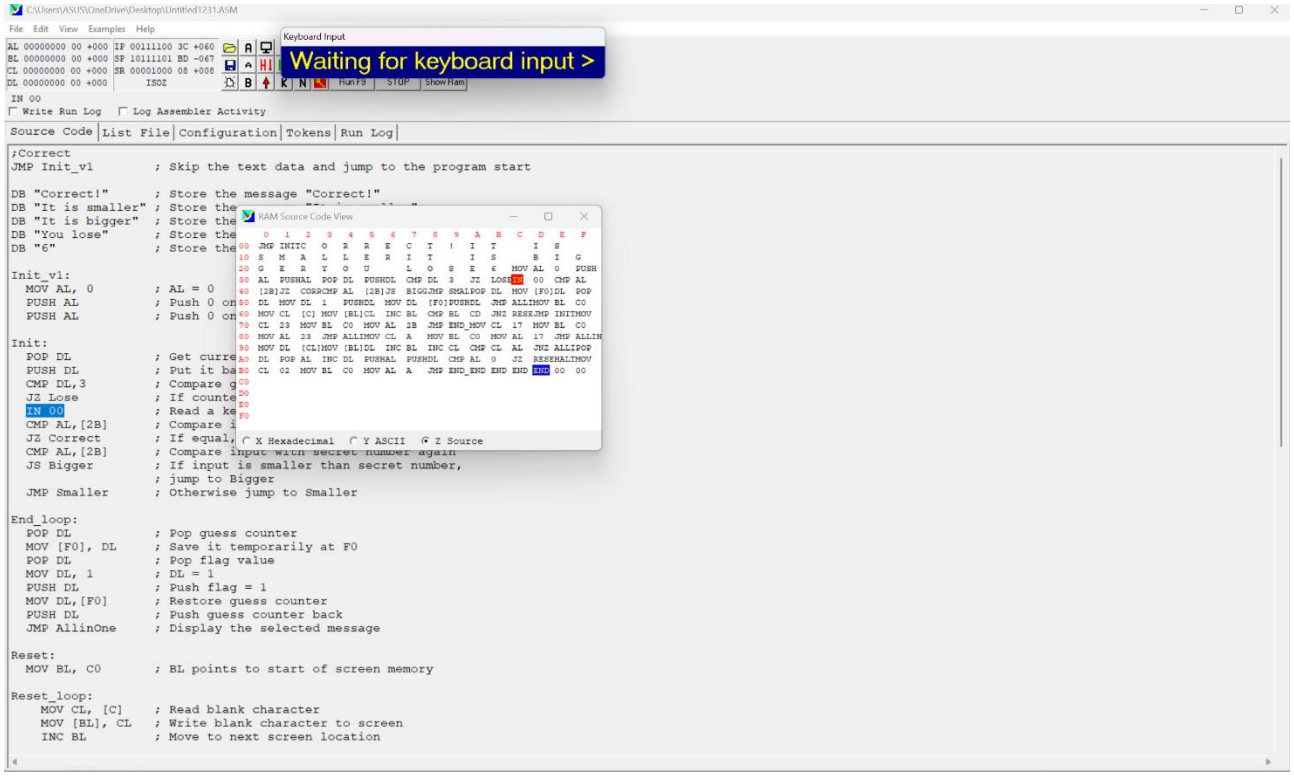
AllinOne:
  MOV DL,[CL]     ; Read one character from message
  MOV [BL], DL    ; Write character to screen
  INC BL          ; Move to next screen position
  INC CL          ; Move to next character in message
  CMP CL, AL      ; Reached end of message?
  JNZ AllinOne    ; No, continue copying characters
  POP DL          ; Pop guess counter
  POP AL          ; Pop flag
  INC DL          ; Increase guess counter by 1
  PUSH AL         ; Restore flag
  PUSH DL         ; Restore updated guess counter
```

```
  CMP AL, 0       ; Check if flag is 0
  JZ Reset        ; If 0, clear screen and restart
  HALT            ; Otherwise stop program

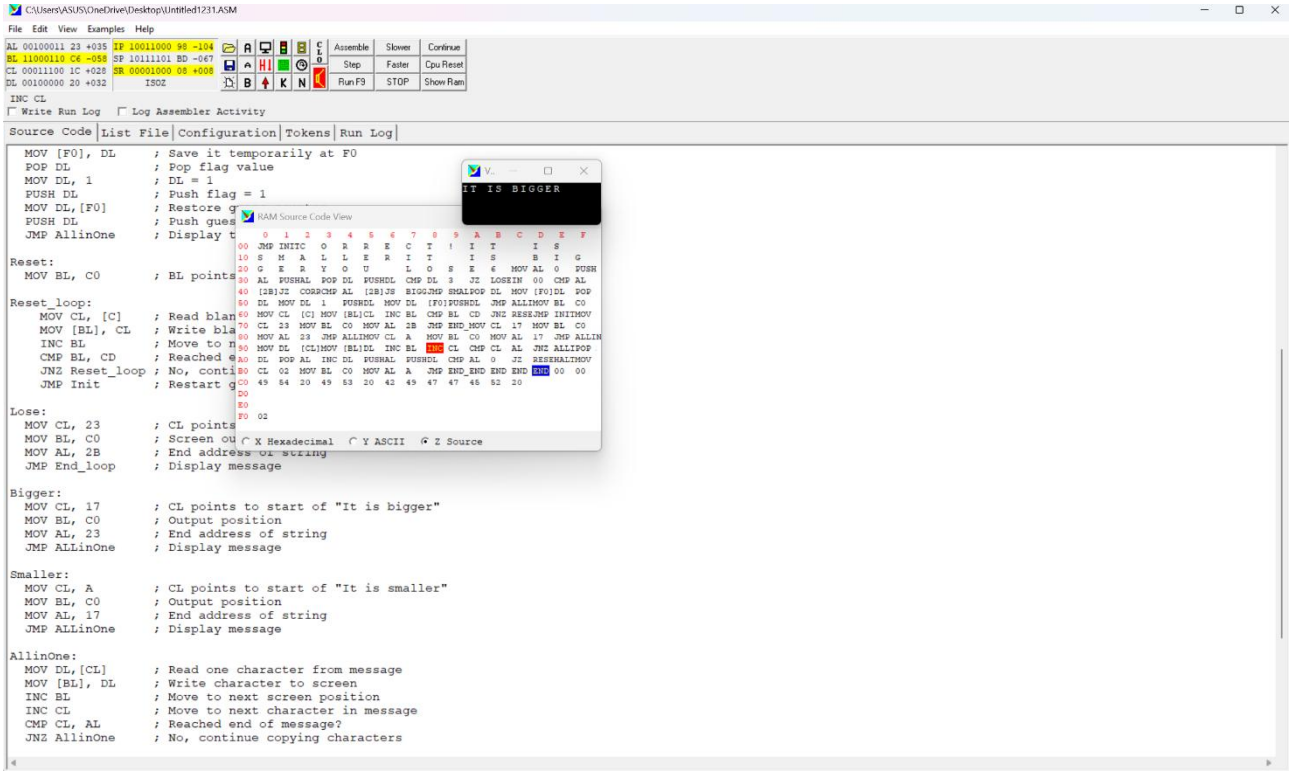
Correct:
  MOV CL, 02      ; CL points to start of "Correct!"
  MOV BL, C0      ; Output starts at screen address C0
  MOV AL, A       ; End address of string
  JMP End_loop    ; Display message

END
```

When we run the code:

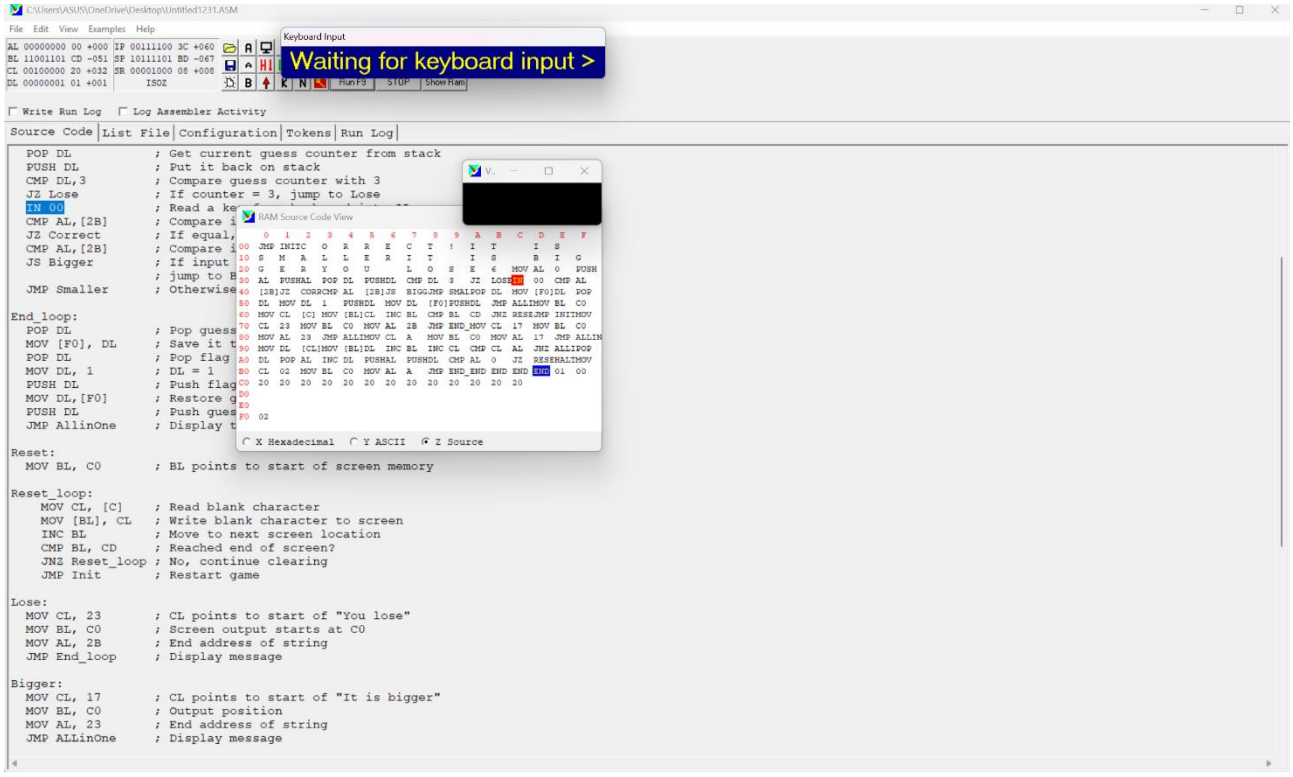


First Keyboard input was 2

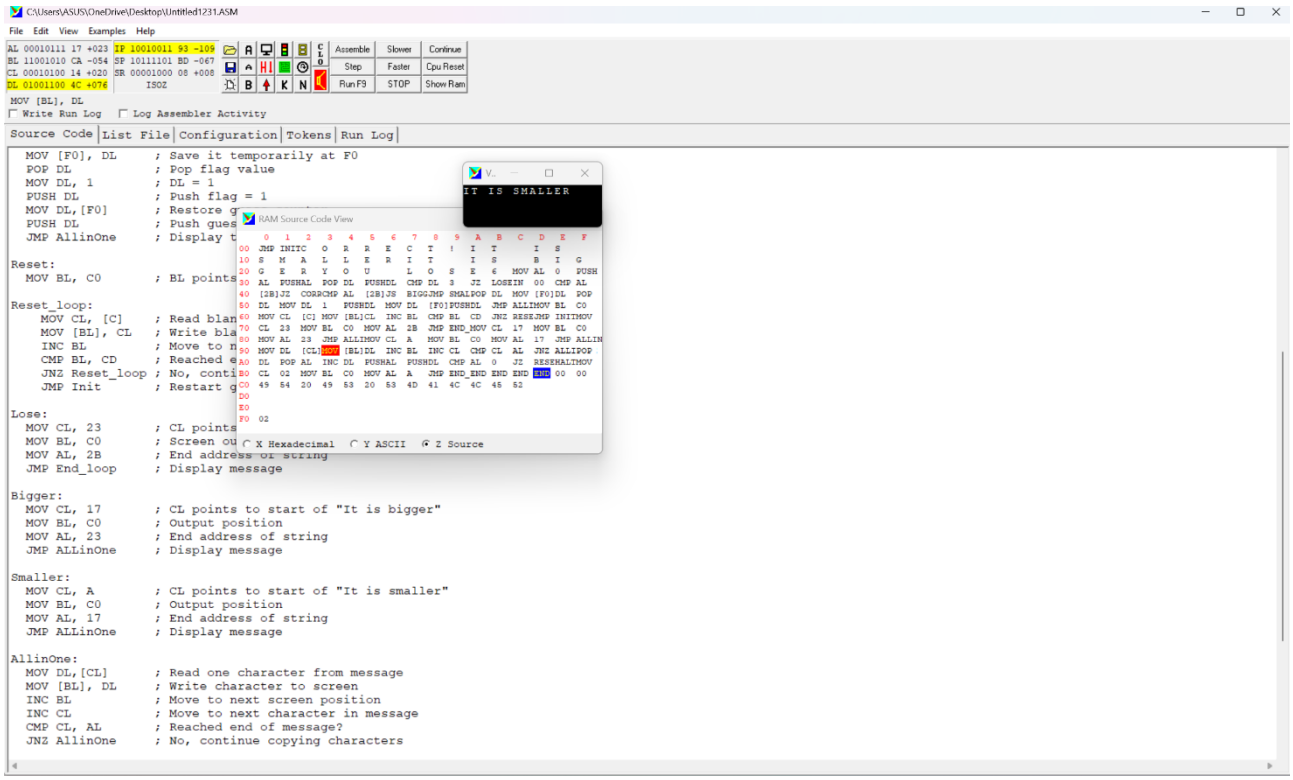


IT IS BIGGER

Suggest user to input again

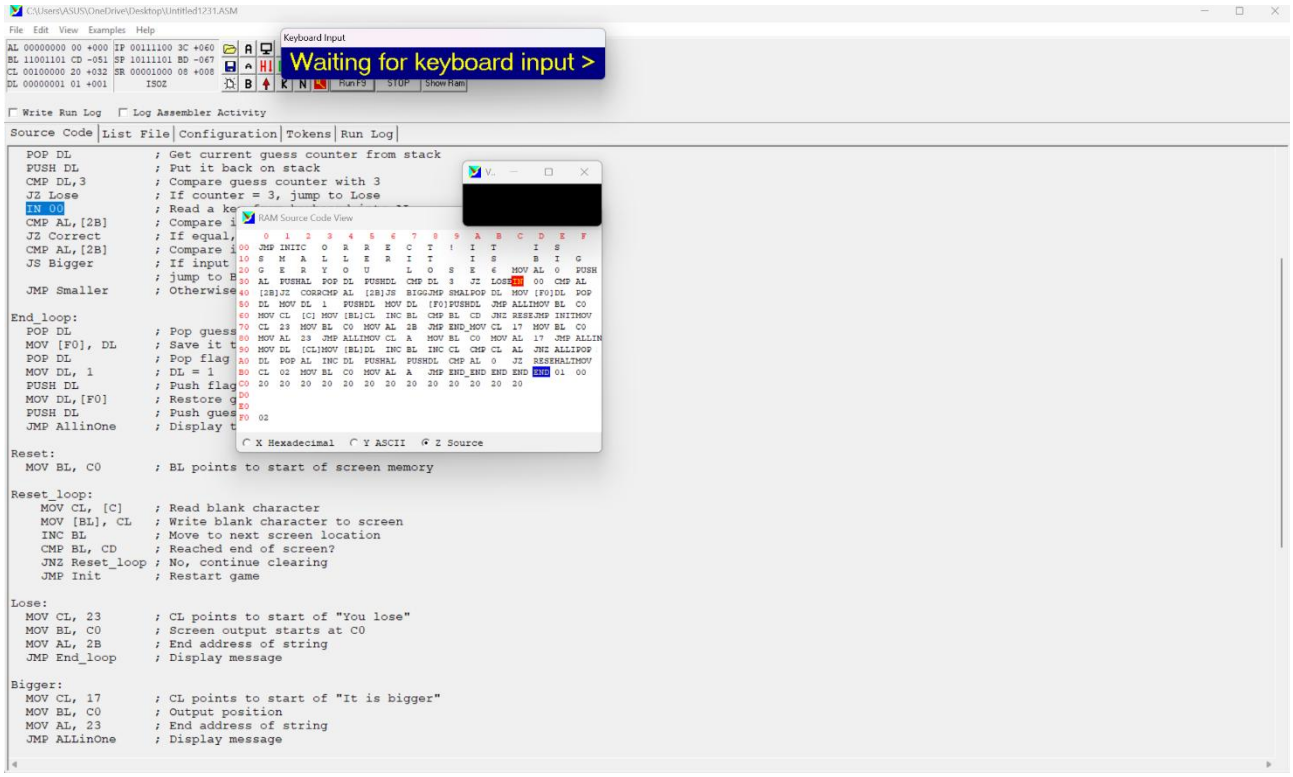


Second Keyboard input was 8

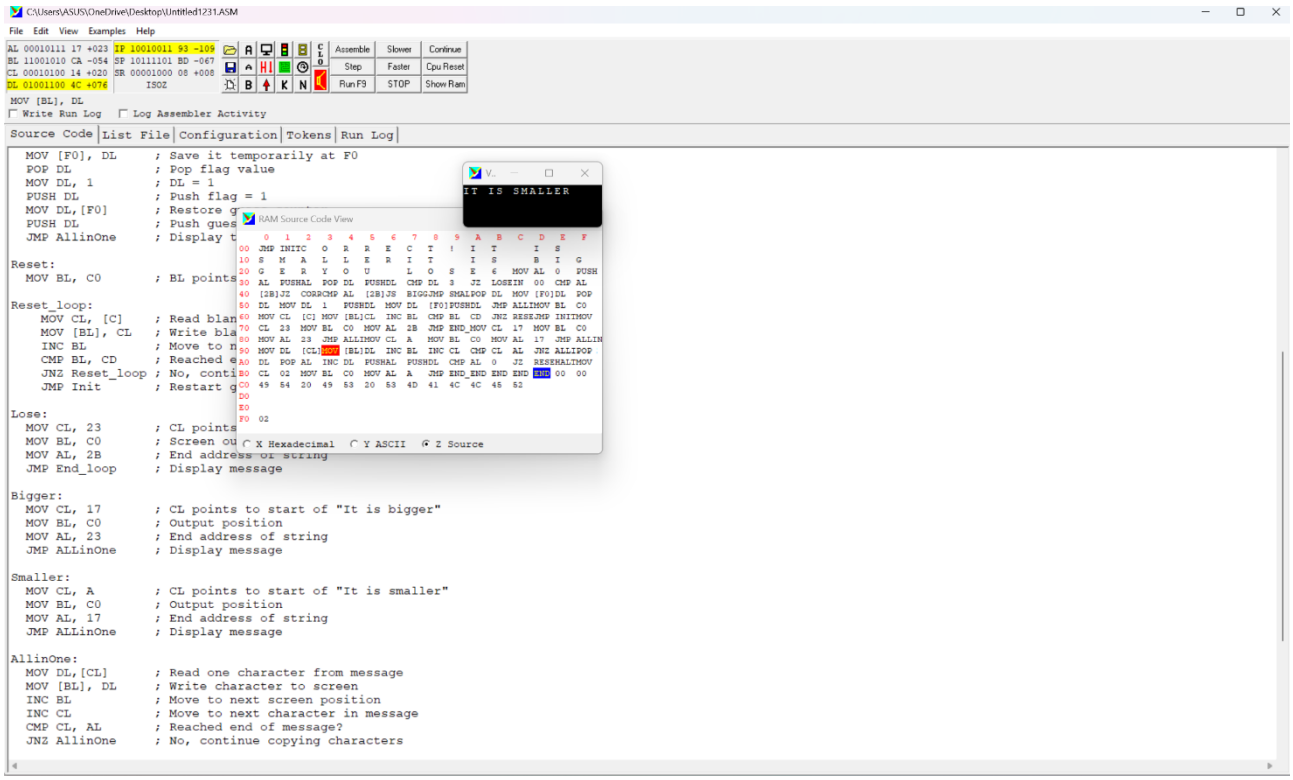


IT IS SMALLER

Suggest user to input again

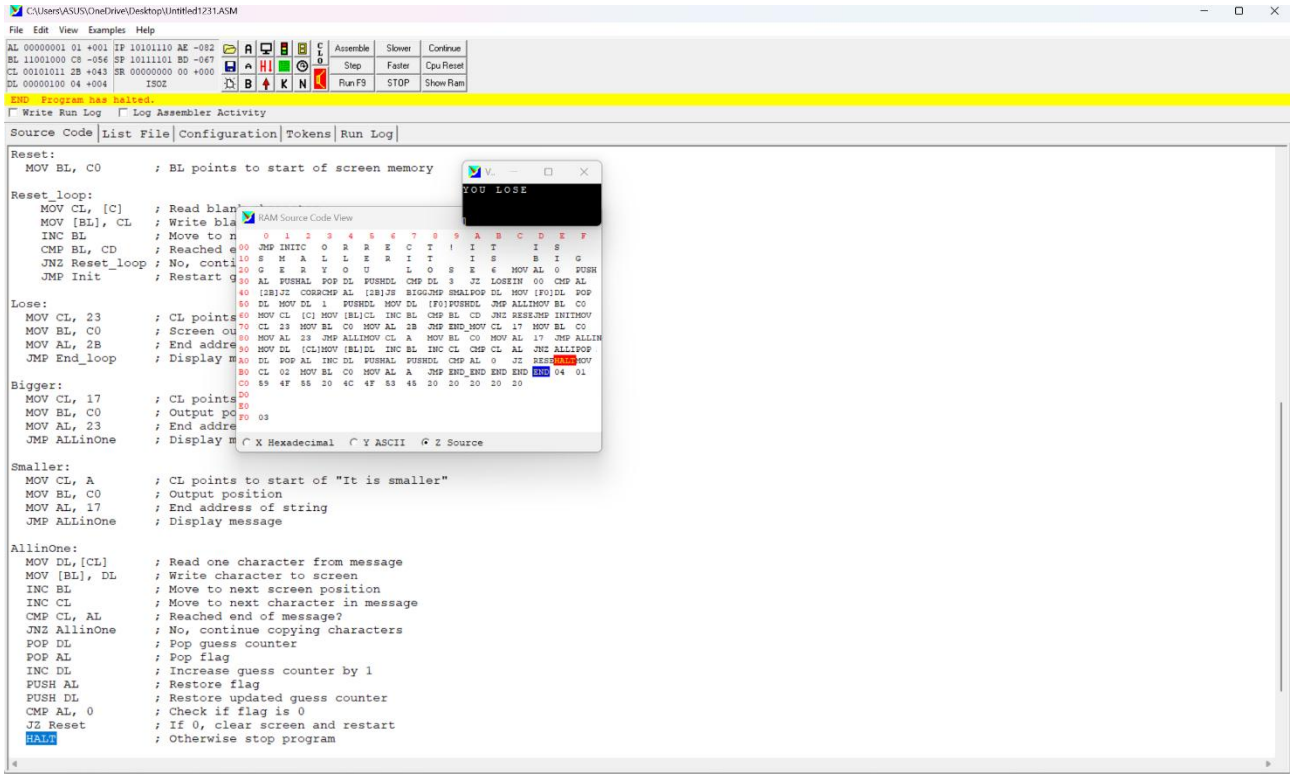


Third Keyboard input was 7

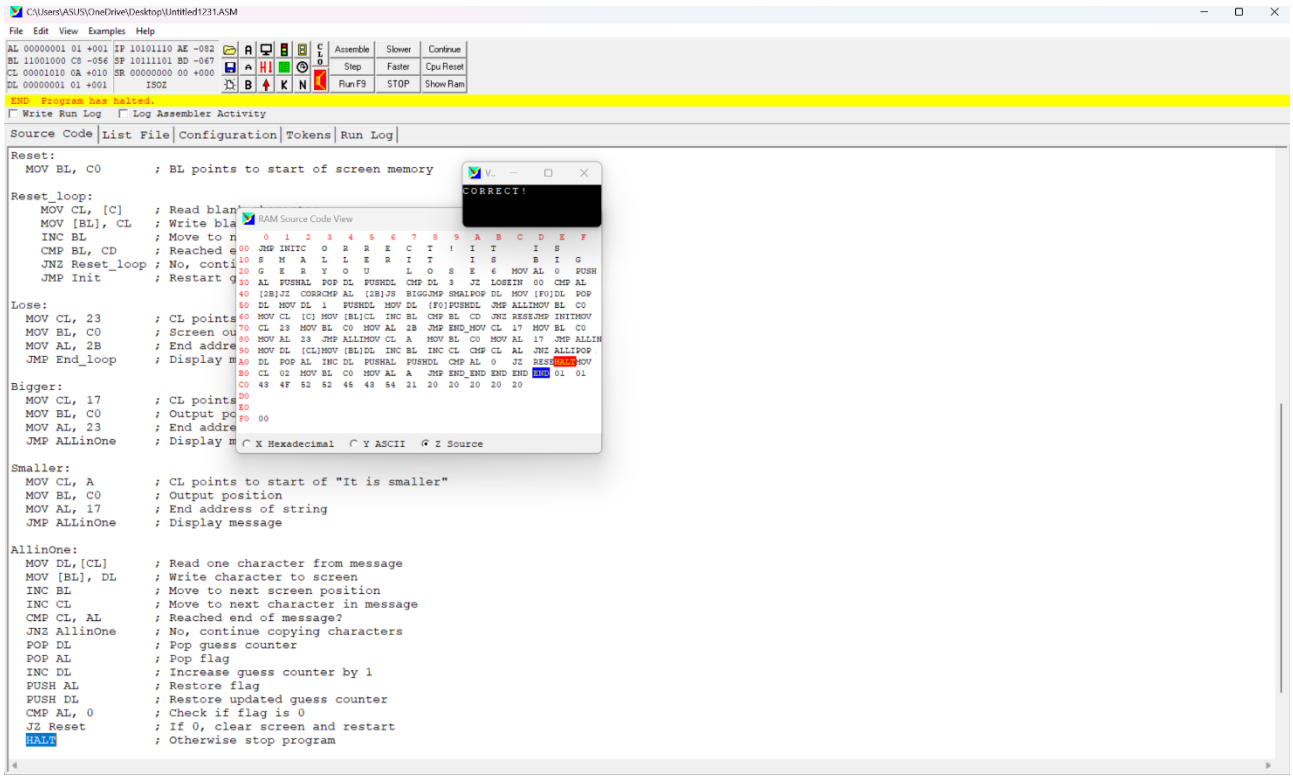


IT IS SMALLER

Third input and you still wrong program was ended



When you input right number



CORRECT! And then end program