



Assignment

Microprocessor and Assembly Language Lab
CSEL-3106

Submitted by

Md. Waliul Islam Rayhan
ID: B190305034

Submitted to

Arnisha Akhter
Lecturer
Dept. of CSE, Jagannath University

emu8086 - assembler and microprocessor emulator 4.08

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator converter options help about

```

20 MOU [BX], AL
21 INC BX
22 LOOP INPUT
23
24 MOU CX, ?
25 DEC CX
26
27 OUTERLOOP:
28 MOU BX,CX
29 MOU SI,0
30
31 COMPLOOP:
32 MOU AL,ARR[SI]
33 MOU DL,ARR[SI+1]
34 CMP AL,DL
35
36 JC NOSWAP
37
38 MOU ARR[SI],DL
39 MOU ARR[SI+1],AL
40
41 NOSWAP:
42 INC SI
43 DEC BX
44 JNZ COMPLOOP
45
46 LOOP OUTERLOOP
47 ;NEW LINE
48
49 MOU AH,2
50 MOU DL,10
51 INT 21H
52
53 MOU DL,13
54 INT 21H
55
56 PRINT "AFTER SORTING"
57
58 MOU CX,?
59 MOU BX,OFFSET ARR
60
61 ;THIS LOOP TO DISPLAY EL
62 OUTPUT:
63 MOU DL,[BX]
64 MOU AH,2
65 INT 21H
66
67 MOU DL,32
68 MOU AH,2
69 INT 21H
70
71 INC BX
72 LOOP OUTPUT
73
74 MAIN ENDP
75
76 RET

```

original source c... file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers F400:0154 F400:0154

	H	L
AX	02	20
BX	01	09
CX	00	00
DX	00	20
CS	F400	
IP	0154	
SS	0700	
SP	FFFA	
BP	0000	
SI	0001	
DI	0000	
DS	0700	
ES	0700	

F4150: FF 255 RES
F4151: FF 255 RES
F4152: CD 205 =
F4153: 20 032 SPA
F4154: CF 207 ↳
F4155: 00 000 NULL
F4156: 00 000 NULL
F4157: 00 000 NULL
F4158: 00 000 NULL
F4159: 00 000 NULL
F415A: 00 000 NULL
F415B: 00 000 NULL
F415C: 00 000 NULL
F415D: 00 000 NULL
F415E: 00 000 NULL
F415F: 00 000 NULL
F4160: FF 255 RES
F4161: FF 255 RES
F4162: CD 205 =
F4163: 1A 026 →
F4164: CF 207 ↳

BIOS DI
INT 020h
LRET
ADD BX + SI, AL
ADD BH, BH
DEC BP
SBB CL, BH
ADD BX + SI, AL
ADD BH, CL
ADD BX + SI, AL

screen source reset aux vars debug stack flags

message
PROGRAM HAS RETURNED CONTROL
TO THE OPERATING SYSTEM

OK

line: 76 col: 4 drag a file here to open

Search

12:18 PM 7/19/2023

Screenshot: 01 — 1. Sort numbers in Ascending Order (1)

The screenshot shows the emu8086 assembly editor and emulator interface. The assembly code is as follows:

```
emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator converter options help about
original source c...
19 MOU [BX], AL
20 INC BX
21 LOOP INPUTS
22
23 MOU CX, ?
24 DEC CX
25
26 OUTERLOOP:
27 MOU BX,CX
28 MOU SI,0
29
30 COMLOOP:
31 MOU AL,ARR[SI]
32 MOU DL,ARR[SI+1]
33 CMP AL,DL
34
35 JNC NOSWAP
36
37 MOU ARR[SI],DL
38 MOU ARR[SI+1],AL
39
40 NOSWAP:
41 INC SI
42 DEC BX
43 JNZ COMLOOP
44
45 LOOP OUTERLOOP
46 ;NEW LINE
47
48 MOU AH,2
49 MOU DL,10
50 INT 21H
51
52 MOU DL,13
53 INT 21H
54
55 PRINT "AFTER SORTING"
56
57 MOU CX,?
58 MOU BX,OFFSET ARR
59
60 :THIS LOOP TO DISPLAY
61 OUTPUTS:
62 MOU DL,[BX]
63 MOU AH,2
64 INT 21H
65
66 MOU DL,32
67 MOU AH,2
68 INT 21H
69
70 INC BX
71 LOOP OUTPUTS
72
73 MAIN ENDP
74 RET
```

The emulator screen (80x25 chars) displays:

```
ENTER ? NUMBER IN ARRAY:2359876
AFTER SORTING ARRAY IN DESSENDING ORDER:9 8 7 6 5 3 2
```

The registers window shows:

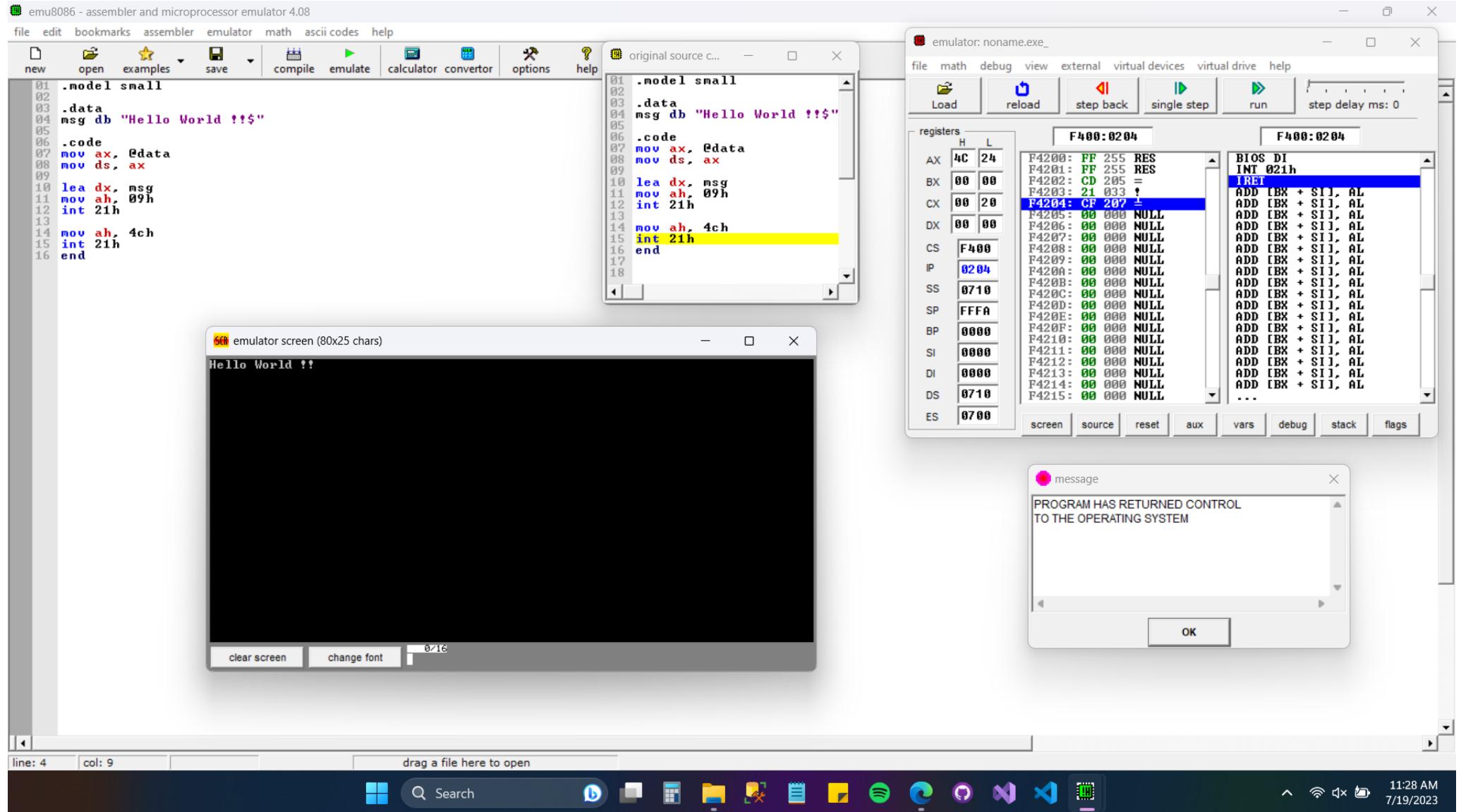
	H	L
AX	02	20
BX	01	09
CX	00	00
DX	00	20
CS	F400	
IP	0154	
SS	0700	
SP	FFFA	
BP	0000	
SI	0001	
DI	0000	
DS	0700	
ES	0700	

The stack window shows:

	F400:0154	F400:0154
F4150:	FF 255 RES	BIOS DI
F4151:	FF 255 RES	INT 029h
F4152:	CD 205 =	IRET
F4153:	20 032 SPA	ADD [BX + SI], AL
F4154:	CF 207 ±	ADD [BX + SI], AL
F4155:	00 000 NULL	ADD [BX + SI], AL
F4156:	00 000 NULL	ADD [BX + SI], AL
F4157:	00 000 NULL	ADD [BX + SI], AL
F4158:	00 000 NULL	ADD [BX + SI], AL
F4159:	00 000 NULL	DEC BP
F415A:	00 000 NULL	SBB CL, BH
F415B:	00 000 NULL	ADD [BX + SI], AL
F415C:	00 000 NULL	ADD [BX + SI], AL
F415D:	00 000 NULL	ADD [BX + SI], AL
F415E:	00 000 NULL	ADD [BX + SI], AL
F415F:	00 000 NULL	ADD [BX + SI], AL
F4160:	FF 255 RES	ADD BH, BH
F4161:	FF 255 RES	DEC BP
F4162:	CD 205 =	ADD BH, CL
F4163:	1A 026 →	ADD [BX + SI], AL
F4164:	CF 207 ±	...

A message box says: PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM.

Screenshot: 02 – 1. Sort numbers in Descending Order (2)



Screenshot: 03 — 4. Program to display a string.

The screenshot shows the emu8086 assembly editor and emulator interface. The main window displays assembly code for finding the smallest value in an array. Key sections include:

- ASM Editor:** Shows the assembly code with labels like .SMALLEST, .MODEL SMALL, .STACK, .DATA, .CODE, MAIN PROC, and SMALLESTP PROC.
- Registers Window:** Displays CPU registers (AX, BX, CX, DX, CS, IP, SS, SP, BP, SI, DI, DS, ES) with their current values.
- Memory Dump Window:** Shows memory dump starting at address 07200, displaying BIOS data and INT 021h.
- Variables Window:** Shows variables ARR (size: byte, elements: 1) containing the value 5, and SMALLEST containing the value 2.
- Emulator Screen:** A terminal-like window showing the output of the program execution.
- Message Window:** A dialog box stating "PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM".

The status bar at the bottom indicates "line: 69 col: 4" and the system tray shows various icons including a search bar, file explorer, and system notifications.

Screenshot: 04 – 5. Find the smallest number.

The screenshot displays the emu8086 development environment with multiple windows open:

- Assembly Editor:** Shows the assembly code for the program. The code initializes variables, loops through input, finds the largest value, and stores it. It uses `MOU` and `INT 21H` instructions.
- Registers Window:** Displays the CPU register values at address 07200:0000. Registers include AX, BX, CX, DX, CS, IP, SS, SP, BP, SI, DI, DS, and ES. Many registers show values related to the input array.
- Memory Dump:** Shows the memory dump starting at address F400:0204. It contains BIOS data and INT 021h interrupt vectors.
- Variables Window:** Shows the variable table with entries for ARR (size: byte, elements: 1) containing the value 5, and LARGEST (size: byte, elements: 1) containing the value 9.
- Emulator Screen:** A window titled "emulator screen (80x25 chars)" showing the output of the program. The screen displays the value 56894.
- Message Dialog:** A "message" dialog box stating "PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM" with an "OK" button.

The system tray at the bottom shows standard icons for file explorer, search, task manager, and other system utilities. The status bar indicates the current time as 12:26 PM and date as 7/19/2023.

Screenshot: 05 — 6. Find the largest number.

emu8086 - assembler and microprocessor emulator 4.08

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator converter options help

```

001 ; this is a program in 8086 assembly language that
002 ; accepts a character string from the keyboard and
003 ; stores it in the string array. the program then converts
004 ; all the lower case characters of the string to upper case.
005 ; if the string is empty (null), it doesn't do anything.
006 ; name "upper"
007
008 org 100h
009
010 jmp start
011
012
013
014
015
016 ; first byte is buffer size,
017 ; second byte will hold number
018 ; of used bytes for string.
019 ; all other bytes are for characters:
020 string db 20, 22 dup('?')
021
022 new_line db 0Dh,0Ah, '$' ; new line code.
023
024 start:
025
026 ; int 21h / ah=0Ah - input of a string to ds:dx,
027 ; fist byte is buffer size, second byte is number
028 ; of chars actually read. does not add '$' in the
029 ; end of string. to print using int 21h / ah=09h
030 ; you must set dollar sign at the end of it and
031 ; start printing from address ds:dx + 2.
032
033 lea dx, string
034
035 mov ah, 0Ah
036 int 21h
037
038 mov bx, dx
039 mov ah, 0
040 mov al, ds:[bx+1]
041 add bx, ax ; point to end of string.
042
043 mov byte ptr [bx+2], '$' ; put dollar to the end.
044
045 ; int 21h / ah=09h - output of a string at ds:dx.
046 ; string must be terminated by '$' sign.
047 lea dx, new_line
048 mov ah, 09h
049 int 21h
050
051
052 lea bx, string
053
054 mov ch, 0
055 mov cl, [bx+1] ; get string size.
056
057 jcxz null ; is string is empty?
058

```

original source c... file math debug view external virtual devices virtual drive help

Load Reload step back single step run step delay ms: 0

registers		F400:0154	F400:0154
AX	1C 0D	F4150: FF 255 RES	BIOS DI
BX	01 0A	F4151: FF 255 RES	INT 020h
CX	00 00	F4152: CD 205 =	IRET
DX	01 04	F4153: 20 032 SPA	ADD [BX + SI], AL
CS	F400	F4154: CF 207 =	ADD [BX + SI], AL
IP	0154	F4155: 00 000 NULL	ADD [BX + SI], AL
SS	0700	F4156: 00 000 NULL	ADD [BX + SI], AL
SP	FFFA	F4157: 00 000 NULL	ADD [BX + SI], AL
BP	0000	F4158: 00 000 NULL	ADD BH, BH
SI	0000	F4159: 00 000 NULL	DEC BP
		F415A: 00 000 NULL	SBB CL, BH
		F415B: 00 000 NULL	ADD [BX + SI], AL
		F415C: 00 000 NULL	ADD [BX + SI], AL
		F415D: 00 000 NULL	...

screen source reset aux vars debug stack flags

emulator screen (80x25 chars)

rayhan RAYHAN

clear screen change font 8/16

line: 101 col: 2 drag a file here to open

Search

11:44 AM 7/19/2023

message

PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM

OK

Screenshot: 06 – 7. Convert lowercase letter to uppercase.

The screenshot displays the emu8086 interface with several windows open:

- Assembly Editor:** Shows the assembly code for a program that prompts for two numbers, adds them, and prints the result. The code uses BIOS interrupt 21h for input and output.
- Registers Window:** Displays the CPU registers (AX, BX, CX, DX, CS, IP, SS, SP, BP, SI) and memory dump (F400:F4204). The instruction at F400:0204 is highlighted as RET.
- Stack Window:** Shows the stack contents starting with BIOS DI INT 021h.
- Terminal Window:** Shows the emulator screen output where the user enters "1" and "2", and the program outputs "Summation of two number is: 3".
- Message Box:** A modal dialog box titled "message" with the text "PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM" and an "OK" button.

At the bottom, the Windows taskbar shows the date and time as 11:47 AM, 7/19/2023.

```
01 .model small
02 .data
03 str1 db "Enter First Number: $"
04 str2 db 13,10,"Enter Second Number: $"
05 str3 db 13,10,"Summation of two number is: $"
06 .code
07 main proc
08     mov ax,@data
09     mov ds,ax
10    lea dx,str1
11    mov ah,9
12    int 21h
13    mov ah,1
14    int 21h
15    mov bh,al
16    lea dx,str2
17    mov ah,9
18    int 21h
19    mov ah,1
20    int 21h
21    mov bh,al
22    sub bh,48
23    sub bl,48
24    lea dx,str3
25    mov ah,9
26    int 21h
27    add bh,b1
28    add bh,48
29    mov dl,bh
30    mov ah,2
31    int 21h
32    mov ah,4Ch
33    int 21h
34 main endp
35 end
```

Screenshot: 07 — 9. Add 2 numbers.

The screenshot shows the emu8086 development environment. The assembly code window displays the following code:

```
01 DATA SEGMENT
02 X DW 04H
03 SQUARE DW ?
04 CUBE DW ?
05 DATA ENDS
06 CODE SEGMENT
07 ASSUME CS:CODE, DS:DATA
08 START: MOV AX, DATA
09 MOU DS, AX
10 MOU AX, X
11 MOU BX, X
12 MUL BX
13 MOU SQUARE, AX
14 MUL BX
15 MOU CUBE, AX
16 MOU AH, 4CH
17 INT 21H
18 CODE ENDS
19 END START
```

The code editor has a yellow selection bar under the instruction `INT 21H`. The registers window shows:

	H	L
AX	4C	40
BX	00	04
CX	00	2A
DX	00	00
CS	F400	
IP	0204	
SS	0710	
SP	FFFA	
BP	0000	
SI	0000	
DI	0000	
DS	0710	
ES	0700	

The stack window shows:

F400:0204	BIOS DI
F400:0205	INT 021h
F4204: CF 207 =	IRET
F4205: 00 000 NULL	ADD [BX + SI], AL
F4206: 00 000 NULL	ADD [BX + SI], AL
F4207: 00 000 NULL	ADD [BX + SI], AL
F4208: 00 000 NULL	ADD [BX + SI], AL
F4209: 00 000 NULL	ADD [BX + SI], AL
F420A: 00 000 NULL	ADD [BX + SI], AL
F420B: 00 000 NULL	ADD [BX + SI], AL
F420C: 00 000 NULL	ADD [BX + SI], AL
F420D: 00 000 NULL	ADD [BX + SI], AL
F420E: 00 000 NULL	ADD [BX + SI], AL
F420F: 00 000 NULL	ADD [BX + SI], AL
F4210: 00 000 NULL	ADD [BX + SI], AL
F4211: 00 000 NULL	ADD [BX + SI], AL
F4212: 00 000 NULL	ADD [BX + SI], AL
F4213: 00 000 NULL	ADD [BX + SI], AL
F4214: 00 000 NULL	ADD [BX + SI], AL
...	

A message dialog box is displayed with the text: "PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM".

At the bottom, the taskbar shows the Windows Start button, Search bar, and various pinned icons.

Screenshot: 08 – 13. Find square and cube of a number.

The screenshot shows the emu8086 interface with the following components:

- Assembler Window:** Displays assembly code for finding the LCM of two numbers (Num1=5, Num2=10). The code uses AX, BX, and DX registers to perform division, multiplication, and comparison operations.
- Registers Window:** Shows CPU register values at address F400:0204. AX contains 4C, BX contains 08, CX contains 3F, DX contains 00, CS contains F400, IP contains 0204, SS contains 0710, SP contains FFFA, BP contains 0000, SI contains 0000, DI contains 0000, DS contains 0710, and ES contains 0700. The instruction at F400:0204 is INT 21h.
- Variables Window:** Shows variable definitions: NUM1 (10), NUM2 (12), and ANS (60).
- Message Dialog:** A "message" dialog box appears with the text "PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM".
- System Taskbar:** At the bottom, it shows the Windows taskbar with icons for File Explorer, Search, and other applications, along with system status indicators like battery level and date/time (11:56 AM, 7/19/2023).

Screenshot: 09 – 14. Find the LCM of the given number.

```

01 DATA SEGMENT
02 NUM1 DW 000AH
03 NUM2 DW 0004H
04 GCD DW ?
05 DATA ENDS
06 CODE SEGMENT
07 ASSUME CS:CODE, DS:DATA
08 START: MOV AX, DATA ;Load the Data to AX.
09 MOU DS, AX ;Move the Data AX to DS.
10 MOU AX, NUM1 ;Move the first number to AX.
11 MOU BX, NUM2 ;Move the second number to BX.
12 UP: CMP AX, BX ;Compare the two numbers.
13 JE EXIT ;If equal, go to EXIT label.
14 JB EXCG ;If first number is below than second,
           ;go to EXCG label.
15 UP1: MOU DX, 0H ;Initialize the DX.
16 DIV BX ;Divide the first number by second number.
17 CMP DX, 0 ;Compare remainder is zero or not.
18 JE EXIT ;If zero, jump to EXIT label.
19 MOU AX, DX ;If non-zero, move remainder to AX.
20 JMP UP ;Jump to UP label.
21 EXCG: XCHG AX, BX ;Exchange the remainder and quotient.
22 JMP UP1 ;Jump to UP1.
23 EXIT: MOU GCD, BX ;Store the result in GCD.
24 MOU AH, 4CH
25 INT 21H
26 CODE ENDS
27 END START

```

original source c... - X

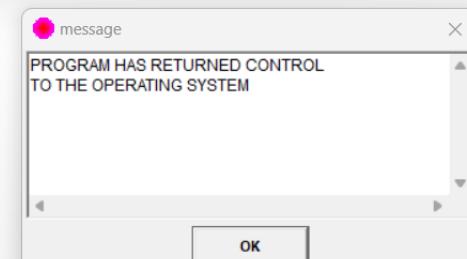
```

09 MOU DS, AX ;Move the
10 MOU AX, NUM1 ;Move the
11 MOU BX, NUM2 ;Move the
12 UP: CMP AX, BX ;Co
13 JE EXIT ;If equ
14 JB EXCG ;If fir
15 ;go to EXCG label.
16 UP1: MOU DX, 0H ;In
17 DIV BX ;Divide
18 CMP DX, 0 ;Compare
19 JE EXIT ;If zero
20 MOU AX, DX ;If non-
21 JMP UP ;Jump to
22 EXCG: XCHG AX, BX ;Exc
23 JMP UP1 ;Jump to
24 EXIT: MOU GCD, BX ;Sto
25 MOU AH, 4CH
26 INT 21H
27 CODE ENDS

```

variables	
size:	word
elements:	1
edit	show as: signed
NUM1	15
NUM2	4
GCD	I

	F400:0204	F400:0204
AX	H 4C L 03	BIOS DI
BX	H 00 L 01	INT 021h
CX	H 00 L 3B	LERET
DX	H 00 L 00	ADD IBX + SI1, AL
CS	H F400 L	ADD IBX + SI1, AL
IP	H 0204 L	ADD IBX + SI1, AL
SS	H 0710 L	ADD IBX + SI1, AL
SP	H FFFA L	ADD IBX + SI1, AL
BP	H 0000 L	ADD IBX + SI1, AL
SI	H 0000 L	ADD IBX + SI1, AL
DI	H 0000 L	ADD IBX + SI1, AL
DS	H 0710 L	ADD IBX + SI1, AL
ES	H 0700 L	...



Screenshot: 10 – 16. Find GCD of two numbers.

emu8086 - assembler and microprocessor emulator 4.08

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator converter options help

```

001 ; this example gets the number from the user,
002 ; and calculates factorial for it.
003 ; supported input from 0 to 8 inclusive!
004
005 name "fact"
006
007 ; this macro prints a char in AL and advances
008 ; the current cursor position:
009 putc macro char
010   push ax
011   mov al, char
012   mov ah, 0eh
013   int 10h
014   pop ax
015 endm
016
017 org 100h
018
019 jmp start
020
021 result dw ?
022
023
024 start:
025
026
027
028
029 ; get first number:
030 mov dx, offset msg1
031 mov ah, 9
032 int 21h
033 jmp ni
034 msg1 db 0Dh,0Ah, 'enter the number: $'
035 ni:
036
037 call scan_num
038
039
040 ; factorial of 0 = 1:
041 mov ax, 1
042 cmp cx, 0
043 je print_result
044
045 ; move the number to bx:
046 ; cx will be a counter:
047 mov bx, cx
048
049 mov ax, 1
050 mov bx, 1
051
052 calc_it:
053 mul bx
054 cmp dx, 0
055 jne overflow
056
057 overflow:
058 mov ah, 9
059 int 21h
060 jmp start
061
062 exit:
063
064 ; wait for any key press
065 mov ah, 0
066 int 16h
067
068 ret

```

original source c... file math debug view external virtual devices virtual drive help

Load Reload step back single step run step delay ms: 0

registers		F400:0154	F400:0154
AX	1C 0D	F4150: FF 255 RES	BIOS DI
BX	00 09	F4151: FF 255 RES	INT 020h
CX	00 00	F4152: CD 205 =	IRET
DX	01 4B	F4153: 20 032 SPA	ADD [BX + SI], AL
CS	F400	F4154: CF 207 =	ADD [BX + SI], AL
IP	0154	F4155: 00 000 NULL	ADD [BX + SI], AL
SS	0700	F4156: 00 000 NULL	ADD [BX + SI], AL
SP	FFFA	F4157: 00 000 NULL	ADD [BX + SI], AL
BP	0000	F4158: 00 000 NULL	ADD BH, BH
SI	0000	F4159: 00 000 NULL	DEC BP
		F415A: 00 000 NULL	SBB CL, BH
		F415B: 00 000 NULL	ADD [BX + SI], AL
		F415C: 00 000 NULL	ADD [BX + SI], AL
		F415D: 00 000 NULL	...

screen source reset aux vars debug stack flags

emulator: fact.com_ file math debug view external virtual devices virtual drive help

50 emulator screen (80x25 chars)

enter the number: 8

factorial: 40320

clear screen change font 0/16

line: 329 col: 90 drag a file here to open

Search

11:41 AM 7/19/2023

message

PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM

OK

Screenshot: 11 – 17. Find the factorial of a given number.

emu8086 - assembler and microprocessor emulator 4.08

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator converter options help about

```

01 DATA SEGMENT
02 X DB 0AAH
03 ONE DB ?
04 ZERO DB ?
05 DATA ENDS
06 CODE SEGMENT
07 ASSUME CS:CODE,DS:DATA
08 START: MOU AX,DATA
09 MOU DS,AX
10 MOU AH,X
11 MOU BL,8
12 MOU CL,1
13 UP: ROR AH,CL
14 JNC DOWN
15 INC ONE
16 JMP DOWN1
17 INC ZERO
18 DOWN1: DEC BL
19 JNZ UP
20 MOU AH,4CH
21 INT 21H
22 CODE ENDS
23 END START
24

```

original source.c..

```

04 ZERO DB ?
05 DATA ENDS
06 CODE SEGMENT
07 ASSUME CS:CODE,DS:DATA
08 START: MOU AX,DATA
09 MOU DS,AX
10 MOU AH,X
11 MOU BL,8
12 MOU CL,1
13 UP: ROR AH,CL
14 JNC DOWN
15 INC ONE
16 JMP DOWN1
17 DOWN: INC ZERO
18 DOWN1: DEC BL
19 JNZ UP
20 MOU AH,4CH
21 INT 21H
22 CODE ENDS
23 END START
24

```

registers

	H	L
AX	4C	10
BX	00	00
CX	00	01
DX	00	00
CS	F400	
IP	0204	
SS	0710	
SP	FFFA	
BP	0000	
SI	0000	
DI	0000	
DS	0710	
ES	0700	

F400:0204 F400:0204

```

F4200: FF 255 RES
F4201: FF 255 RES
F4202: CD 205 =
F4203: 21 033 !
F4204: CF 207 =
F4205: 00 000 NULL
F4206: 00 000 NULL
F4207: 00 000 NULL
F4208: 00 000 NULL
F4209: 00 000 NULL
F420A: 00 000 NULL
F420B: 00 000 NULL
F420C: 00 000 NULL
F420D: 00 000 NULL
F420E: 00 000 NULL
F420F: 00 000 NULL
F4210: 00 000 NULL
F4211: 00 000 NULL
F4212: 00 000 NULL
F4213: 00 000 NULL
F4214: 00 000 NULL

```

BIOS DI
INT 021h
IRET

screen source reset aux vars debug stack flags

variables

size: byte elements: 1

X 10001010b
ONE 3
ZERO 5

message

PROGRAM HAS RETURNED CONTROL
TO THE OPERATING SYSTEM

OK

line: 24 col: 1 drag a file here to open

Search

12:29 PM 7/19/2023

Screenshot: 12 – 22. Logical ones and zeroes in given data.

 emu8086 - assembler and microprocessor emulator

file edit bookmarks assembler emulator math ascii codes help



1



```

01 DATA SEGMENT
02 NUM DB 8H
03 MES1 DB 10,13,'DATA IS POSITIVE $'
04 MES2 DB 10,13,'DATA IS NEGATIVE $'
05 DATA ENDS
06 CODE SEGMENT
07 ASSUME CS:CODE, DS:DATA
08 START: MOU AX, DATA
09 MOU DS, AX
10 MOU AL, NUM
11 ROL AL,1
12 JC NEGA
13 ;Move the Number to AL.
14 ;Perform the rotate left side for 1 bit position
15 ;Check for the negative number.
16 MOU DX, OFFSET MES1;Declare it positive.
17 JMP EXIT ;Exit program.
18 NEGA: MOU DX, OFFSET MES2;Declare it negative.
19 EXIT: MOU AH,09H
20 INT 21H
21 MOU AH,4CH
22 INT 21H
23 CODE ENDS
24 END START

```

```
original source c... ━ ━ ━
05 DATA ENDS
06 CODE SEGMENT
07 ASSUME CS:CODE, DS:DATA
08 START: MOU AX, DATA
09 MOU DS, AX
10 MOU AL, NUM
11 ROL AL, 1
12 JC NEGA
13 ;Move the Number to AL
14 ;Perform the rotate left
15 ;Check for the negative
16 MOU DX, OFFSET MES1 ;Delete
17 JMP EXIT ;Exit program
18 NEGA: MOU DX, OFFSET MES
19 EXIT: MOU AH, 09H
20 INT 21H
21 MOU AH, 4CH
22 INT 21H
23 CODE ENDC
```

SCR emulator screen (80x25 chars)

DATA IS POSITION

line: 8 col: 49

drag a file here to open

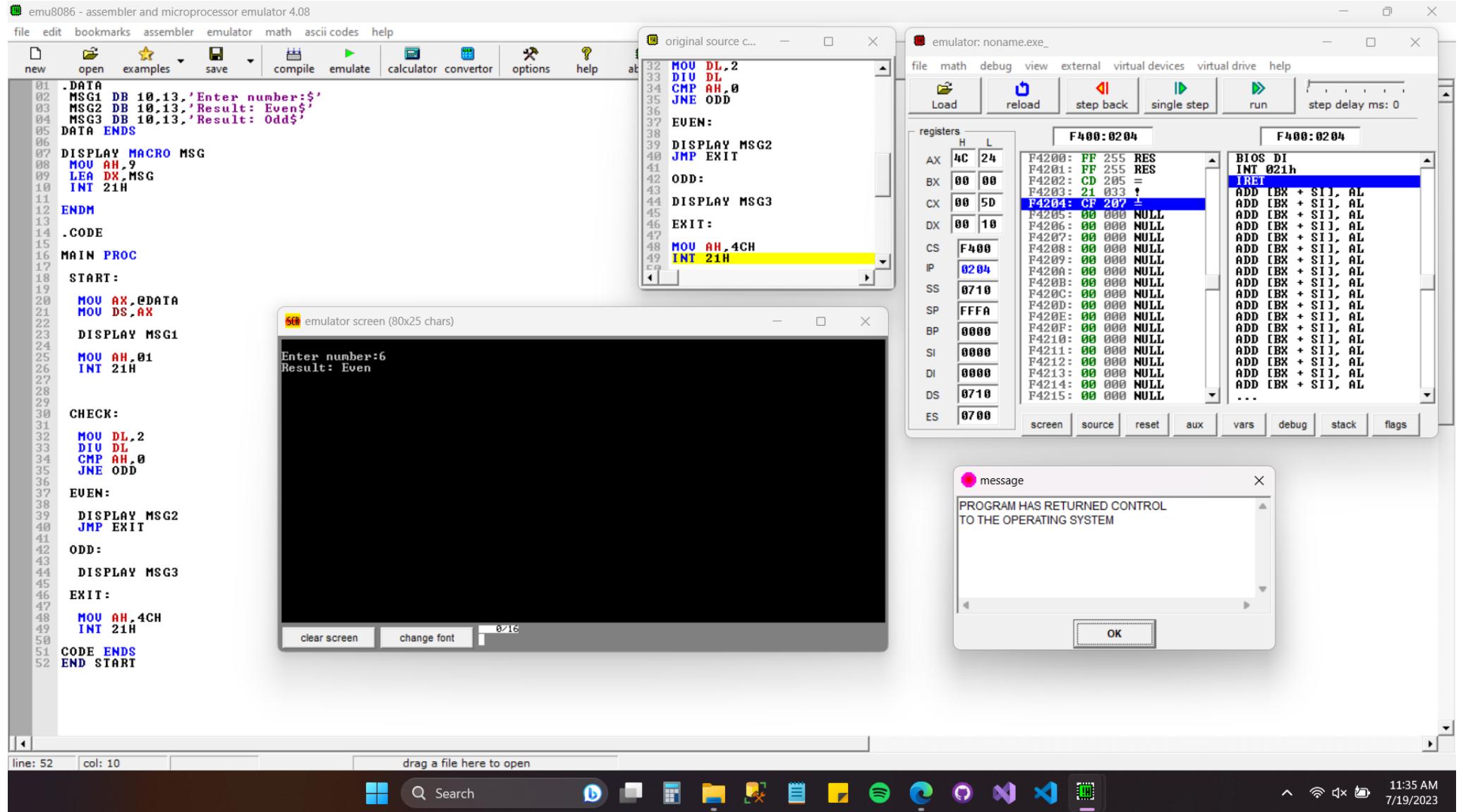
0/1

clear screen

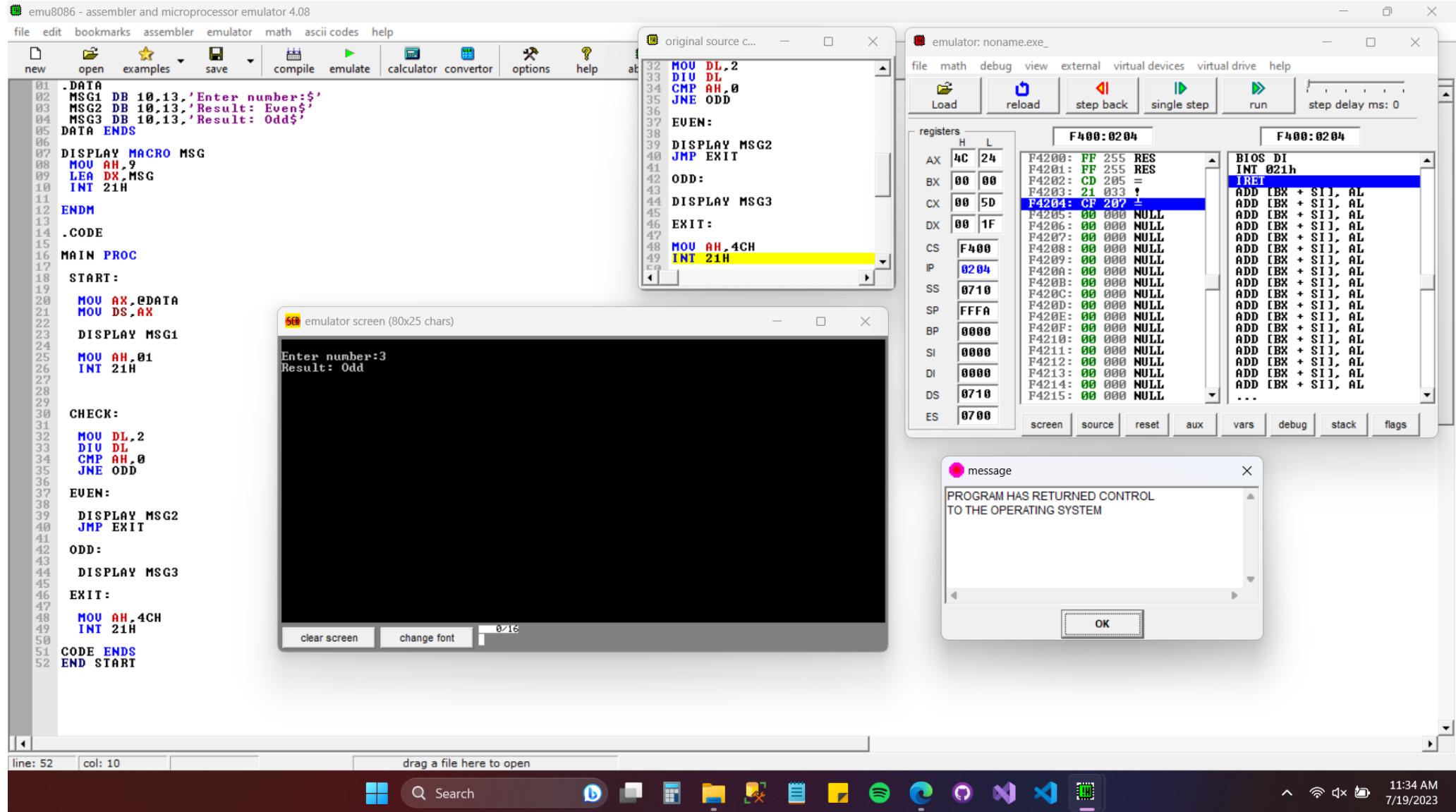
[change font](#)

The screenshot shows a debugger interface with the title bar "emulator: noname.exe_". The menu bar includes file, math, debug, view, external, virtual devices, virtual drive, and help. Below the menu are several control buttons: Load, reload, step back, single step, run, and step delay ms: 0. A registers window displays memory addresses F4000 to F420F with their corresponding values. The CPU pane shows assembly code starting with F4200: FF 255 RES, followed by a series of ADD [BX + SI], AL instructions. The stack pane shows the instruction F4204: CF 207 ÷ highlighted. A message dialog box in the foreground says "PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM" with an OK button.

Screenshot: 13 – 23. Find the data is positive or negative.



Screenshot: 14 — 24. Find the data is odd or even (even).



Screenshot: 15 – 24. Find the data is odd or even (odd).

The screenshot shows the emu8086 development environment with several windows open:

- Assembler Window:** Displays assembly code for a program named "noname.exe". The code includes segments for DATA and CODE, and handles BCD to binary conversion.
- Registers Window:** Shows the CPU register state at address F400:0204. AX is 4C, BX is 05, CX is 0A, DX is 00, CS is F400, IP is 0204, SS is 0710, SP is FFFA, BP is 0000, SI is 0000, DI is 0000, DS is 0710, and ES is 0700. The instruction at F400:0204 is INT 21H.
- Memory Dump Window:** Shows memory starting at address F400:0204. The value at F400:0204 is FF 255, which corresponds to the INT 21H instruction.
- Variables Window:** Shows a variable named "BCD" of size word with elements 1. It displays the value 53 in BCD format and its binary representation 0000000111100101b.
- Message Dialog:** A modal dialog titled "message" displays the text "PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM" and has an "OK" button.

The system tray at the bottom right shows the date and time as 7/19/2023, 12:32 PM.

Screenshot: 16 – 27. BCD to Binary code conversion.

emu8086 - assembler and microprocessor emulator 4.08

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator converter options help about

```

01 DATA SEGMENT
02 NUM1 DW 0FFFFH,0FFFFH
03 NUM2 DW 1111H,1111H
04 SUM DW 4 DUP(0)
05 DATA ENDS
06 CODE SEGMENT
07 ASSUME CS:CODE, DS:DATA
START: MOV AX, DATA
08 MOU DS, AX
09 MOU AX, NUM1
10 ADD AX, NUM2
11 MOU SUM, AX
12 MOU AX, NUM1+2
13 MOU AX, NUM2+2
14 ADC AX, NUM2+2
; Move LSB of NUM1 to AX
15 ; Add LSB of NUM2 to AX
16 ; Store the LSB in SUM
17 ; Move MSB of NUM1 to AX
18 ; Move MSB of NUM2 to AX
19 ; Add MSB of NUM2 to AX
20 JNC DOWN
21 MOU SUM+4, 01H
DOWN: MOU SUM+2, AX
22 MOU AH, 4CH
23 INT 21H
24
25 CODE ENDS
END START
26
27

```

original source c... 0710:0008 0710:0008

```

07 ASSUME CS:CODE, DS:DATA
08 START: MOU AX, DATA
09 MOU DS, AX
10 MOU AX, NUM1
11 ADD AX, NUM2
12 MOU SUM, AX
13 MOU AX, NUM1+2
14 ADC AX, NUM2+2
; Move LSB of NUM1 to AX
15 ; Add LSB of NUM2 to AX
16 ; Store the LSB in SUM
17 ; Move MSB of NUM1 to AX
18 ; Move MSB of NUM2 to AX
19 ; Add MSB of NUM2 to AX
20 JNC DOWN
21 MOU SUM+4, 01H
DOWN: MOU SUM+2, AX
22 MOU AH, 4CH
23 INT 21H
24
25 CODE ENDS
END START
26
27

```

; Check for carry

variables screen source reset aux vars debug stack flags

	H	L
AX	4C	10
BX	00	00
CX	00	35
DX	00	00
CS	F400	
IP	0204	
SS	0710	
SP	FFFA	
BP	0000	
SI	0000	
DI	0000	
DS	0710	
ES	0700	

0710:0008 0710:0008

```

07100: 0F 015 * NULL
07101: 00 000 NULL
07102: FF 255 RES
07104: 10 016 ▶
07105: 00 000 NULL
07106: 11 017 ▲
07107: 11 017 ▲
07108: 1F 031 ▼
07109: 00 000 NULL
0710A: 10 016 ▶
0710B: 11 017 ▲
0710C: 01 001 @
0710D: 00 000 NULL
0710E: 00 000 NULL
0710F: 00 000 NULL
07110: B8 184 ↴
07111: 10 016 ▶
07112: 07 007 BEEP
07113: 8E 142 ↴
07114: D8 216 ↴

```

message OK

PROGRAM HAS RETURNED CONTROL TO THE OPERATING SYSTEM

line: 27 col: 1 drag a file here to open 12:11 PM 7/19/2023

Screenshot: 17 – 29. 32-bit Addition.