

University of Information Technology & Sciences

Department of Computer Science and Engineering



Lab Report-03

Course Title: Microprocessors and Microcontrollers Lab
Course Code: CSE-360

Submitted To

Md. Ismail
Lecturer
Of
CSE Department

Submitted By

Name : Mobarok Hossain Zobaer
Id : 0432220005101029
Batch : 52
Semester: Spring 25
Section : 6A1

Problem Description: Printing of digit 0 to 9 using loop at Assembly Code

Implementation:

```
.MODEL SMALL
.stack 100h
.code

main proc          ;main procedude

MOV CL,0
PRINT:

MOV AH,2
MOV BL,CL
ADD BL,48
MOV DL,BL
INT 21H

mov dl,10
int 21h

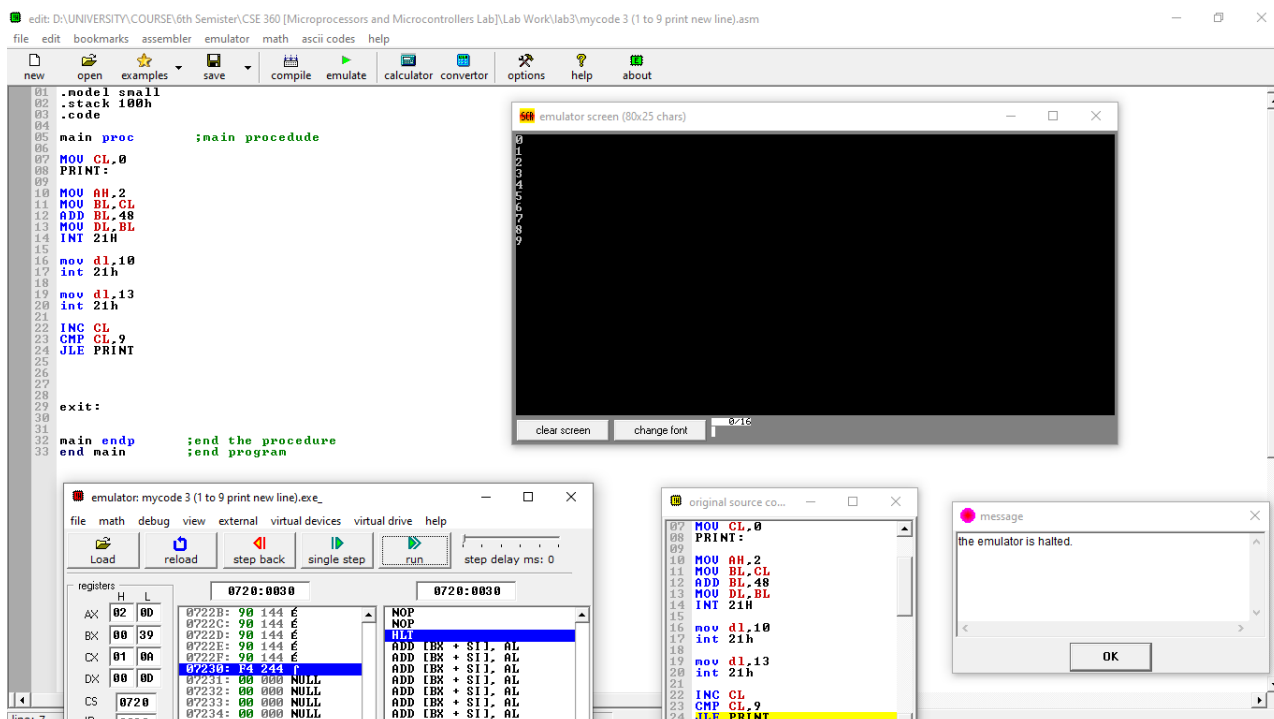
mov dl,13
int 21h

INC CL
CMP CL,9
JLE PRINT

exit:

main endp          ;end the procedure
end main            ;end program
```

Result:



Conclusion:

This assembly program prints the digits 0 to 9 on the screen, with each digit followed by a newline and a carriage return for proper formatting. Here's a summary of the code:

Digit Printing Loop: The program uses a loop, starting with CL = 0, to print digits from 0 to 9. Each digit is converted to its ASCII equivalent by adding 48 to the value of CL, and then printed using the INT 21h DOS interrupt (with function AH = 2).

Formatting: After printing each digit, the program outputs a newline (DL = 10) and a carriage return (DL = 13) to move the cursor to the next line and position it correctly.

Loop Control: The loop continues as long as CL is less than or equal to 9 (JLE), and it increments CL after each iteration.

The output displays the digits 0 through 9, each on a new line with a proper carriage return, ensuring a clean and formatted display. The program demonstrates basic input/output operations and control flow using loops in assembly language.

Problem Description: Printing of digit 9 to 0 using loop at Assembly Code

Implementation:

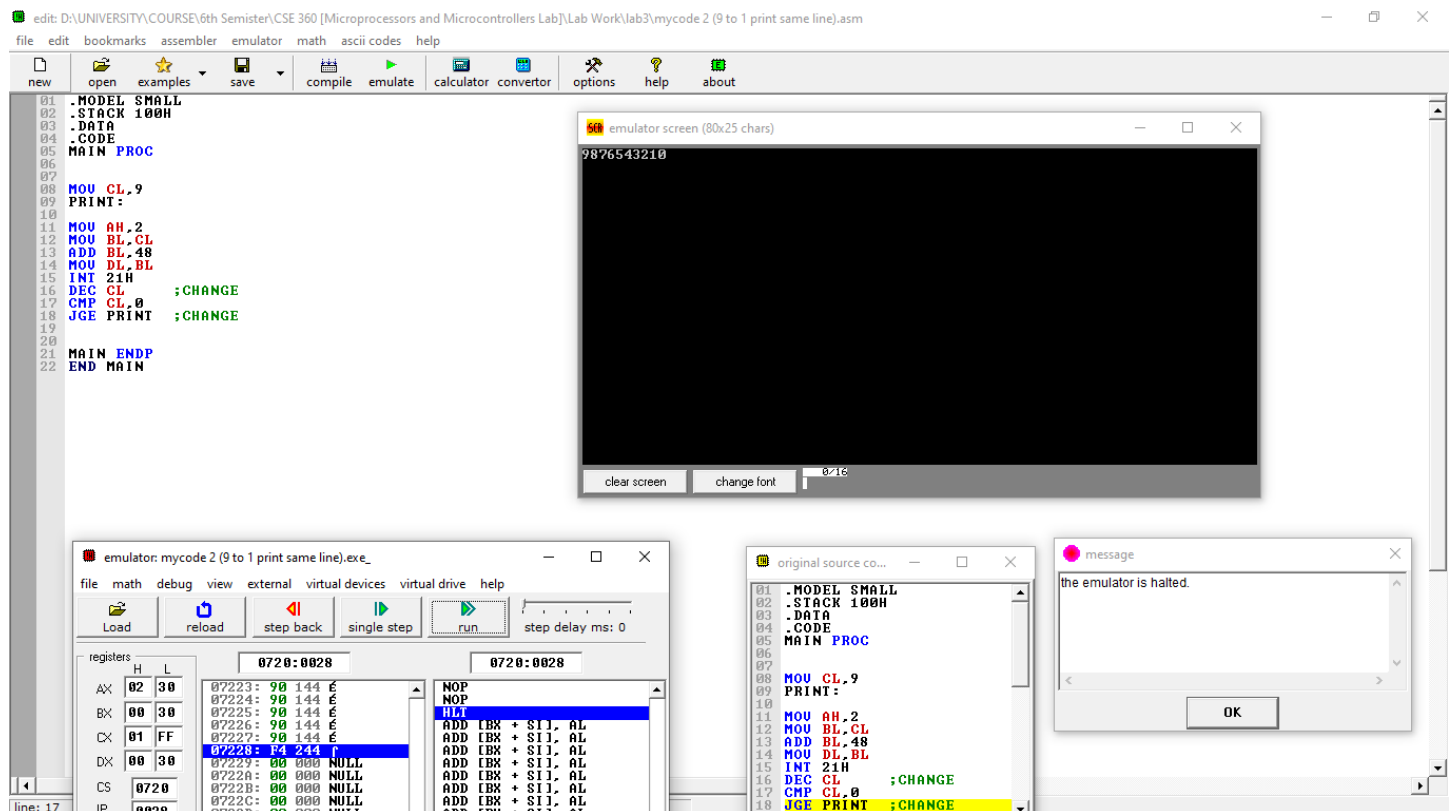
```
.MODEL SMALL
.STACK 100H
.DATA
.CODE
MAIN PROC

MOV CL,9
PRINT:

MOV AH,2
MOV BL,CL
ADD BL,48
MOV DL,BL
INT 21H
DEC CL      ;CHANGE
CMP CL,0
JGE PRINT  ;CHANGE

MAIN ENDP
END MAIN
```

Result:



Conclusion:

This assembly program prints the digits 9 to 0 in descending order on the screen. Here's a breakdown of its functionality:

Loop Initialization: The loop starts with CL = 9, and it prints the digit corresponding to CL by converting it to its ASCII value (by adding 48 to CL) and then printing it using the INT 21h DOS interrupt with function AH = 2.

Decrementing CL: After printing each digit, the program decrements CL (DEC CL) to move to the next lower digit. The loop continues as long as CL is greater than or equal to 0 (JGE PRINT).

Termination Condition: The loop ends when CL becomes negative (i.e., CL is less than 0), and the program stops.

The output displays the digits from 9 down to 0 in a single line. This program demonstrates basic control flow with loops and decrement operations in assembly language, along with formatted output using interrupts.

Problem Description: Printing of alphabets A to Z at Assembly code

Implementation:

```
.model small
.stack 100h
.code

main proc          ;main procedude

MOV CL,65
PRINT:

MOV AH,2
MOV BL,CL
          ;ADD BL,48
MOV DL,BL
INT 21H

mov dl,10
int 21h

mov dl,13
int 21h

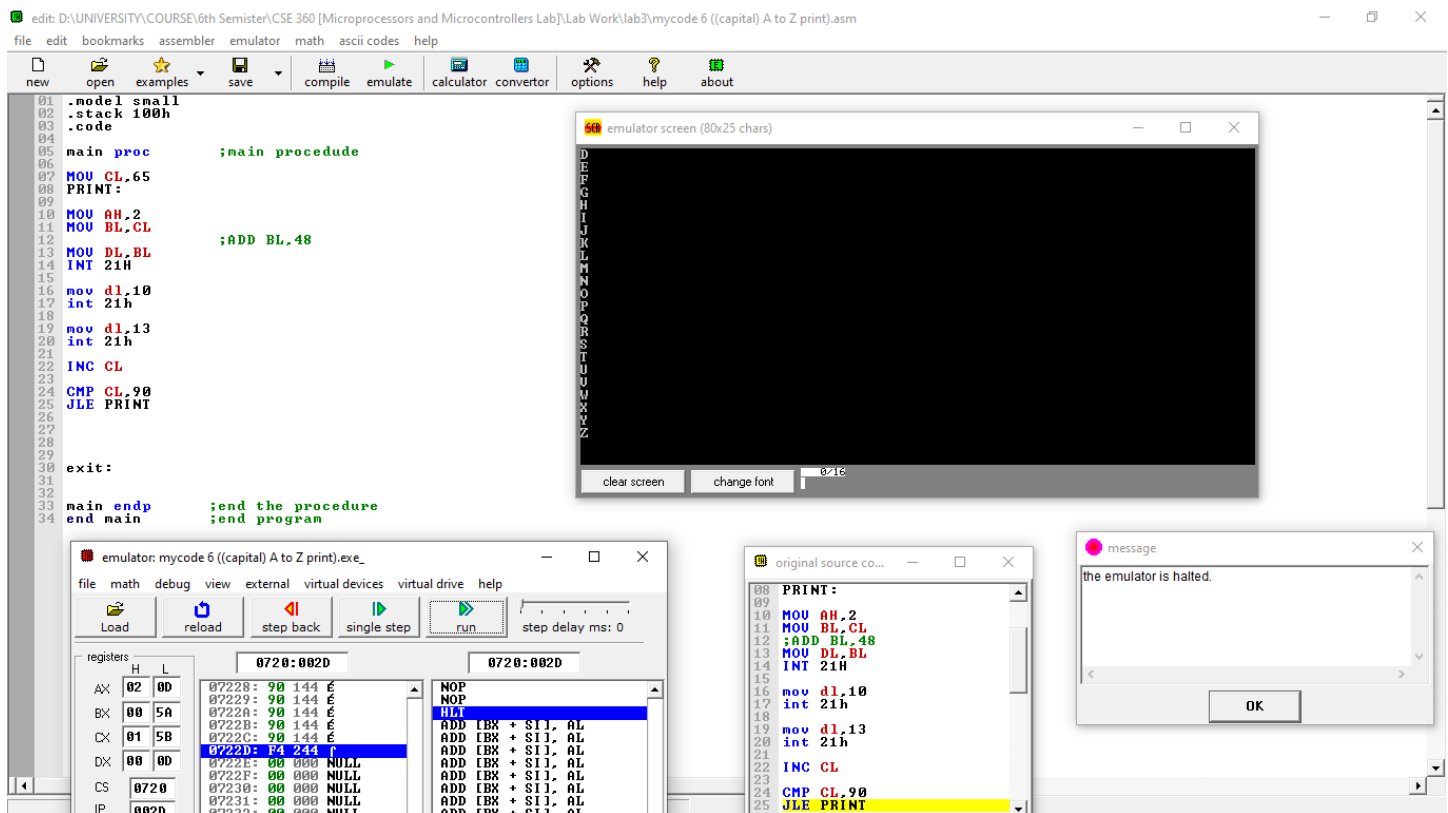
INC CL

CMP CL,90
JLE PRINT

exit:

main endp          ;end the procedure
end main           ;end program
```

Result:



Conclusion:

This assembly program prints the uppercase English alphabet (A to Z) on the screen. Here's how the program works:

Loop Initialization: The program starts with `CL = 65`, which corresponds to the ASCII value of the letter 'A'. It uses a loop to print each letter by moving the value of `CL` into the `BL` register and then into `DL` for display using the `INT 21h` DOS interrupt (function `AH = 2`).

Formatting: After printing each letter, the program outputs a newline (`DL = 10`) and a carriage return (`DL = 13`) to format the output with each letter printed on a new line.

Loop Control: The loop increments `CL` after each iteration, and continues printing letters until `CL` reaches 90 (the ASCII value of 'Z'). The loop terminates when `CL` exceeds 90.

The program prints the uppercase alphabet letters, each on a new line, demonstrating the use of loops, ASCII values, and formatted output in assembly language.

Problem Description: Printing of alphabets Z to A at Assembly code

Implementation:

```
.model small
.stack 100h
.code

main proc          ;main procedude

MOV CL,90
PRINT:

MOV AH,2
MOV BL,CL
          ;ADD BL,48
MOV DL,BL
INT 21H

mov dl,10
int 21h

mov dl,13
int 21h

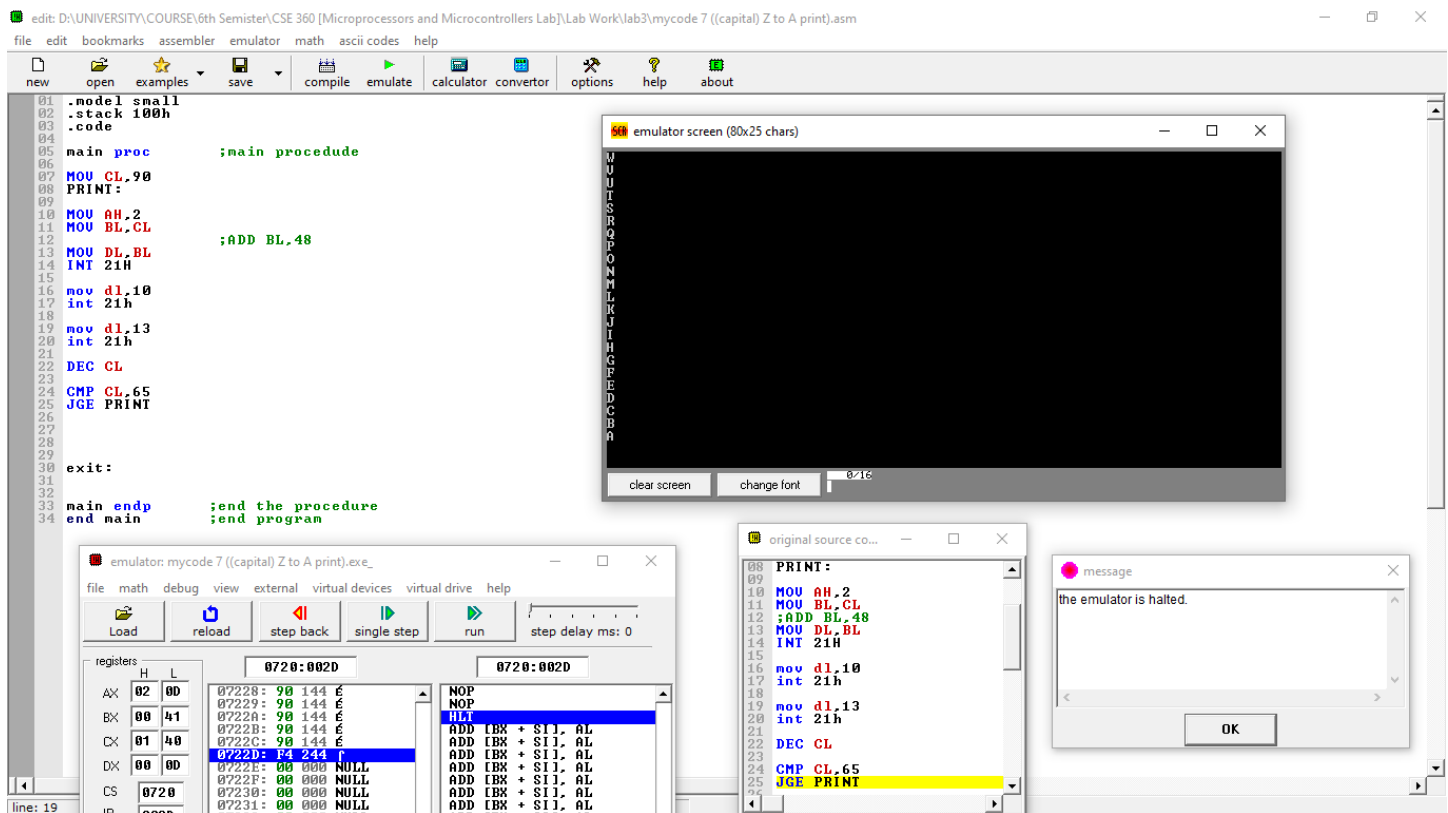
DEC CL

CMP CL,65
JGE PRINT

exit:

main endp          ;end the procedure
end main           ;end program
```


Result:



Conclusion:

This assembly program prints the uppercase English alphabet (Z to A) in reverse order on the screen. Here's a summary of the program's operation:

Loop Initialization: The program starts with `CL = 90`, which corresponds to the ASCII value of 'Z'. It uses a loop to print each letter by moving the value of `CL` into the `BL` register and then to `DL` for display using the `INT 21h` DOS interrupt (function `AH = 2`).

Formatting: After printing each letter, the program outputs a newline (`DL = 10`) and a carriage return (`DL = 13`) to format the output, ensuring that each letter appears on a new line.

Loop Control: The loop decrements `CL` after each iteration, continuing until `CL` reaches 65 (the ASCII value of 'A'). The loop stops once `CL` is less than 65.

The program successfully prints the letters from 'Z' to 'A' in reverse order, each on a new line. It demonstrates basic loop control, ASCII handling, and formatted output in assembly language.

Problem Description: Printing of alphabets a to z at Assembly code

Implementation:

```
.model small
.stack 100h
.code

main proc          ;main procedude

MOV CL,97
PRINT:

MOV AH,2
MOV BL,CL
          ;ADD BL,48
MOV DL,BL
INT 21H

mov dl,10
int 21h

mov dl,13
int 21h

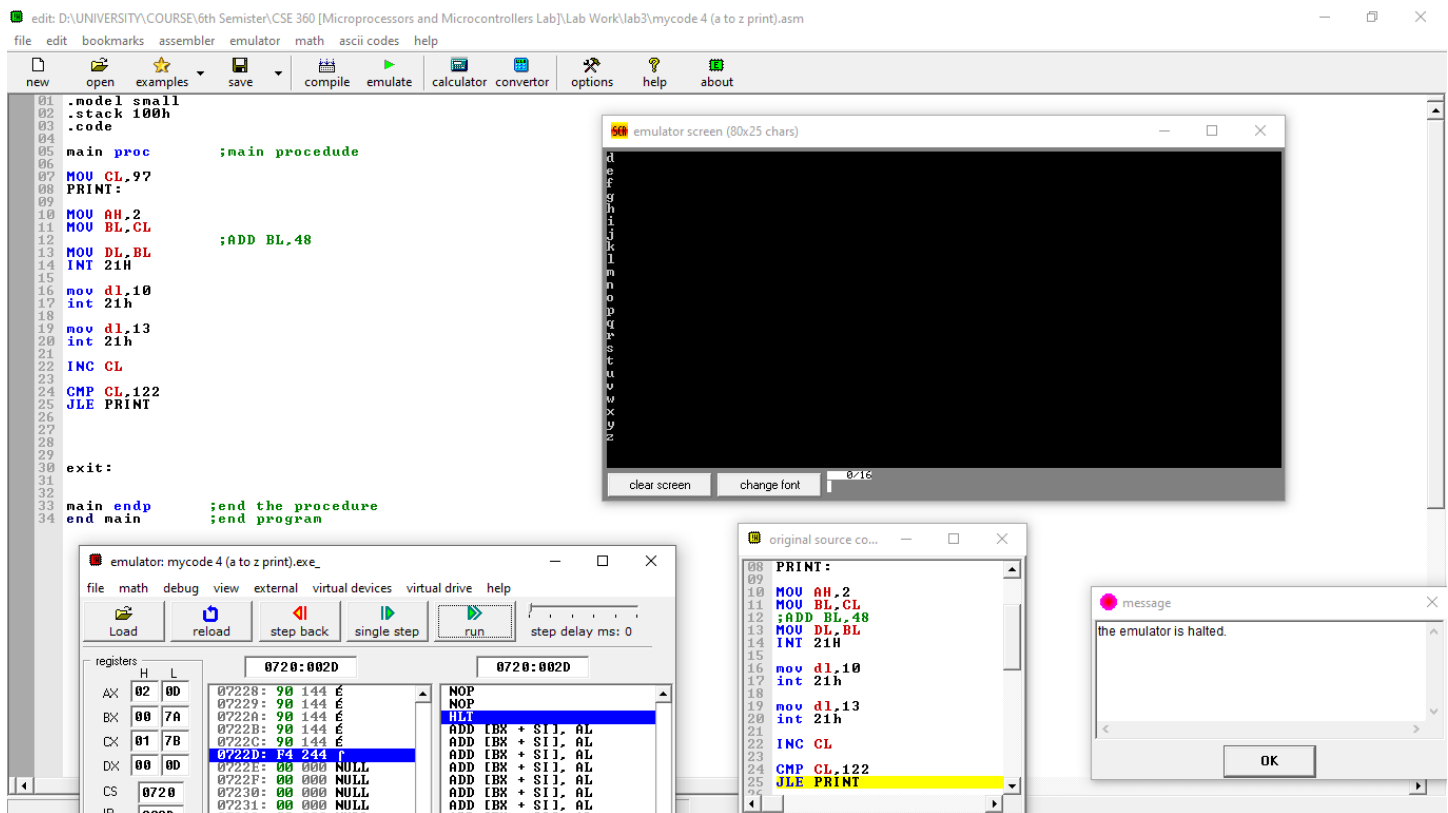
INC CL

CMP CL,122
JLE PRINT

exit:

main endp          ;end the procedure
end main           ;end program
```

Result:



Conclusion:

This assembly program prints the lowercase English alphabet (a to z) on the screen. Here's a breakdown of how the program works:

Loop Initialization: The program begins with `CL = 97`, which corresponds to the ASCII value of 'a'. The loop is used to print each letter by moving the value of `CL` into the `BL` register and then into `DL` for display using the `INT 21h` DOS interrupt (function `AH = 2`).

Formatting: After each letter is printed, the program outputs a newline (`DL = 10`) and a carriage return (`DL = 13`) to ensure that each letter is displayed on a new line.

Loop Control: The loop increments `CL` after each iteration, and continues printing letters until `CL` reaches 122 (the ASCII value of 'z'). The loop stops once `CL` exceeds 122.

The program successfully prints the lowercase alphabet from 'a' to 'z', each on a new line. It demonstrates basic loop control, ASCII handling, and formatted output in assembly language.

Problem Description: Printing of alphabets z to a at Assembly code

Implementation:

```
.model small
.stack 100h
.code

main proc      ;main procedude

MOV CL,122
PRINT:

MOV AH,2
MOV BL,CL
        ;ADD BL,48
MOV DL,BL
INT 21H

mov dl,10
int 21h

mov dl,13
int 21h

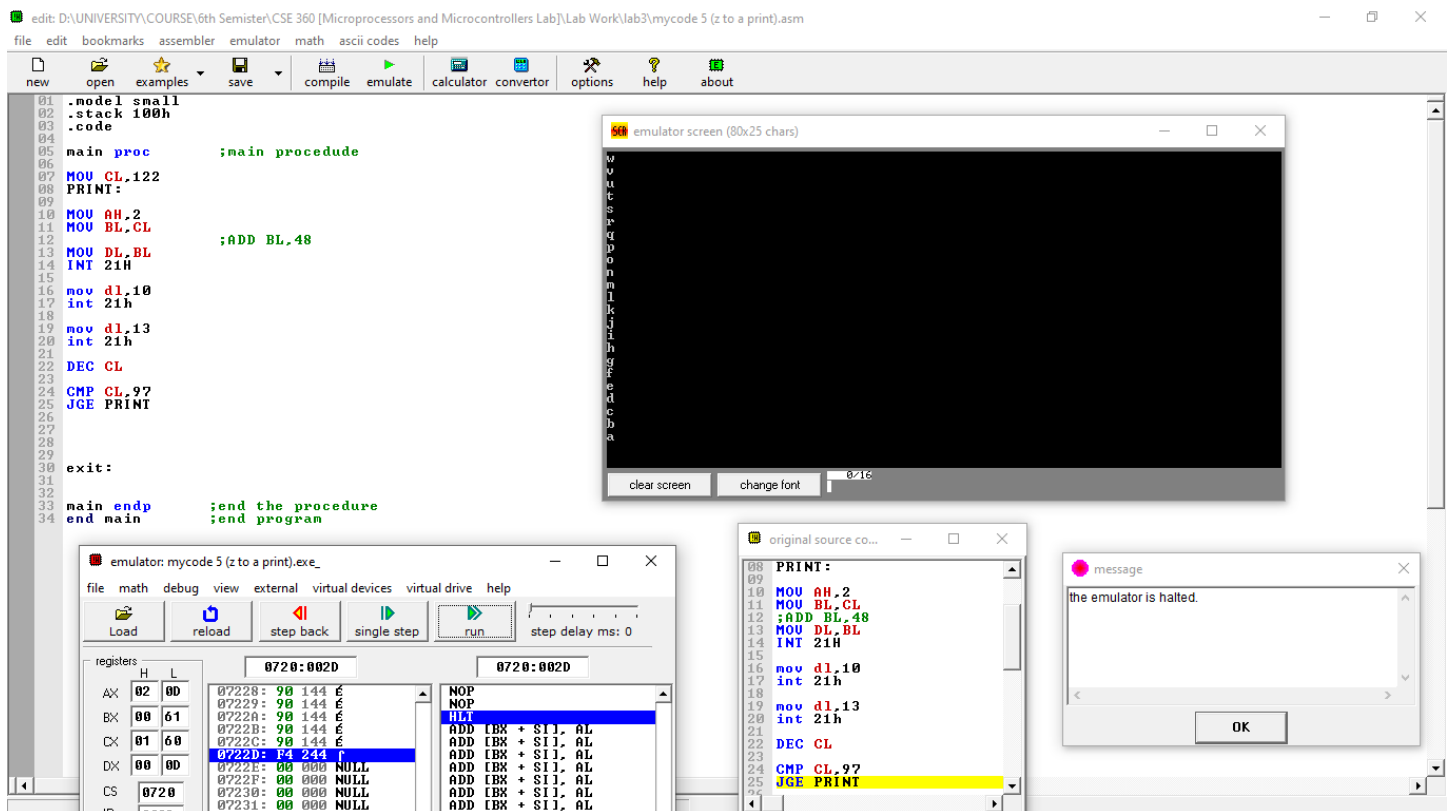
DEC CL

CMP CL,97
JGE PRINT

exit:

main endp      ;end the procedure
end main       ;end program
```

Result:



Conclusion:

This assembly program prints the lowercase English alphabet (z to a) in reverse order. Here's how the program works:

Loop Initialization: The program begins with CL = 122, which corresponds to the ASCII value of 'z'. The loop is used to print each letter by moving the value of CL into the BL register, then transferring it to DL for output using the INT 21h DOS interrupt (function AH = 2).

Formatting: After printing each letter, the program outputs a newline (DL = 10) and a carriage return (DL = 13) to ensure that each letter appears on a new line.

Loop Control: The loop decrements CL after each iteration and continues printing letters until CL reaches 97 (the ASCII value of 'a'). The loop terminates when CL becomes less than 97.

The program successfully prints the lowercase alphabet in reverse order, each letter displayed on a new line. It demonstrates basic loop control, ASCII handling, and formatted output in assembly language.