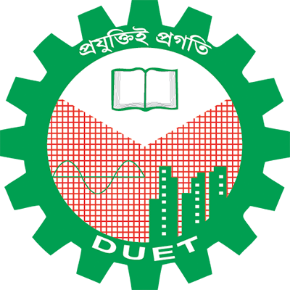
DHAKA UNIVERSITY OF ENGINEERING & TECHNOLOGY, GAZIPUR-1707

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



Course No: CSE 3812

Course Title: Microprocessor and Interfacing Sessional

**Lab 5**

**Report Name:** Logic, Shift and Rotate Instructions & Multiplication and Division Instructions in EMU8086.

Date of Allocation: 10/11/2024 Date of Submission: 23/11/2024

**Submitted To:**

Dr. Md. Jakirul Islam

Associate Professor, Department of CSE

Dhaka University of Engineering & Technology, Gazipur

Sumaya Khatun

Lecturer, Department of CSE

Dhaka University of Engineering & Technology, Gazipur

**Submitted By:**

Name: Imran Raza

Student ID: 214097

Year/Semester: 3rd Year/1st Semester

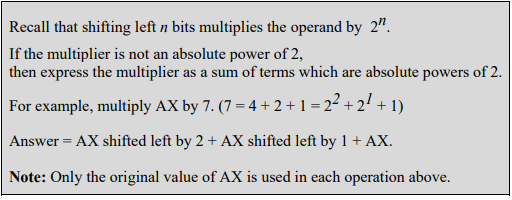
Section: B

**Objectives:**

To understanding Logic, Shift and Rotate Instructions & Multiplication and Division Instructions in EMU8086.

**Problem Statements:**

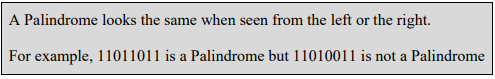
1. Write a program to multiply AX by 27 using only Shift and Add instructions. You should not use the MUL instruction.



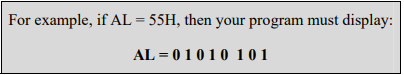
1. Write a program to divide AX by 8 using Shift instructions. You should not use the DIV instruction. Assume AX is a multiple of 8.



1. Write a program to check if a byte is a Palindrome. [Hint: Use Rotate instructions]. If the byte is a Palindrome, then move AAh into BL. Otherwise move 00h in BL.



1. Write a program to display the bits of a register or memory location. Use the INT 21H interrupts to display data on the display monitor. [Hint: Use logical shift instruction to move data bit into the carry flag]



1. Write assembly code for each of the following high-level language assignment statements. Suppose that A, B, and C are word variables and all products will fit in 16 bits. Use IMUL for multiplication. It's not necessary to preserve the contents of variables A, B, and C.
2. A = 5 x A – 7
3. B = (A-B) x (B-10)

**Problem 1:**

ORG 0100H

.DATA

N DW ?

RESULT DW ?

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

;INPUT

;FAST BX=0

XOR BX,BX

INPUT\_LOOP:

;CHAR INPUT

MOV AH,1

INT 21H

;IF\N\R,STOP TAKING INPUT

CMP AL,10

JE END\_INPUT\_LOOP

CMP AL,13

JE END\_INPUT\_LOOP

;FAST CHAR TO DIGIT

;ALSO CLEARS AH

AND AX,000FH

;SAVE AX

MOV CX,AX

;BX=BX\*10+AX

MOV AX,10

MUL BX

ADD AX,CX

MOV BX,AX

JMP INPUT\_LOOP

END\_INPUT\_LOOP:

MOV N,BX

XOR AX,AX

;SHITHING BY 4

MOV BX,N

MOV CX,4

SHL BX,CL

ADD AX,BX

;SHIFTING BY 3

MOV BX,N

MOV CX,3

SHL BX,CL

ADD AX,BX

;SHIFTING BY 1

MOV BX,N

SHL BX,1

ADD AX,BX

;ADD ONE MORE

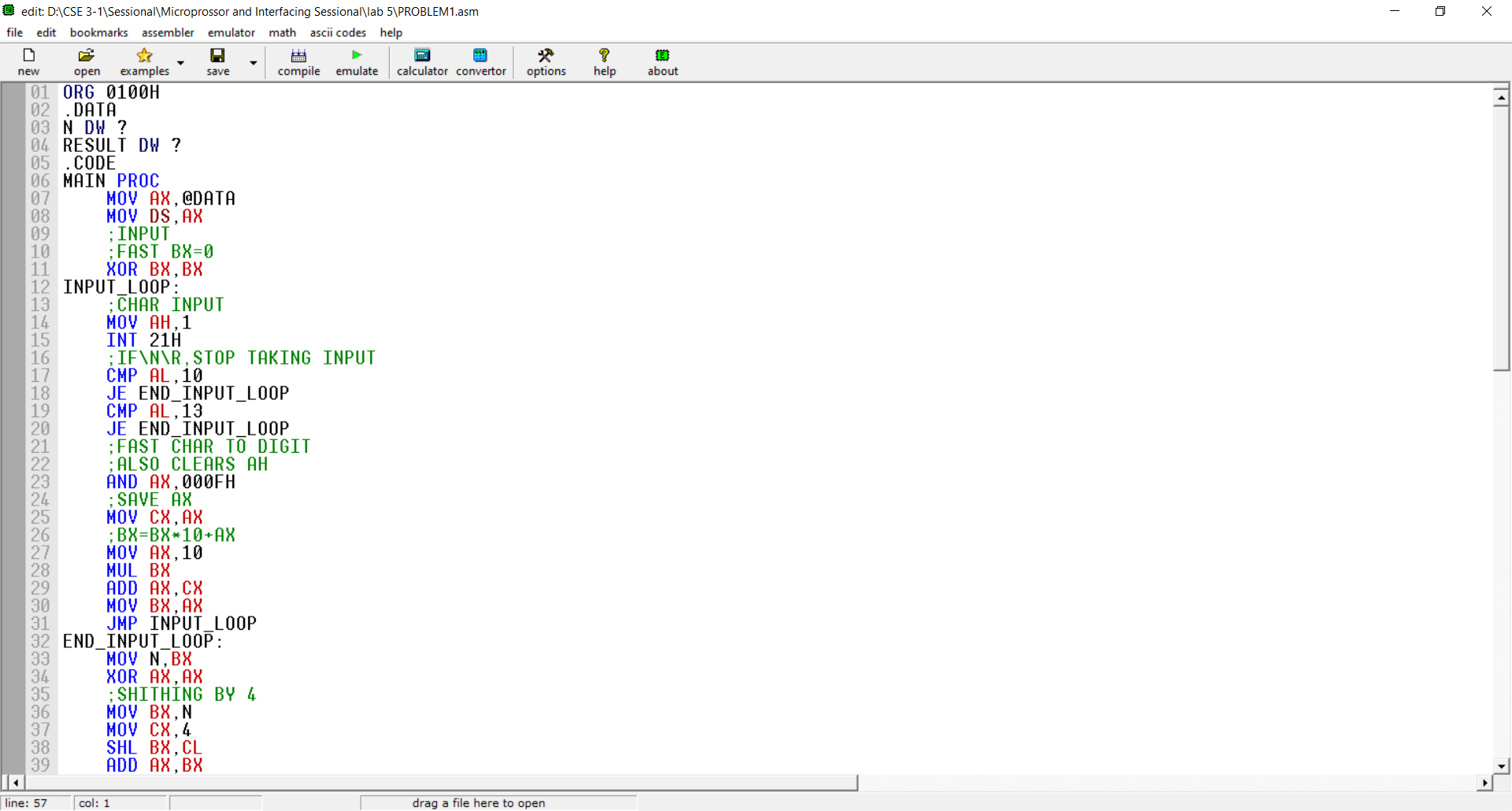
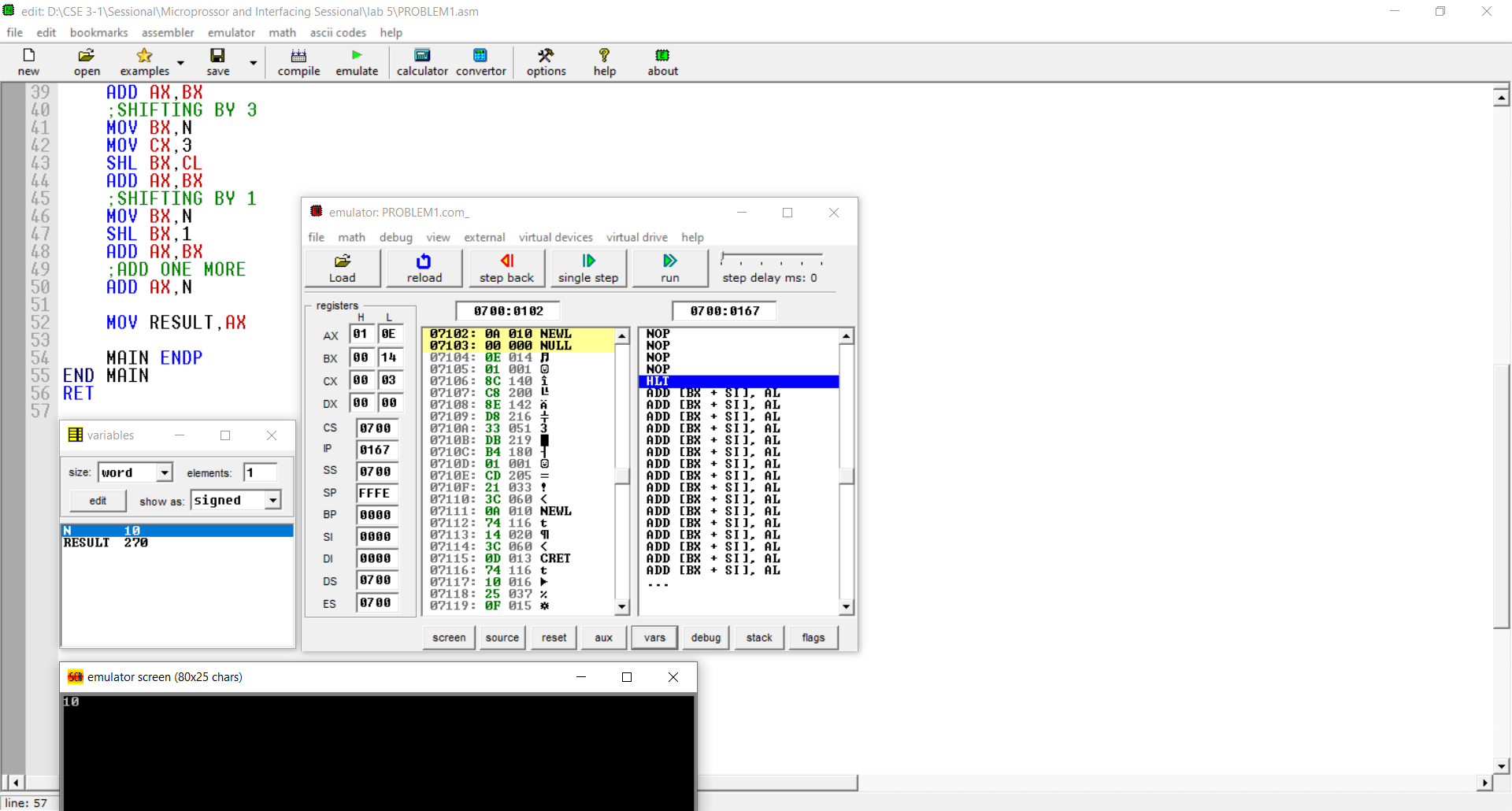
ADD AX,N

MOV RESULT,AX

MAIN ENDP

END MAIN

RET

**Problem 2:**

ORG 0100H

.DATA

N DW ?

RESULT DW ?

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

;INPUT

;FAST BX=0

XOR BX,BX

INPUT\_LOOP:

;CHAR INPUT

MOV AH,1

INT 21H

;IF\N\R,STOP TAKING INPUT

CMP AL,10

JE END\_INPUT\_LOOP

CMP AL,13

JE END\_INPUT\_LOOP

;FAST CHAR TO DIGIT

;ALSO CLEARS AH

AND AX,000FH

;SAVE AX

MOV CX,AX

;BX=BX\*10+AX

MOV AX,10

MUL BX

ADD AX,CX

MOV BX,AX

JMP INPUT\_LOOP

END\_INPUT\_LOOP:

MOV N,BX

XOR AX,AX

;SHITHING BY 3

MOV BX,N

MOV CX,3

SHR BX,CL

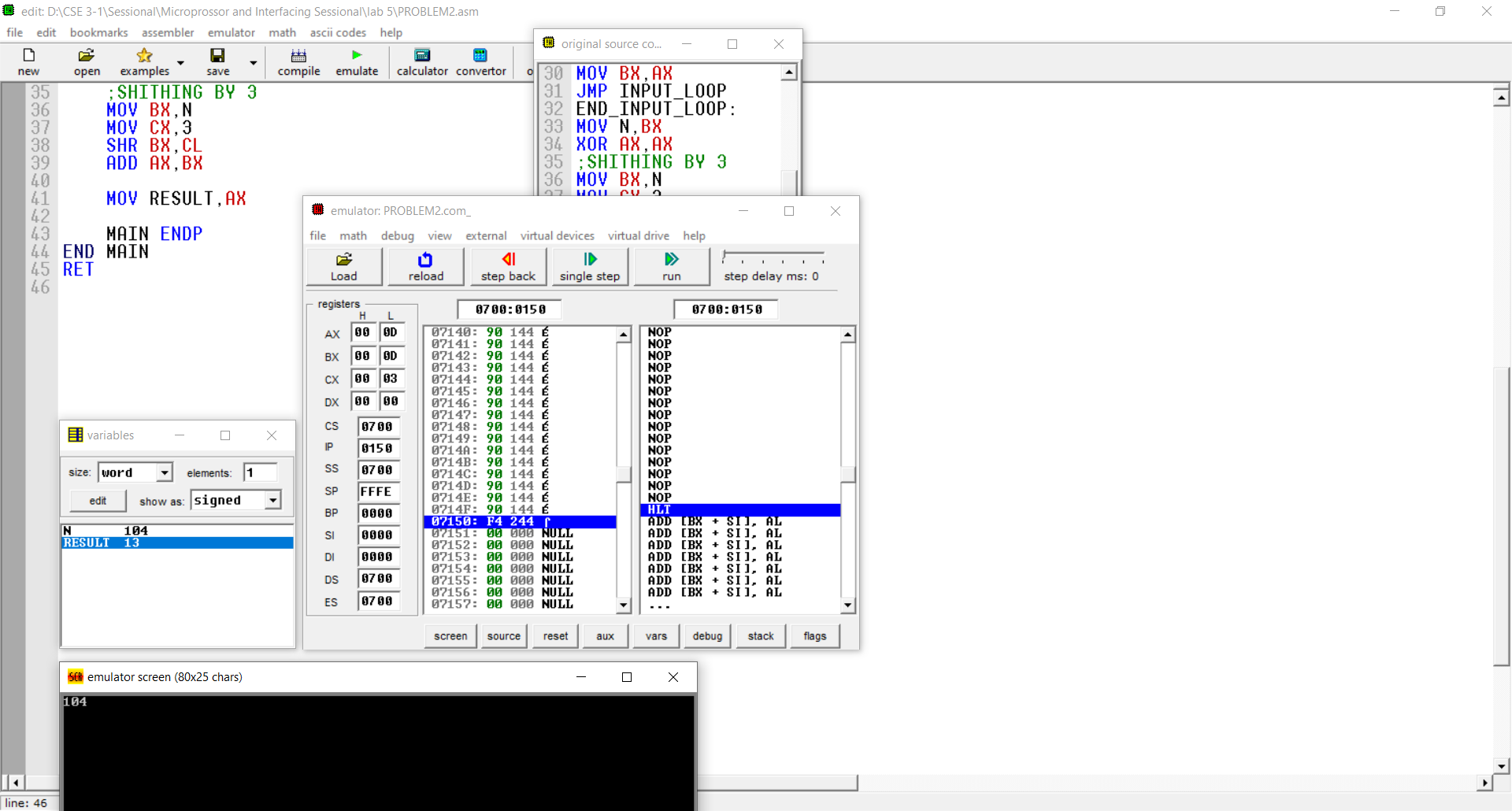
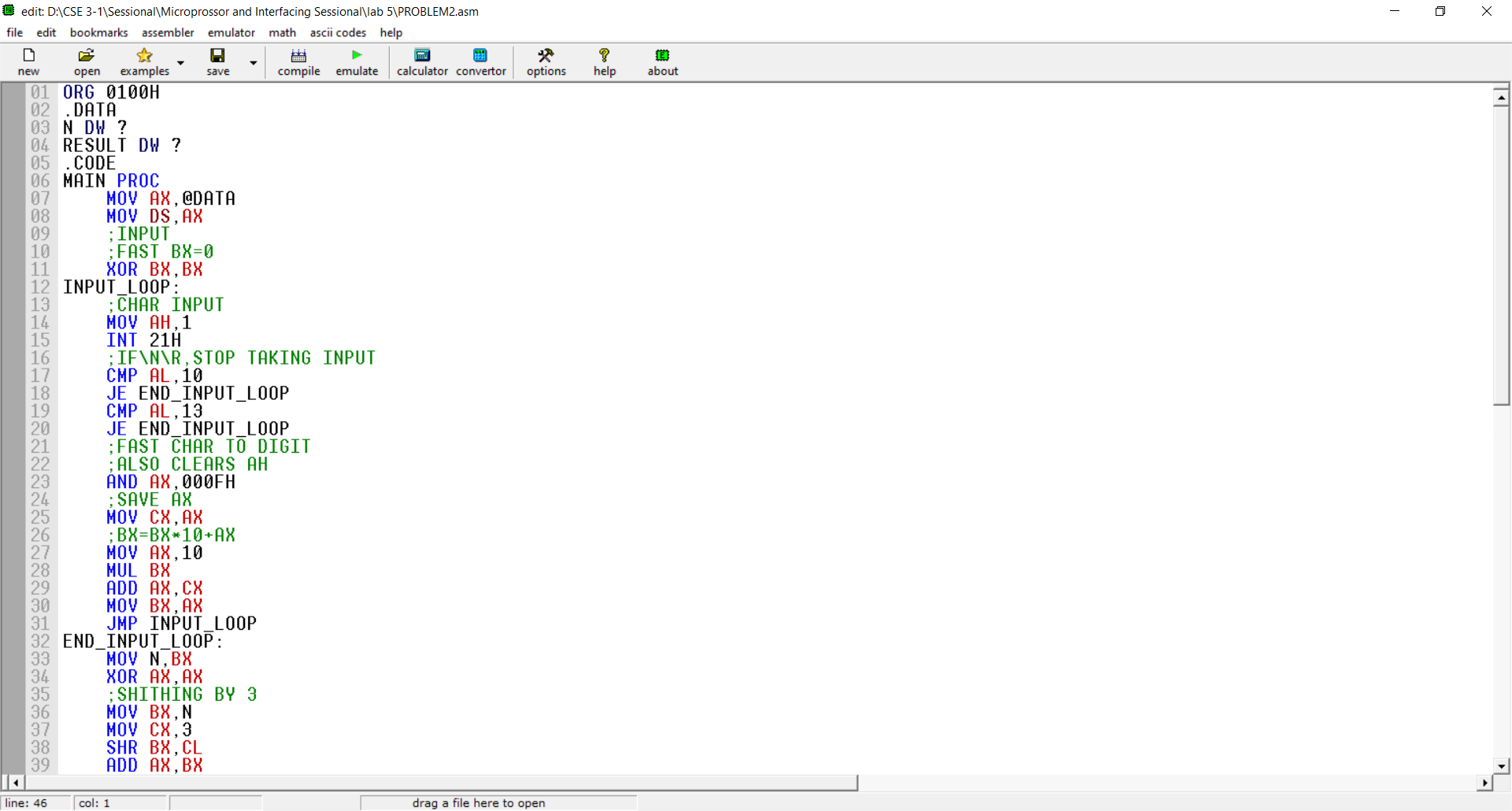
ADD AX,BX

MOV RESULT,AX

MAIN ENDP

END MAIN

RET



**Problem 3:**

ORG 0100H

.DATA

N DB 11011011B

RESULT DB ?

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

MOV AL,N

MOV BL,N

MOV CL,4

CHECK:

ROL AL,1

JC ONEL

MOV AH,0

ZEROL:

ROR BL,1

JC ONER

MOV BH,0

ZEROR:

CMP AH,BH

JNE NOT\_PALINDROME

DEC CL

JNZ CHECK

JMP PALINDROME

ONEL:

MOV AH,1

JMP ZEROL

ONER:

MOV BH,1

JMP ZEROR

PALINDROME:

MOV BL,0AAH

JMP END

NOT\_PALINDROME:

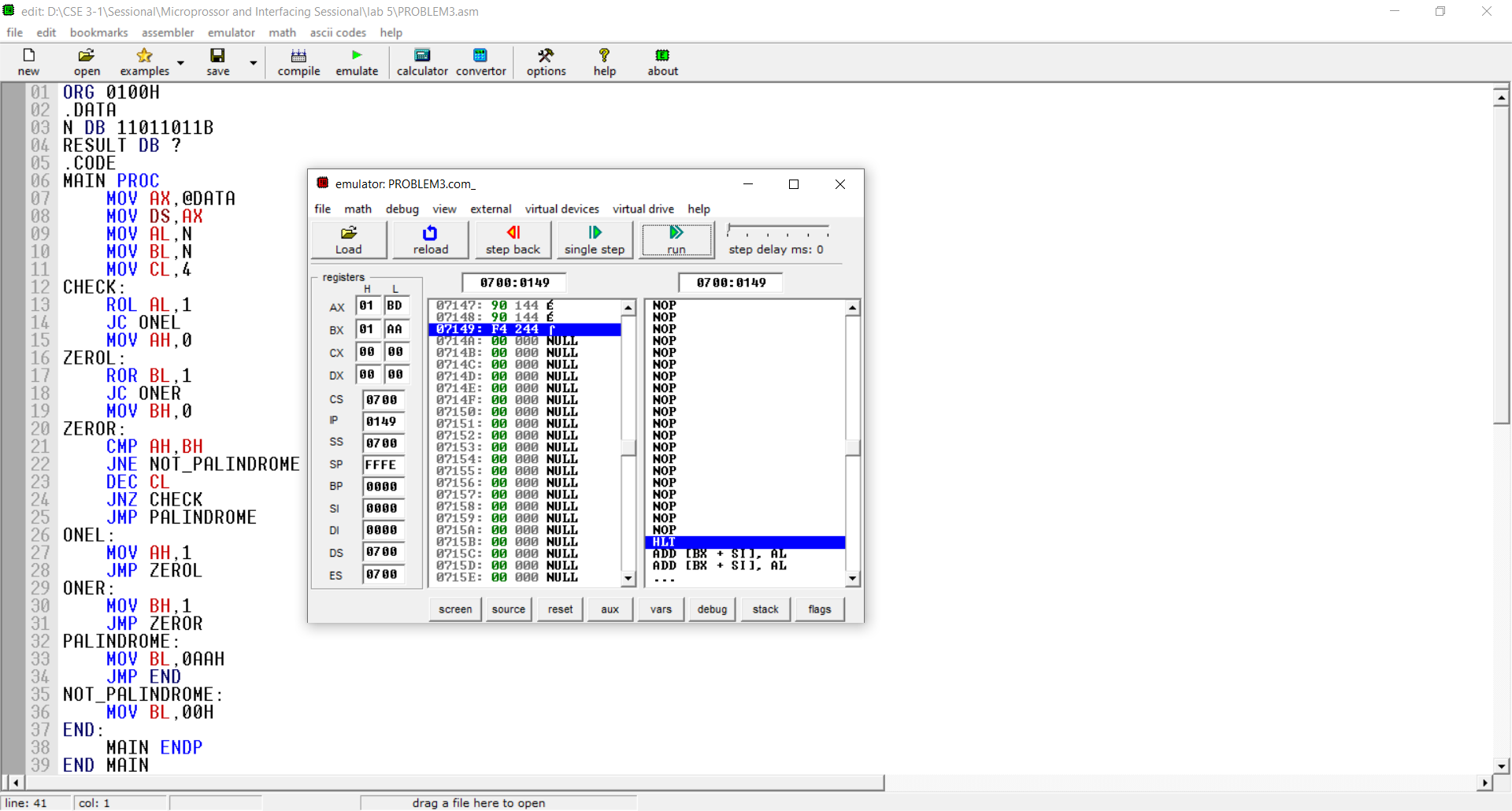
MOV BL,00H

END:

MAIN ENDP

END MAIN

RET



**Problem 4:**

ORG 0100H

.DATA

M DW 'AL = $'

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

;MSG

MOV AH,9

LEA DX,M

INT 21H

MOV CL,8

MOV AL,55H

MOV BL,AL

AGAIN:

ROL BL,1

JC ONE

MOV DL,'0'

JMP DISPLAY

ONE:

MOV DL,'1'

DISPLAY:

MOV AH,2

INT 21H

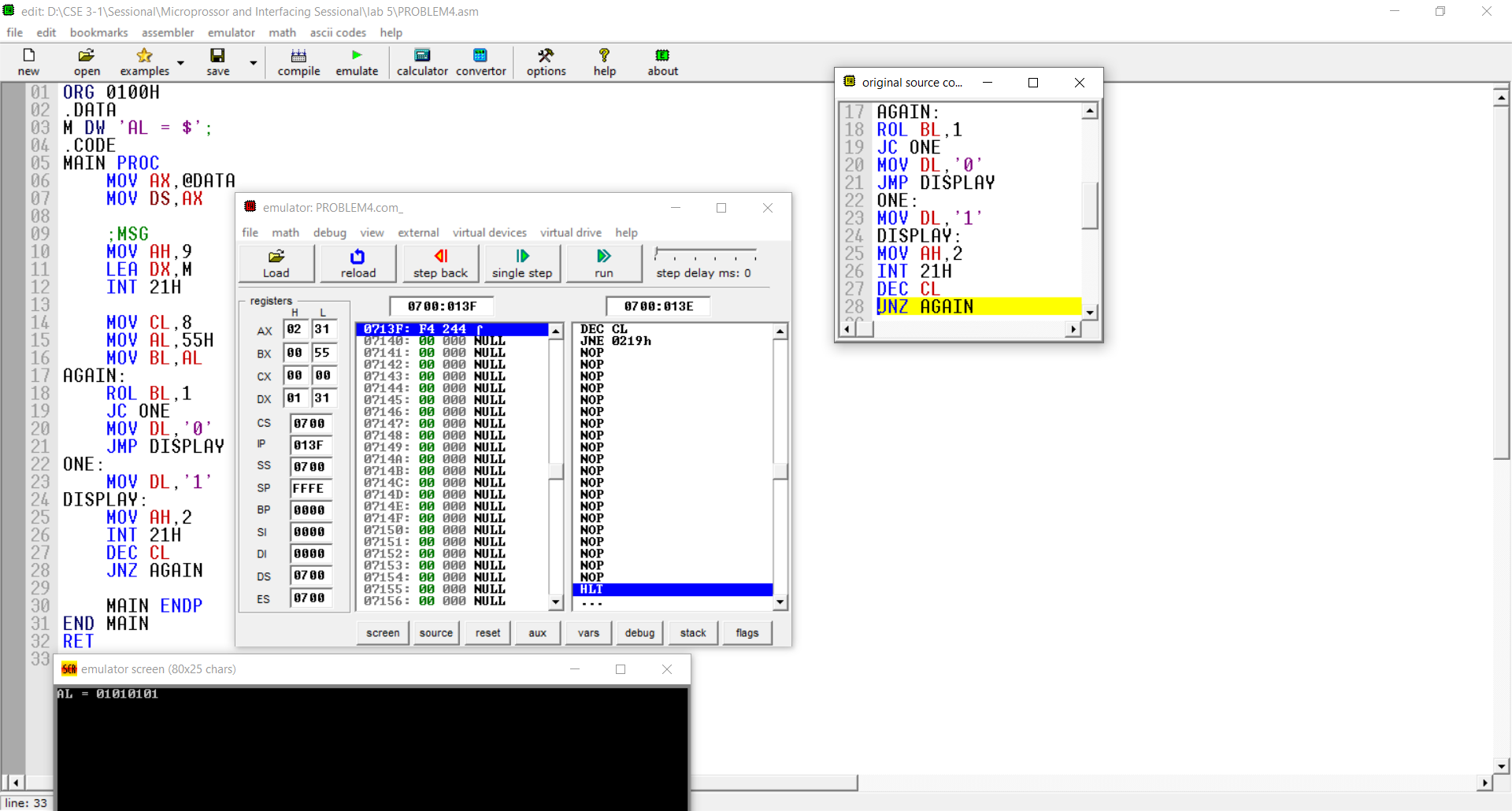
DEC CL

JNZ AGAIN

MAIN ENDP

END MAIN

RET



**Problem 5:**

ORG 0100H

.DATA

A DW ?

B DW ?

C DW ?

AM DW 'VALUE OF A: $'

BM DW 'VALUE OF B: $'

.CODE

MAIN PROC

MOV AX,@DATA

MOV DS,AX

;A MSG

MOV AH,9

LEA DX,AM

INT 21H

;A INPUT

;FAST BX=0

XOR BX,BX

INPUT\_LOOP:

;CHAR INPUT

MOV AH,1

INT 21H

;IF\N\R,STOP TAKING INPUT

CMP AL,10

JE END\_INPUT\_LOOP

CMP AL,13

JE END\_INPUT\_LOOP

;FAST CHAR TO DIGIT

;ALSO CLEARS AH

AND AX,000FH

;SAVE AX

MOV CX,AX

;BX=BX\*10+AX

MOV AX,10

MUL BX

ADD AX,CX

MOV BX,AX

JMP INPUT\_LOOP

END\_INPUT\_LOOP:

MOV A,BX

;NEW LINE

MOV AH,2

MOV DL,0AH

INT 21H

MOV DL,0DH

INT 21H

;B MSG

MOV AH,9

LEA DX,BM

INT 21H

;B INPUT

;FAST BX=0

XOR BX,BX

INPUT\_LOOP2:

;CHAR INPUT

MOV AH,1

INT 21H

;IF\N\R,STOP TAKING INPUT

CMP AL,10

JE END\_INPUT\_LOOP2

CMP AL,13

JE END\_INPUT\_LOOP2

;FAST CHAR TO DIGIT

;ALSO CLEARS AH

AND AX,000FH

;SAVE AX

MOV CX,AX

;BX=BX\*10+AX

MOV AX,10

MUL BX

ADD AX,CX

MOV BX,AX

JMP INPUT\_LOOP2

END\_INPUT\_LOOP2:

MOV B,BX

MOV AX,5

MOV BX,A

IMUL BX

SUB AX,7

MOV A,AX

MOV AX,A

SUB AX,B

MOV BX,B

SUB BX,10

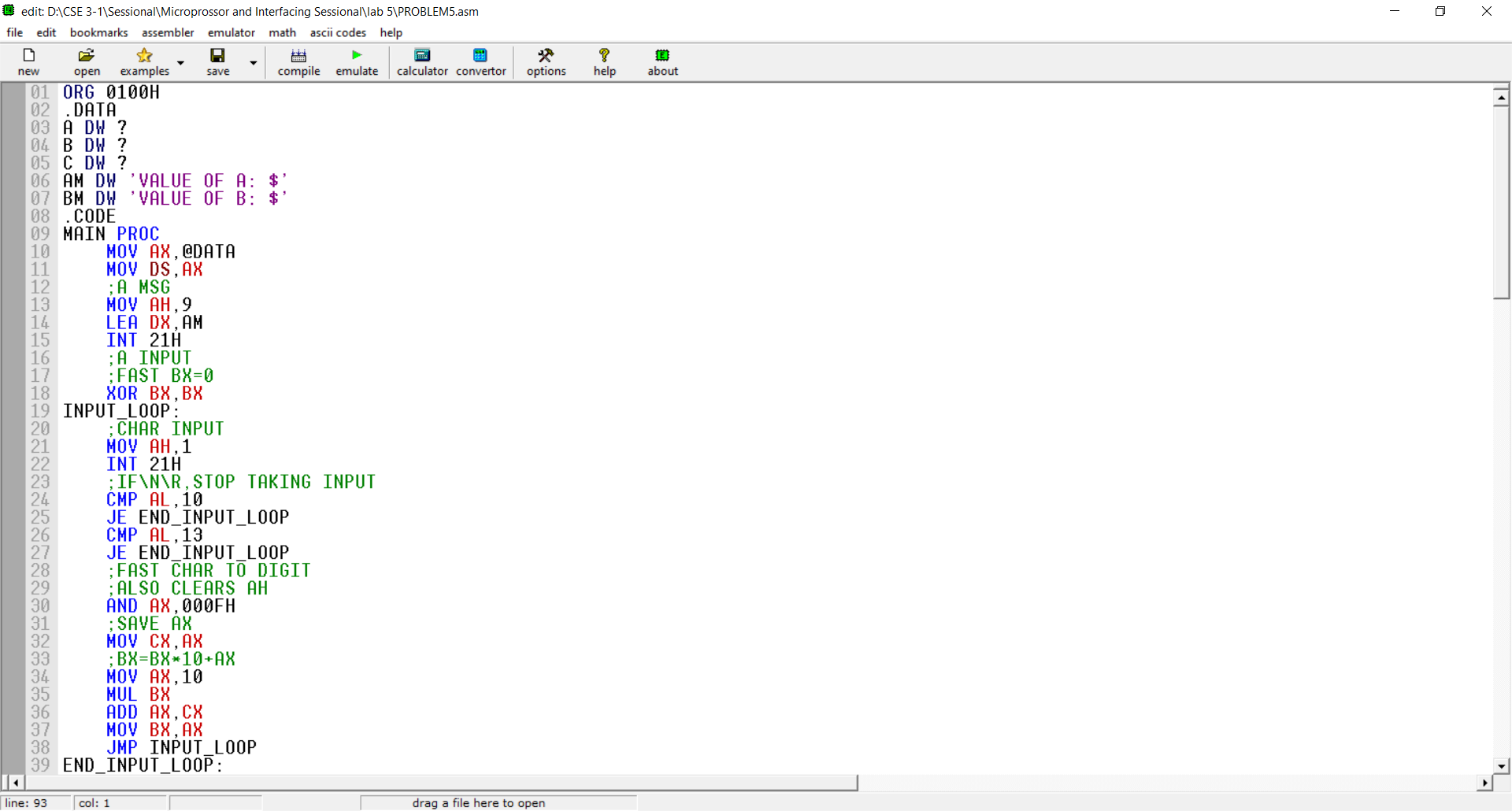
IMUL BX

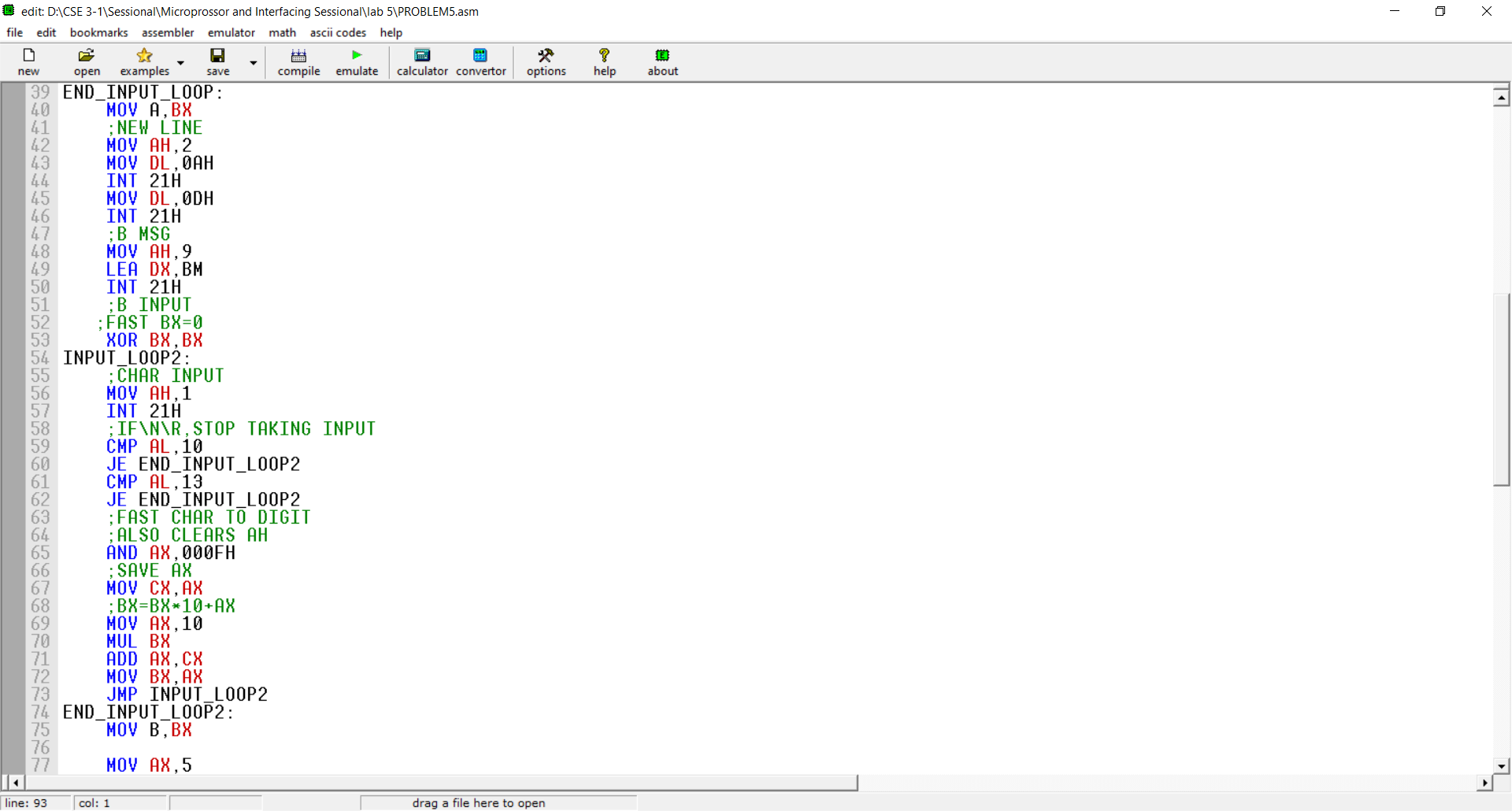
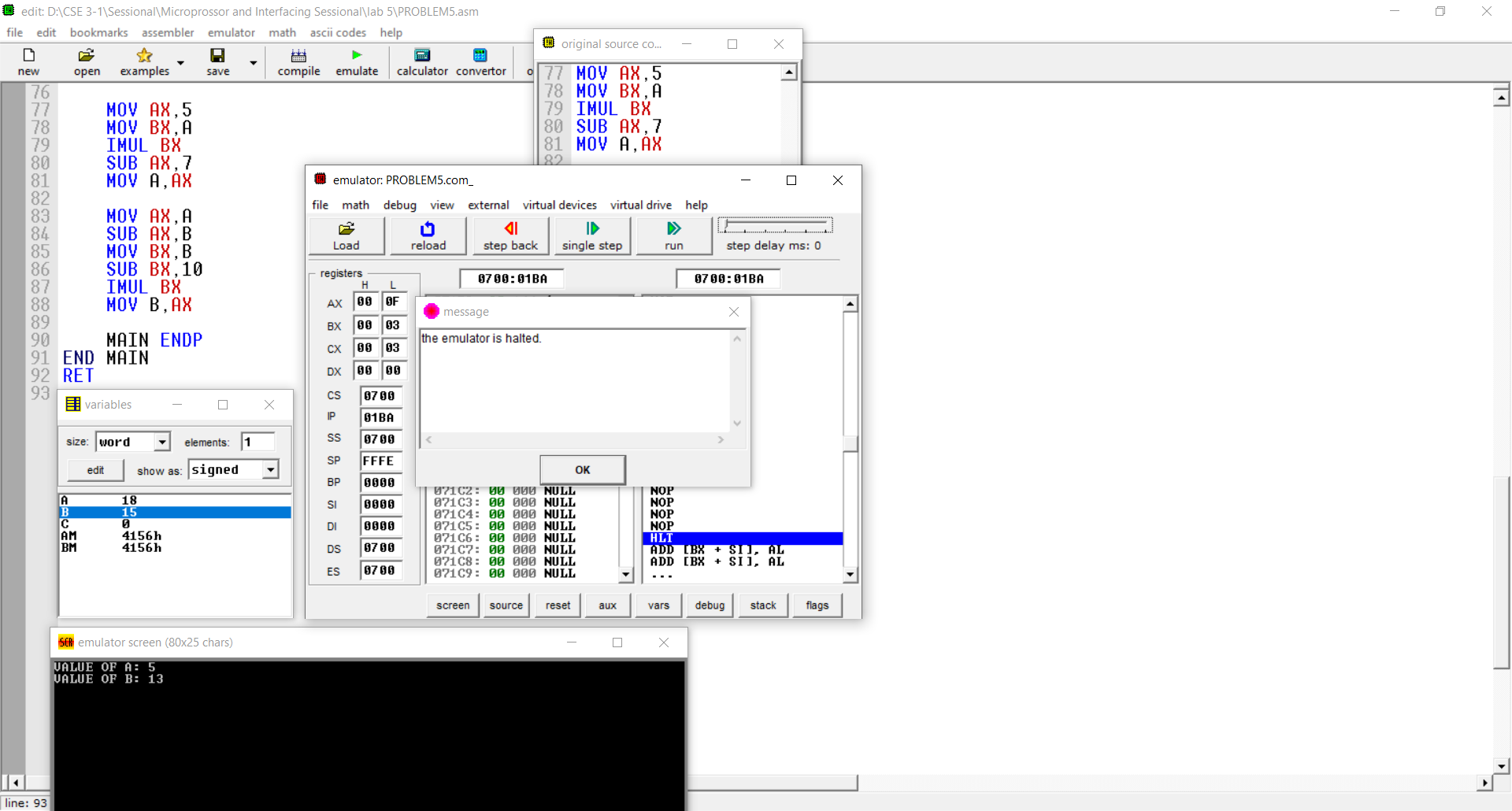
MOV B,AX

MAIN ENDP

END MAIN

RET



**Discussion:**

Finally it can be said that, we learned about so many new instruction called Logic, Shift and Rotate Instructions & Multiplication and Division Instructions. By using those instruction, we are able solve new logical problem like Palindrome Check or Byte Display etc.