MICROPROCESSOR BASED SYSTEM DESIGN (UCS617)

Lab Activity Based Report

Submitted by:

Siddhi Soni	102203431
Mili Kumari	102203182
Manveen Kaur Maan	102203217
Aryan	102203328
Aditya Srivastava	102203258

Group: 3C63
BE Third Year

Submitted to: DR. MANJU KHURANA

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY

PATIALA, PUNJAB (INDIA)

JAN – JUNE 2025

Table of Contents

Program NO: 1

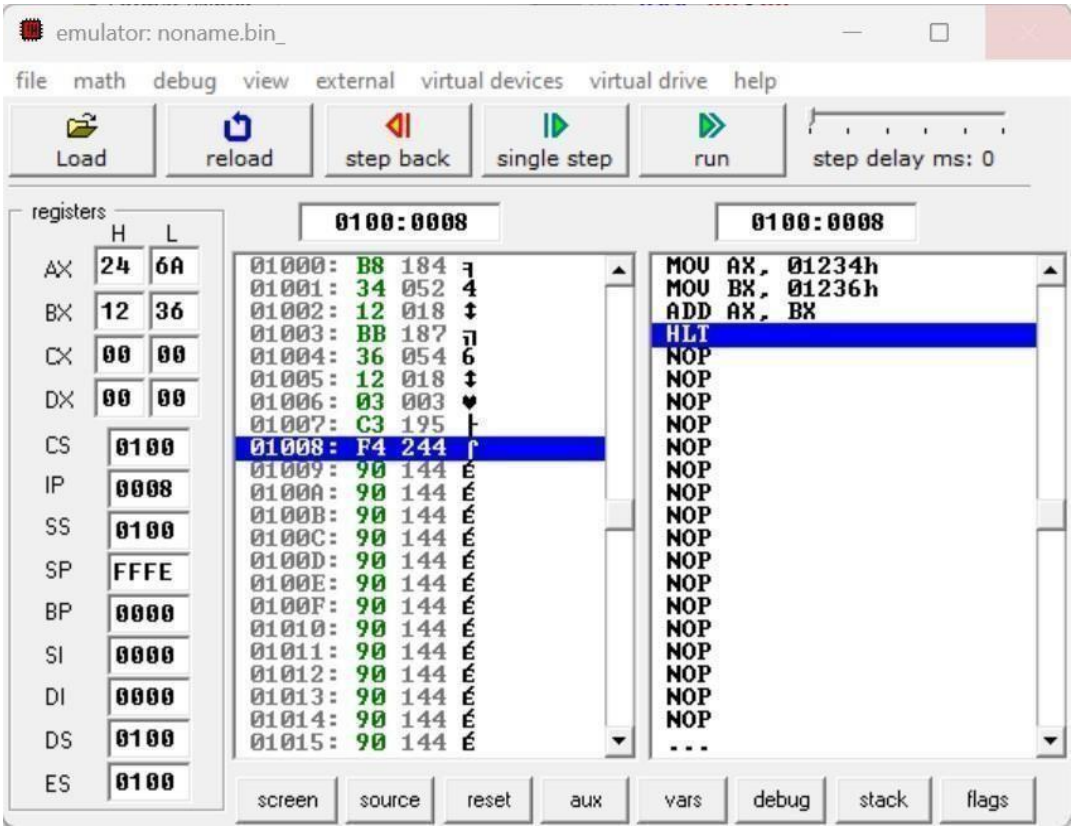
OBJECTIVE:

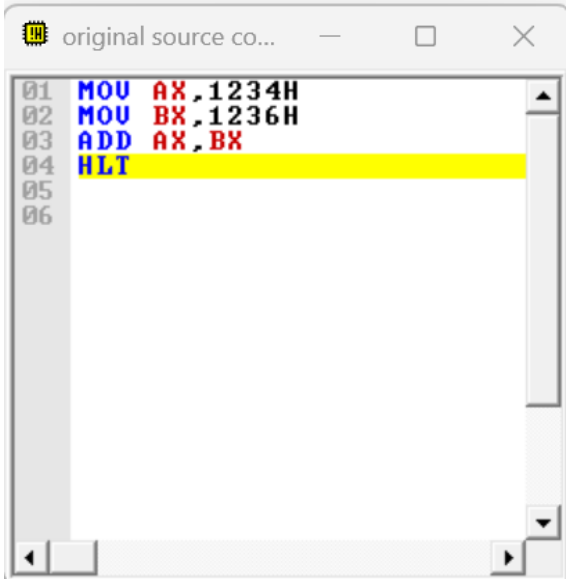
Write an assembly language program to add two 16-bit numbers in 8086.

CODE:

```
MOV
AX,1234H
MOV BX,1236H
ADD AX,BX
HLT
```

Output:




original source co...
01 MOV AX,1234H
02 MOV BX,1236H
03 ADD AX,BX
04 HLT
05
06

Program NO: 2

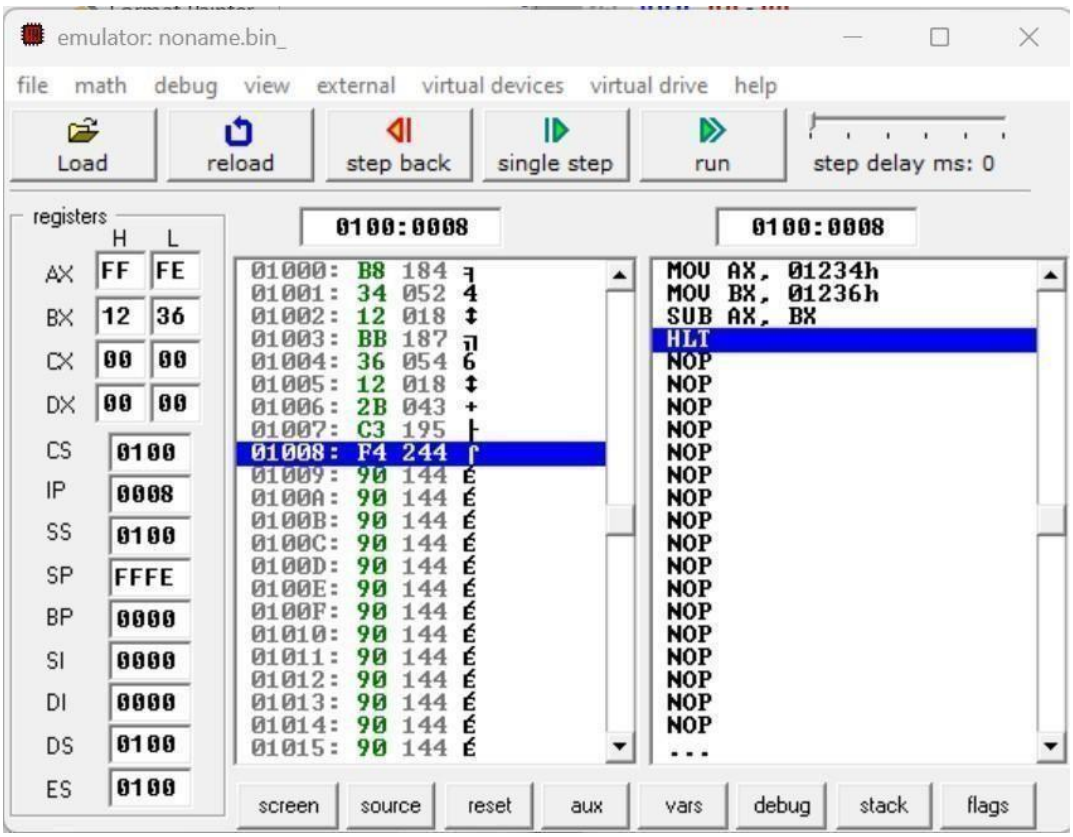
OBJECTIVE:

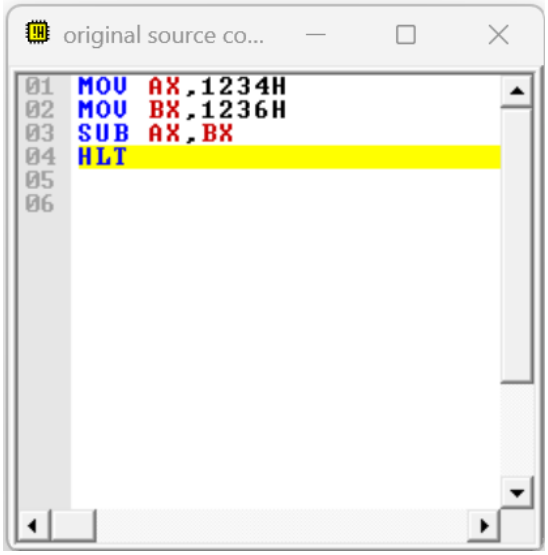
Write an assembly language program to subtract two 16-bit numbers in 8086.

CODE:

```
MOV
AX,1234H
MOV BX,1236H
SUB AX,BX
HLT
```

Output:




original source co...
01 MOV AX,1234H
02 MOV BX,1236H
03 SUB AX,BX
04 HLT
05
06

Program NO: 3

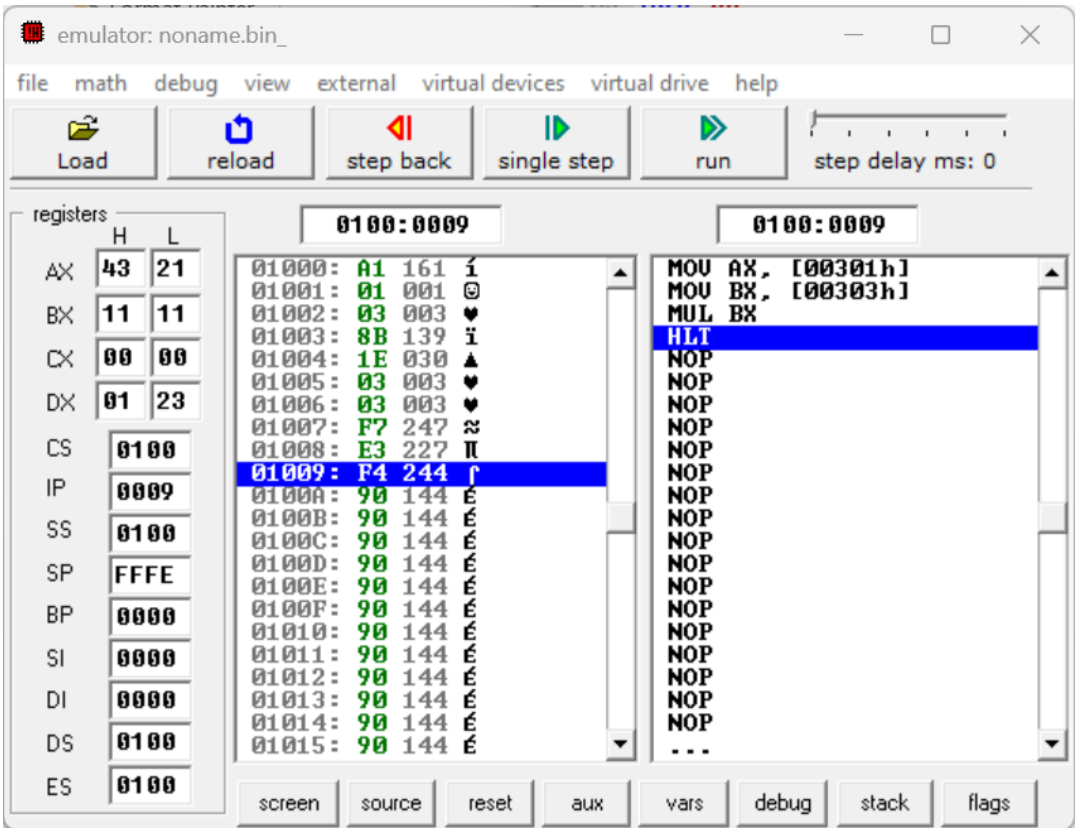
OBJECTIVE:

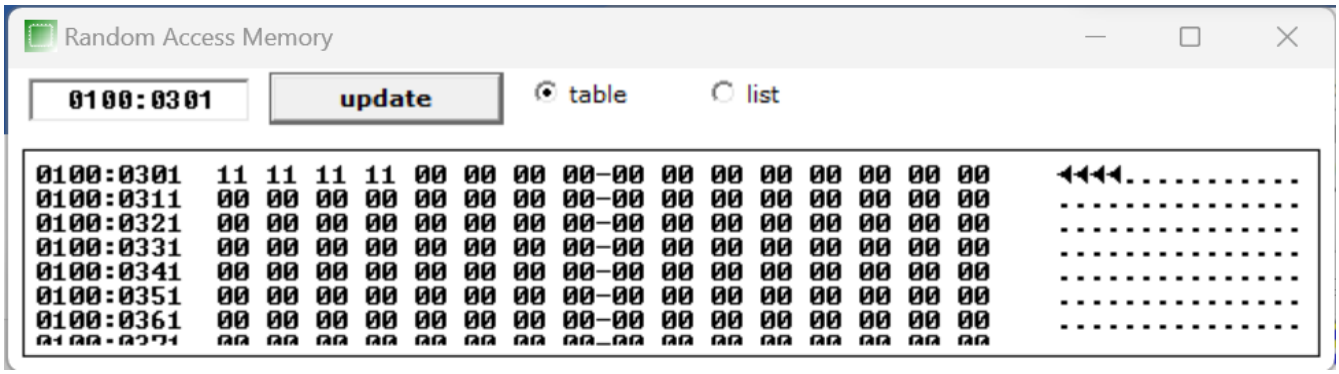
Write an assembly language program to multiply two 16-bit numbers in 8086.

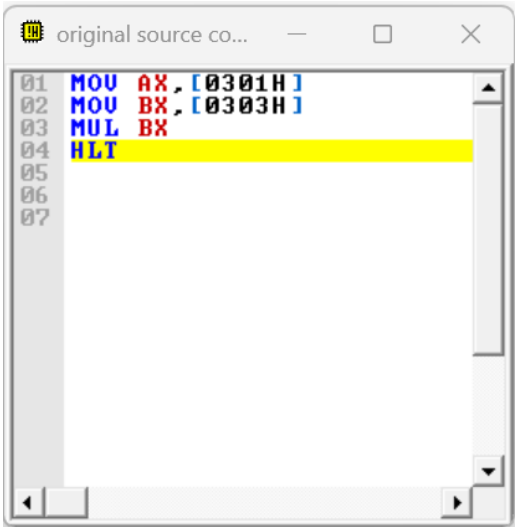
CODE:

```
MOV AX,[0301H]
MOV BX,[0303H]
MUL BX
HLT
```

Output:




Random Access Memory
0100:0301
update
table
list
0100:0301 11 11 11 11 00 00 00 00-00 00 00 00 00 00 00 00
0100:0311 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0100:0321 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0100:0331 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0100:0341 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0100:0351 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0100:0361 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00


original source co...
01 MOV AX,[0301H]
02 MOV BX,[0303H]
03 MUL BX
04 HLT
05
06
07

Program NO: 4

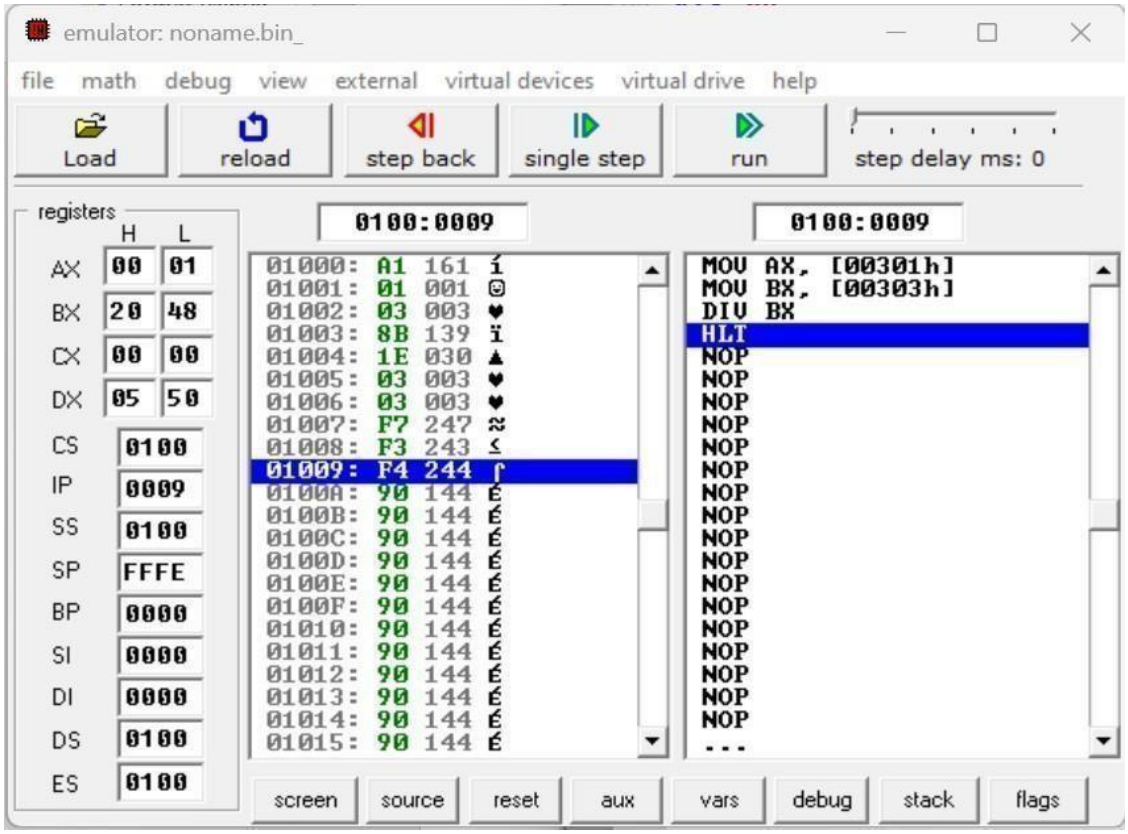
OBJECTIVE:

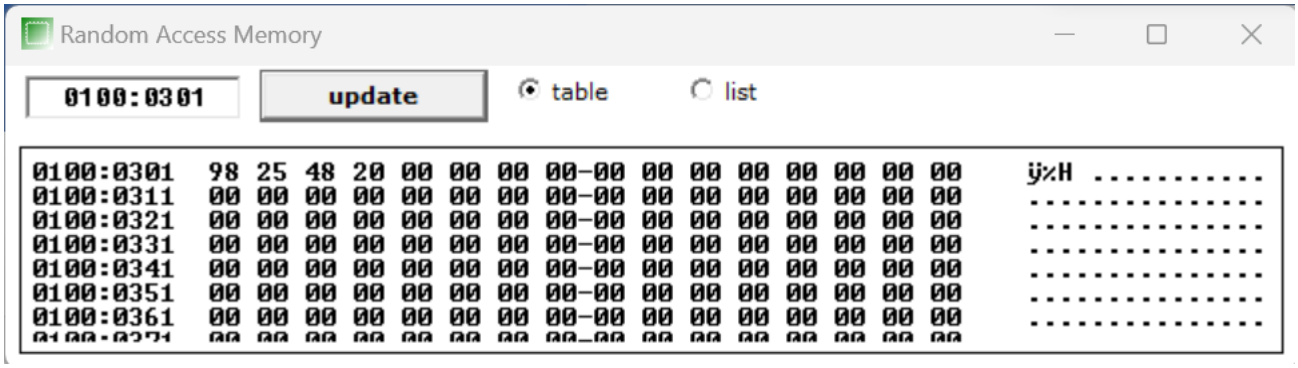
Write an assembly language program to divide two 16-bit numbers in 8086.

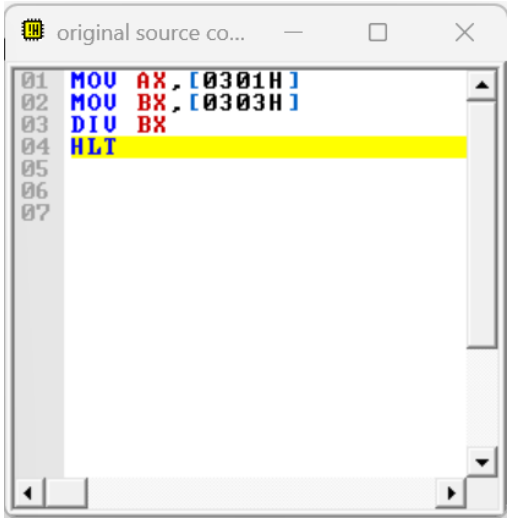
CODE:

```
MOV AX,[0301H]
MOV BX,[0303H]
DIV BX
HLT
```

Output:




Random Access Memory
0100:0301
update
table
list
0100:0301 98 25 48 20 00 00 00 00-00 00 00 00 00 00 00 00 ÿ%H .....
0100:0311 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0100:0321 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0100:0331 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0100:0341 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0100:0351 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0100:0361 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....


original source co...
01 MOV AX,[0301H]
02 MOV BX,[0303H]
03 DIV BX
04 HLT
05
06
07

Program NO: 5

OBJECTIVE:

Write an assembly language program to demonstrate AAA, AAS, AAM, AAD, DAA and DAS in 8086.

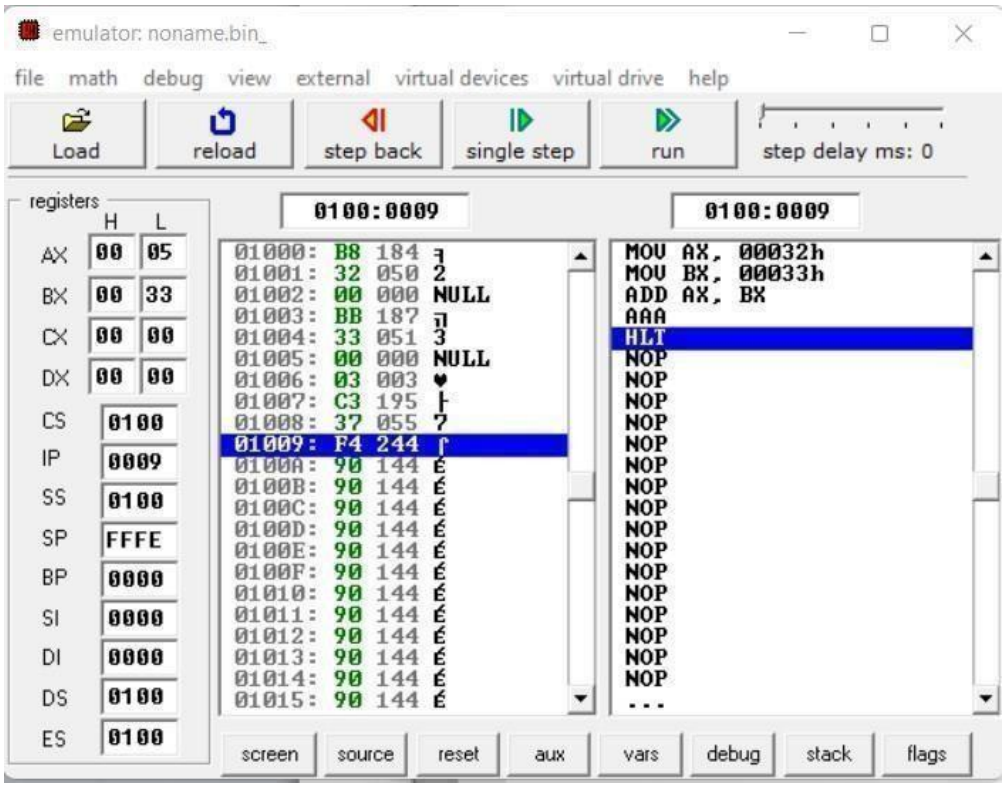
CODE:

AAA	**AAS**
MOV AX,0032H	MOV AL,33H
MOV BX,0033H	SUB AL,39H
ADD AX,BX	AAS
AAA	OR AL,30H
HLT	HLT

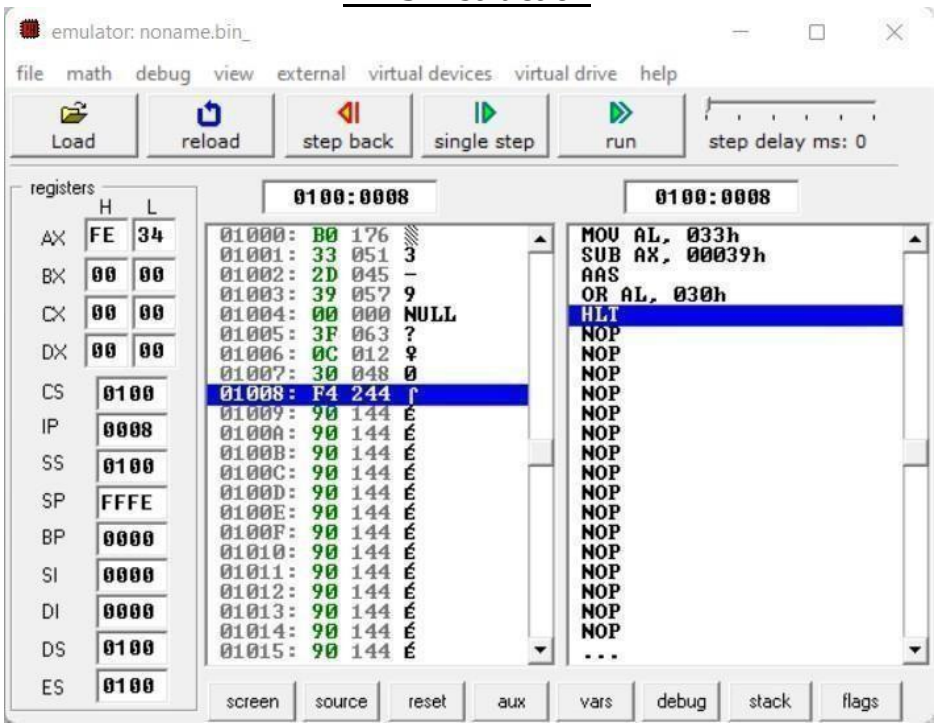
AAM	**AAD**
MOV AL,03H	MOV AX,0033H
MOV BL,09H	MOV BX,0032H
MUL BL	AAD
AAM	DIV BX
OR AX,3030H	HLT
HLT	

DAA	**DAS**
MOV AL,71H	MOV AL,71H
ADD AL,43H	SUB AL,43H
DAA	DAS
HLT	HLT

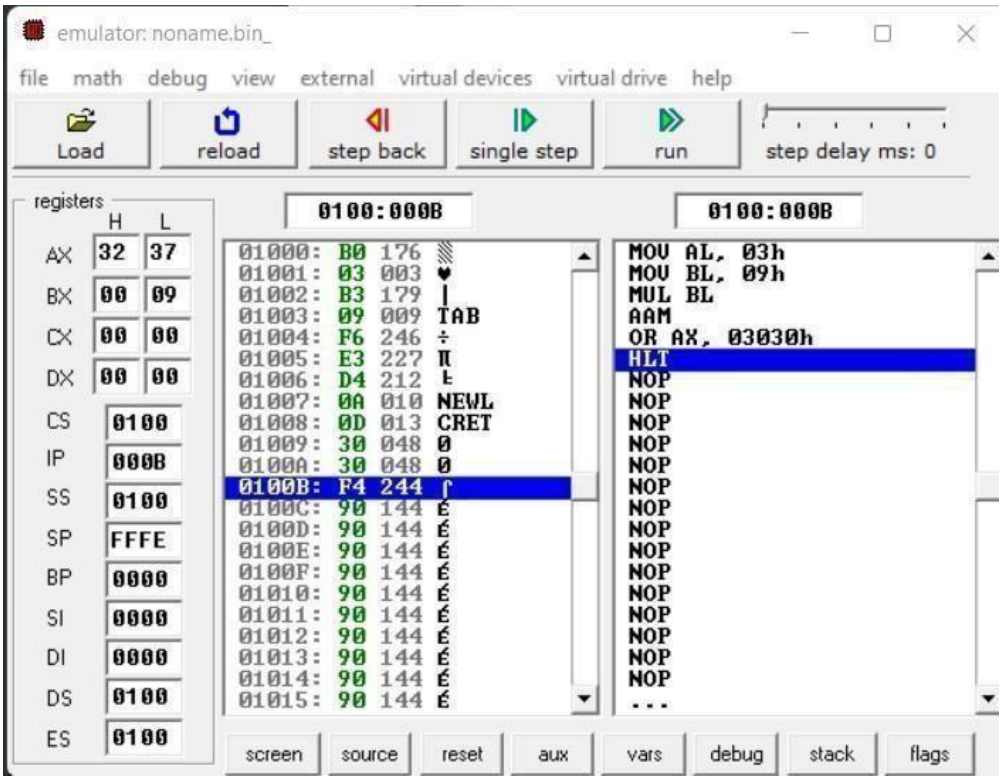
AAA Instruction



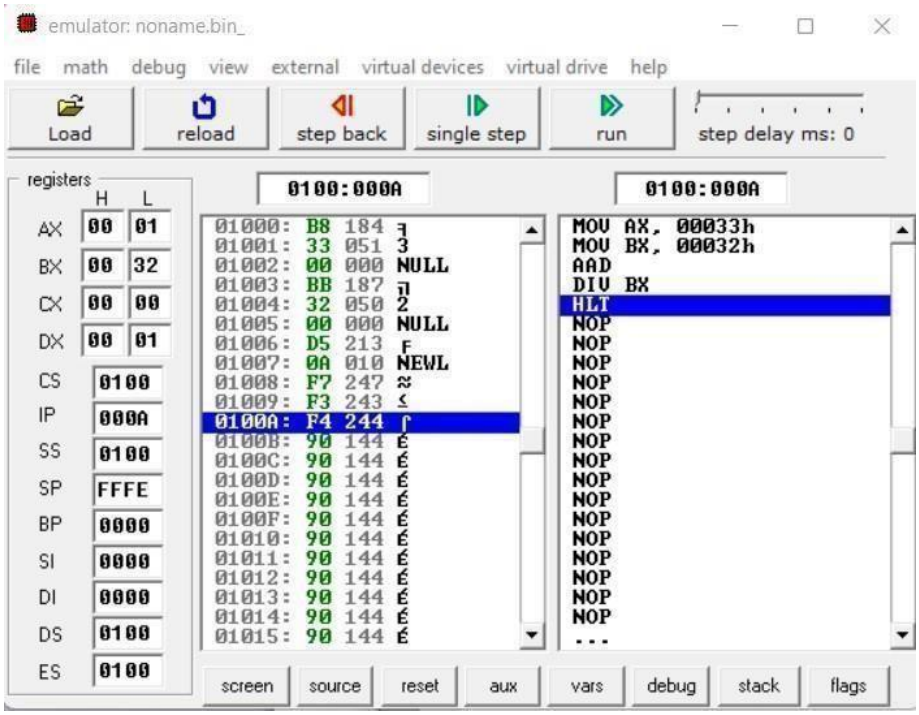
AAS Instruction



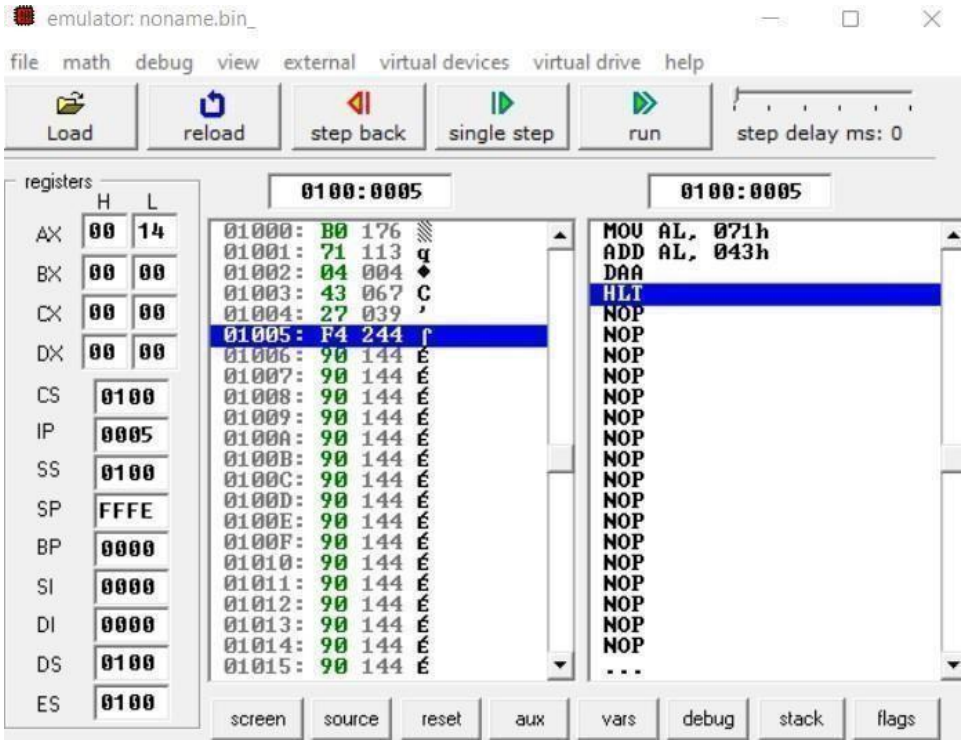
AAM Instruction



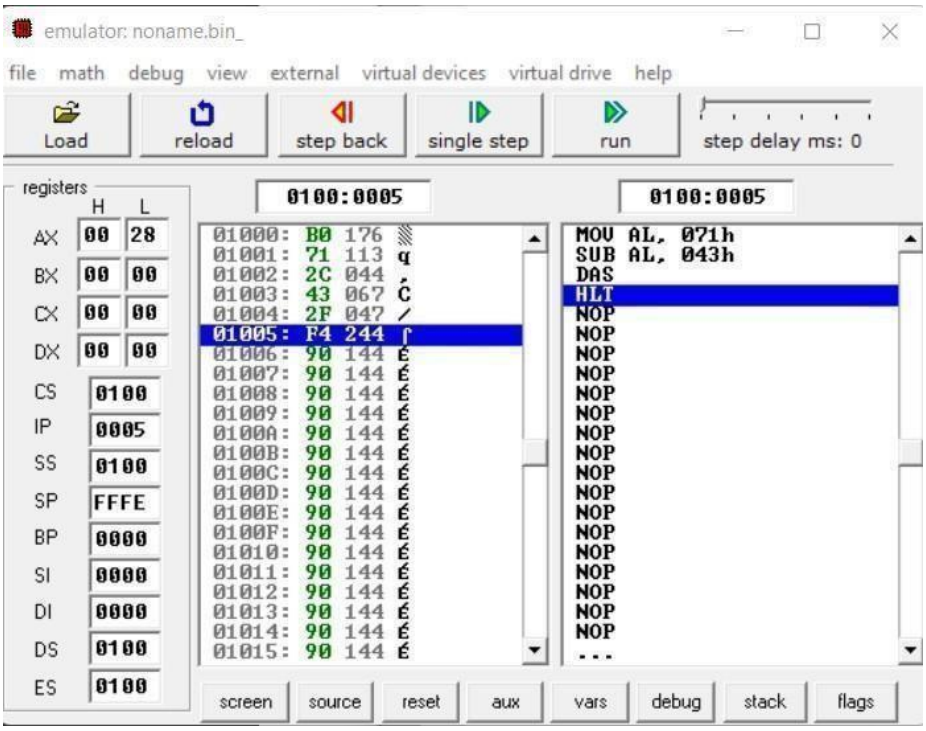
AAD Instruction



DAA Instruction



DAS Instruction



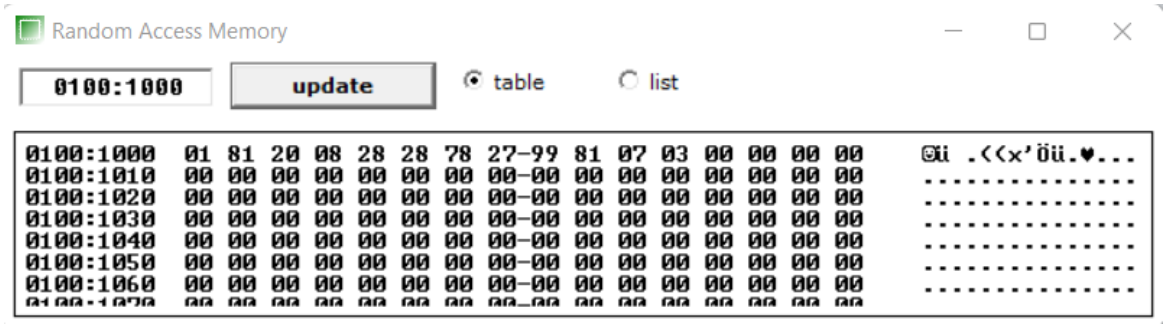
Program NO: 6

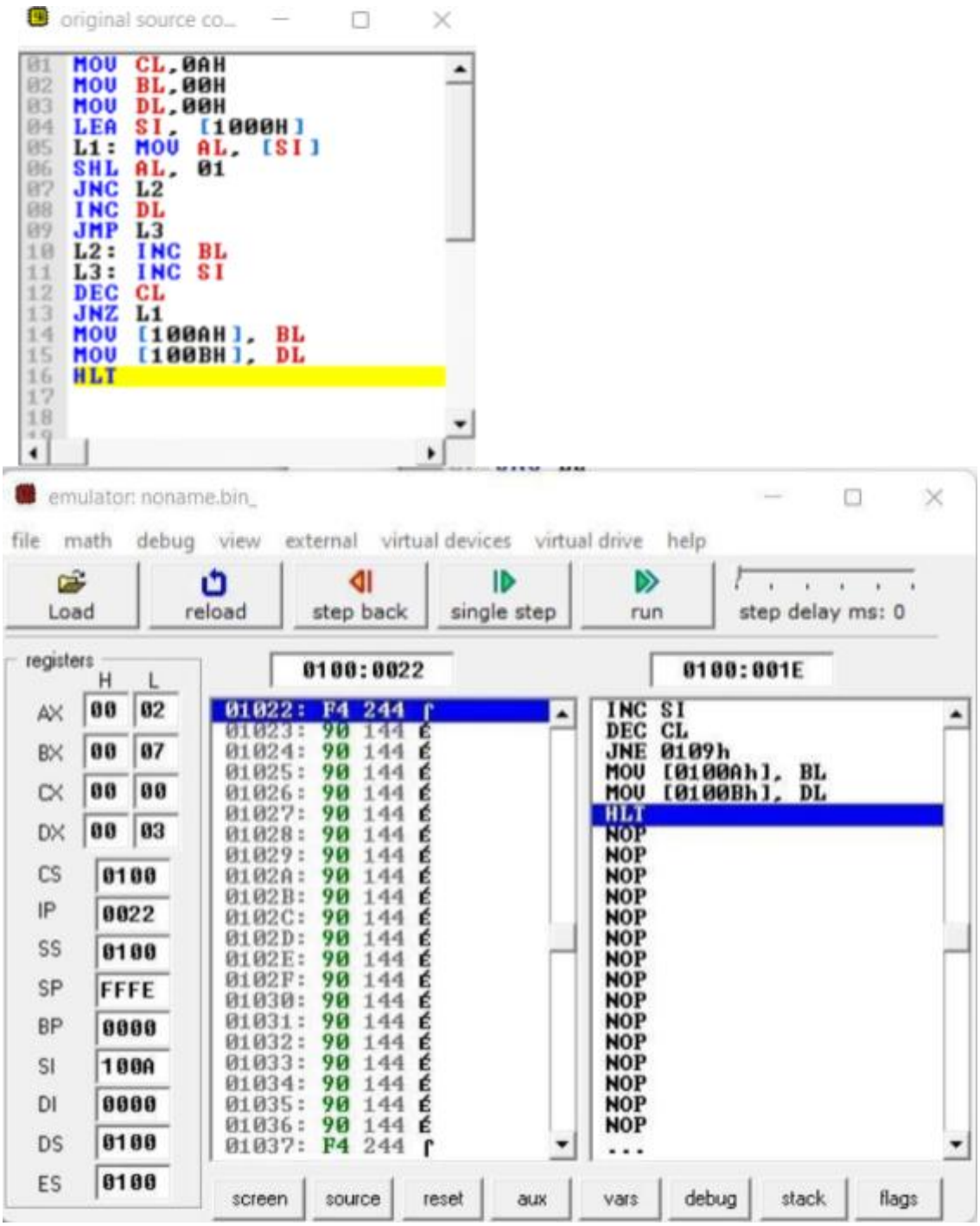
OBJECTIVE:

Write an assembly language program to find out the count of positive numbers and negative numbers from a series of signed numbers in 8086.

CODE:

```
MOV CL,0AH
MOV BL,00H
MOV DL,00H
LEA SI, [1000H]
L1: MOV AL, [SI]
SHL AL, 01
JNC  L2
INC DL
JMP  L3
L2:
INC BL
L3:
INC SI
DEC CL
JNZ L1
MOV [100AH], BL
MOV [100BH], DL
HLT
```




original source co...
MOV CL,0AH
MOV BL,00H
MOV DL,00H
LEA SI, [1000H]
L1: MOV AL, [SI]
SHL AL, 01
JNC L2
INC DL
JMP L3
L2: INC BL
L3: INC SI
DEC CL
JNZ L1
MOV [100AH], BL
MOV [100BH], DL
HLT
emulator: noname.bin
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
registers
H L
AX 00 02
BX 00 07
CX 00 00
DX 00 03
CS 0100
IP 0022
SS 0100
SP FFFE
BP 0000
SI 100A
DI 0000
DS 0100
ES 0100
0100:0022
01022: F4 244
0100:001E
INC SI
DEC CL
JNE 0109h
MOV [0100Ah], BL
MOV [0100Bh], DL
HLT
NOP
screen source reset aux vars debug stack flags

Program NO: 7

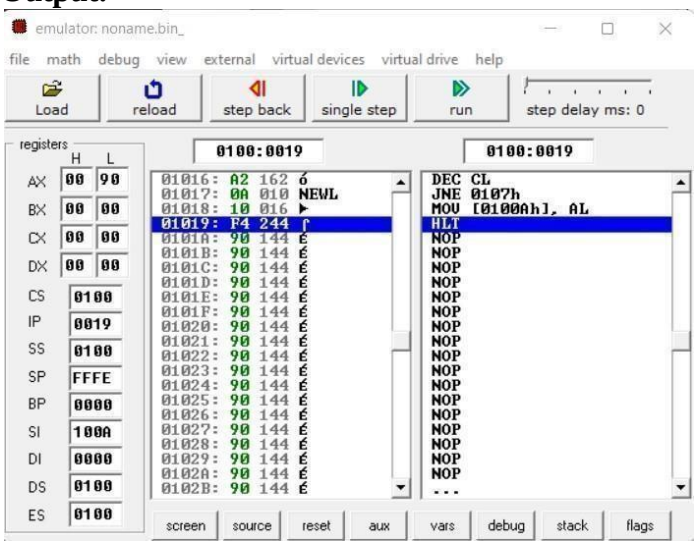
OBJECTIVE:

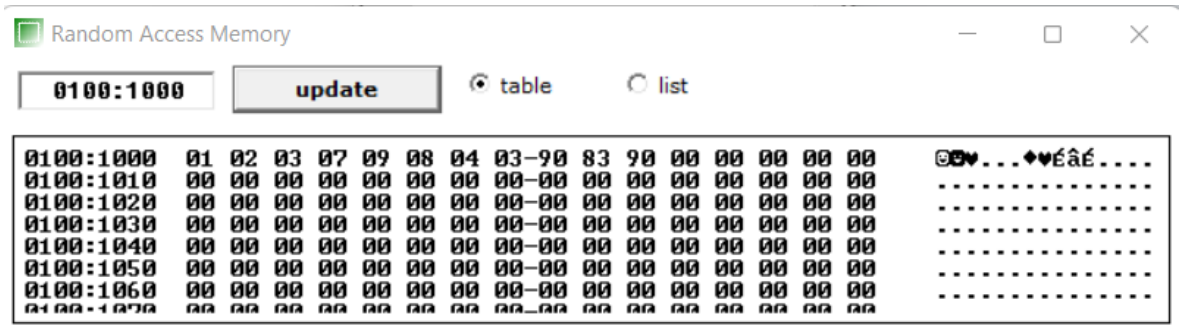
Write an assembly language program to convert to find out the largest number from a given unordered array of 8-bit numbers, stored in the locations starting from a known address in 8086.

CODE:

```
MOV CL, 0AH
LEA SI,
[1000H]
MOV AL, [SI]
L1: INC SI
MOV BL, [SI]
CMP AL, BL
JC L2 JMP L3
L2: MOV AL,
BL L3: DEC CL
JNZ L1
MOV [100AH], AL
HLT
```

Output:-




Random Access Memory
0100:1000
update
table
list
0100:1000 01 02 03 07 09 08 04 03-90 83 90 00 00 00 00 00
0100:1010 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0100:1020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0100:1030 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0100:1040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0100:1050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0100:1060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00

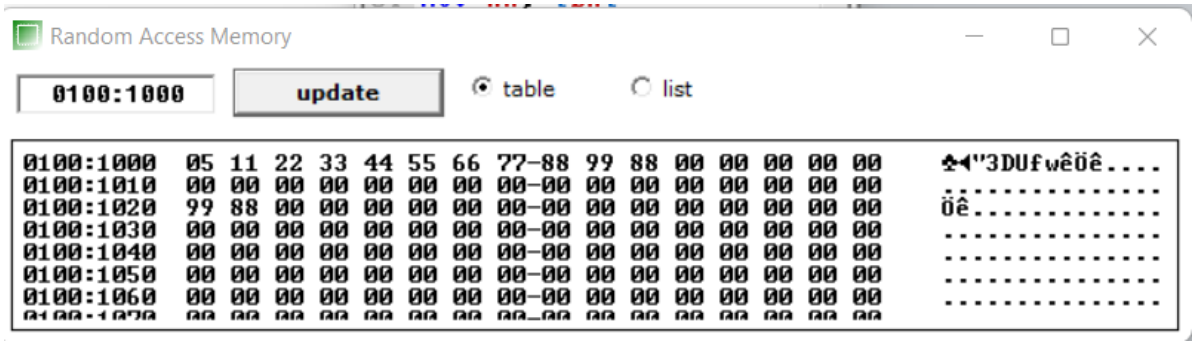
Program NO: 8

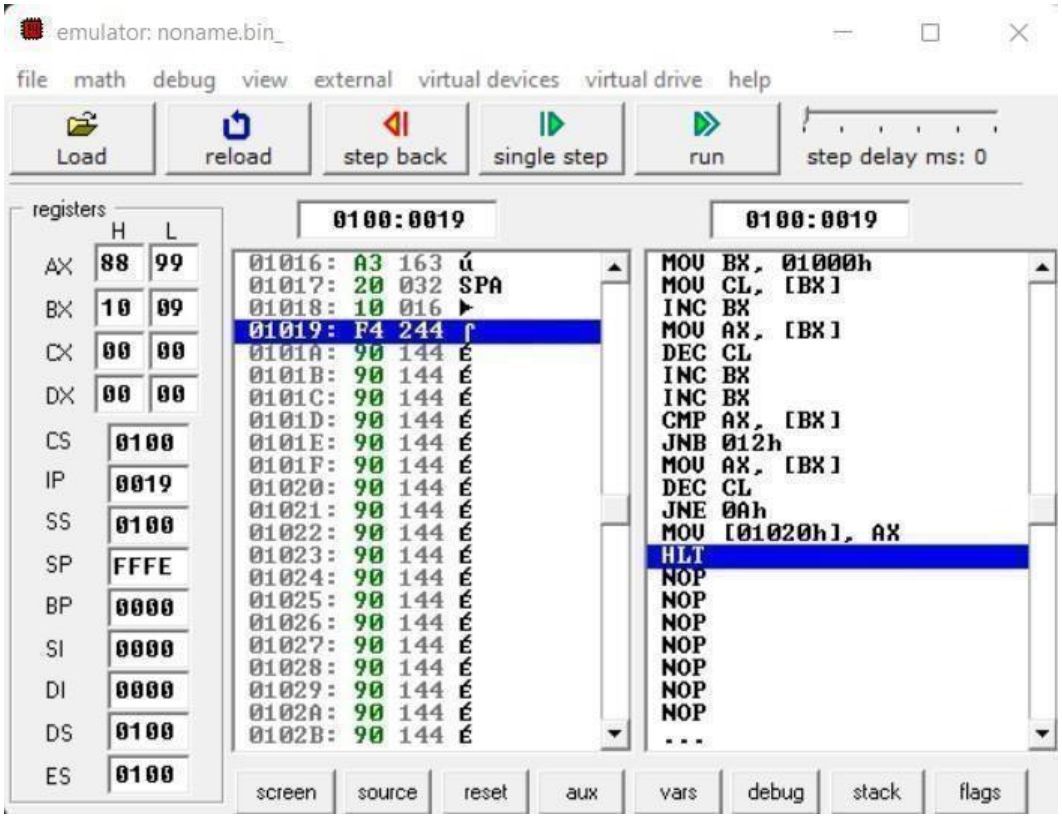
OBJECTIVE:

Write an assembly language program to find out the largest number from a given unordered array of 16-bit numbers, stored in the locations starting from a known address in 8086.

CODE:

```
MOV BX,
1000H
MOV C L, [BX]
INC BX
MOV AX,
[BX[ DEC CL
Back:      INC
BX INC BX
CMP AX, [BX]
JNC Next
MOV AX,
[BX] Next:
DEC CL JNZ
Back
MOV [1020H], AX
HLT
```




emulator: noname.bin_
file math debug view external virtual devices virtual drive help
Load
reload
step back
single step
run
step delay ms: 0
registers
H L
AX 88 99
BX 10 09
CX 00 00
DX 00 00
CS 0100
IP 0019
SS 0100
SP FFFE
BP 0000
SI 0000
DI 0000
DS 0100
ES 0100
0100:0019
01016: A3 163 ú
01017: 20 032 SPA
01018: 10 016 ►
01019: F4 244 ⌠
0101A: 90 144 É
0101B: 90 144 É
0101C: 90 144 É
0101D: 90 144 É
0101E: 90 144 É
0101F: 90 144 É
01020: 90 144 É
01021: 90 144 É
01022: 90 144 É
01023: 90 144 É
01024: 90 144 É
01025: 90 144 É
01026: 90 144 É
01027: 90 144 É
01028: 90 144 É
01029: 90 144 É
0102A: 90 144 É
0102B: 90 144 É
0100:0019
MOV BX, 01000h
MOV CL, [BX]
INC BX
MOV AX, [BX]
DEC CL
INC BX
INC BX
CMP AX, [BX]
JNB 012h
MOV AX, [BX]
DEC CL
JNE 0Ah
MOV [01020h], AX
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
screen
source
reset
aux
vars
debug
stack
flags

Program NO: 9

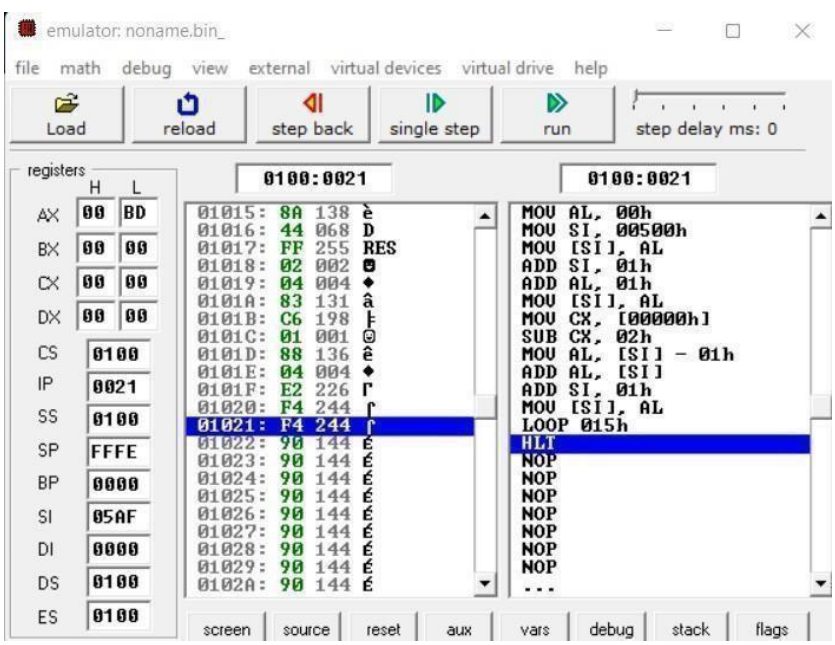
OBJECTIVE:

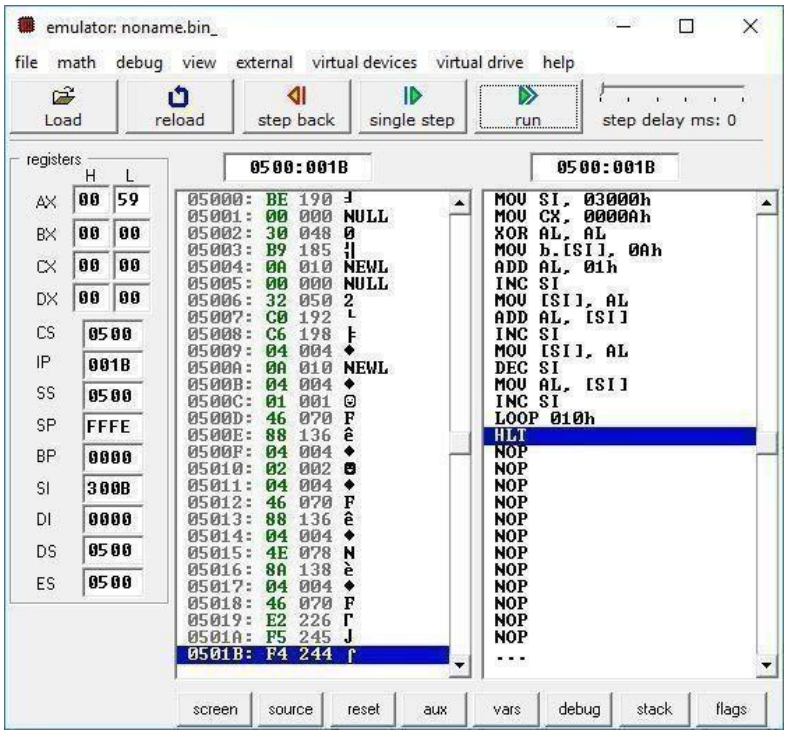
Write an assembly language program to print Fibonacci series in 8086.

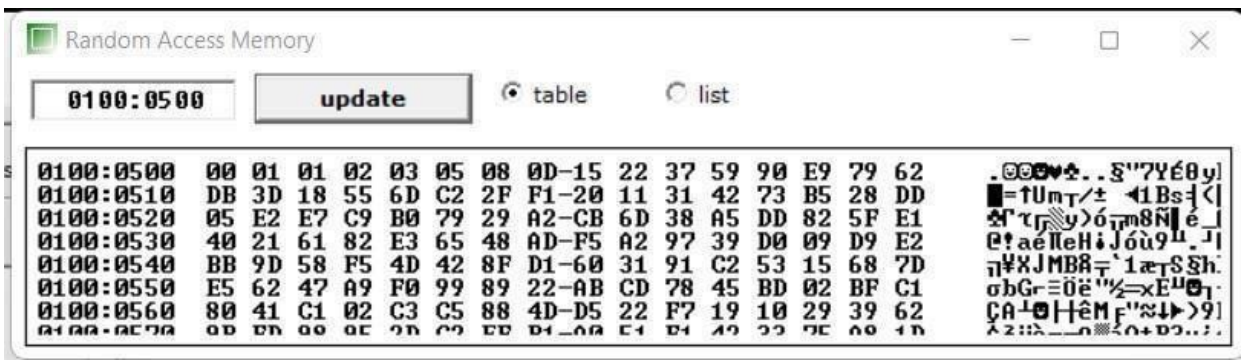
CODE:

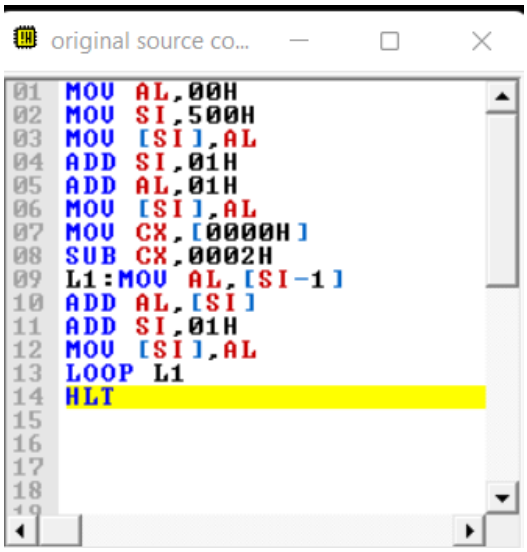
```
MOV AL,00H
MOV SI,500H
MOV [SI],AL
ADD SI,01H ADD
AL,01H
MOV [SI],AL
MOV CX,[0000H]
SUB CX,0002H
L1:MOV
AL,[SI-1] ADD
AL,[SI] ADD
SI,01H MOV
[SI],AL LOOP L1
HLT
```

Output:




emulator: noname.bin_
file math debug view external virtual devices virtual drive help
Load
reload
step back
single step
run
step delay ms: 0
registers
H L
AX 00 59
BX 00 00
CX 00 00
DX 00 00
CS 0500
IP 001B
SS 0500
SP FFFE
BP 0000
SI 300B
DI 0000
DS 0500
ES 0500
0500:001B
05000: BE 190
05001: 00 000 NULL
05002: 30 048 0
05003: B9 185
05004: 0A 010 NEWL
05005: 00 000 NULL
05006: 32 050 2
05007: C0 192
05008: C6 198
05009: 04 004
0500A: 0A 010 NEWL
0500B: 04 004
0500C: 01 001
0500D: 46 070 F
0500E: 88 136
0500F: 04 004
05010: 02 002
05011: 04 004
05012: 46 070 F
05013: 88 136
05014: 04 004
05015: 4E 078 N
05016: 8A 138
05017: 04 004
05018: 46 070 F
05019: E2 226
0501A: F5 245 J
0501B: F4 244
0500:001B
MOV SI, 03000h
MOV CX, 0000Ah
XOR AL, AL
MOV b.[SI], 0Ah
ADD AL, 01h
INC SI
MOV [SI], AL
ADD AL, [SI]
INC SI
MOV [SI], AL
DEC SI
MOV AL, [SI]
INC SI
LOOP 010h
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
screen source reset aux vars debug stack flags


Random Access Memory
0100:0500
update
table
list
0100:0500 00 01 01 02 03 05 08 0D-15 22 37 59 90 E9 79 62
0100:0510 DB 3D 18 55 6D C2 2F F1-20 11 31 42 73 B5 28 DD
0100:0520 05 E2 E7 C9 B0 79 29 A2-CB 6D 38 A5 DD 82 5F E1
0100:0530 40 21 61 82 E3 65 48 AD-F5 A2 97 39 D0 09 D9 E2
0100:0540 BB 9D 58 F5 4D 42 8F D1-60 31 91 C2 53 15 68 7D
0100:0550 E5 62 47 A9 F0 99 89 22-AB CD 78 45 BD 02 BF C1
0100:0560 80 41 C1 02 C3 C5 88 4D-D5 22 F7 19 10 29 39 62


original source co...
01 MOV AL,00H
02 MOV SI,500H
03 MOV [SI],AL
04 ADD SI,01H
05 ADD AL,01H
06 MOV [SI],AL
07 MOV CX,[0000H]
08 SUB CX,0002H
09 L1:MOV AL,[SI-1]
10 ADD AL,[SI]
11 ADD SI,01H
12 MOV [SI],AL
13 LOOP L1
14 HLT

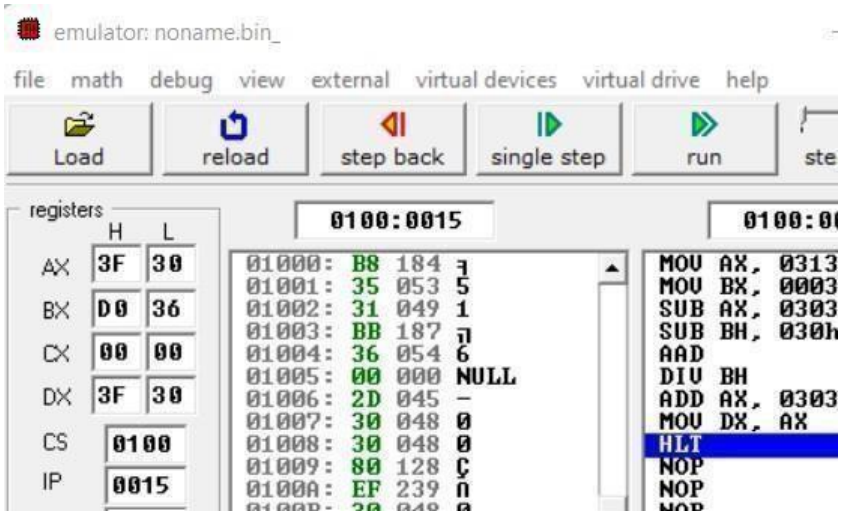
Program NO: 10

OBJECTIVE:

Write an assembly language program to perform the division 15/6 using the ASCII codes. Store the ASCII codes of the result in register DX.

CODE:

MOV AX,"15"
MOV BX, „6"
SUB AX,
3030H SUB
BH, 30H AAD
DIV BH
ADD AX, 3030H
MOV [SI], AX
HL



Experiment 11

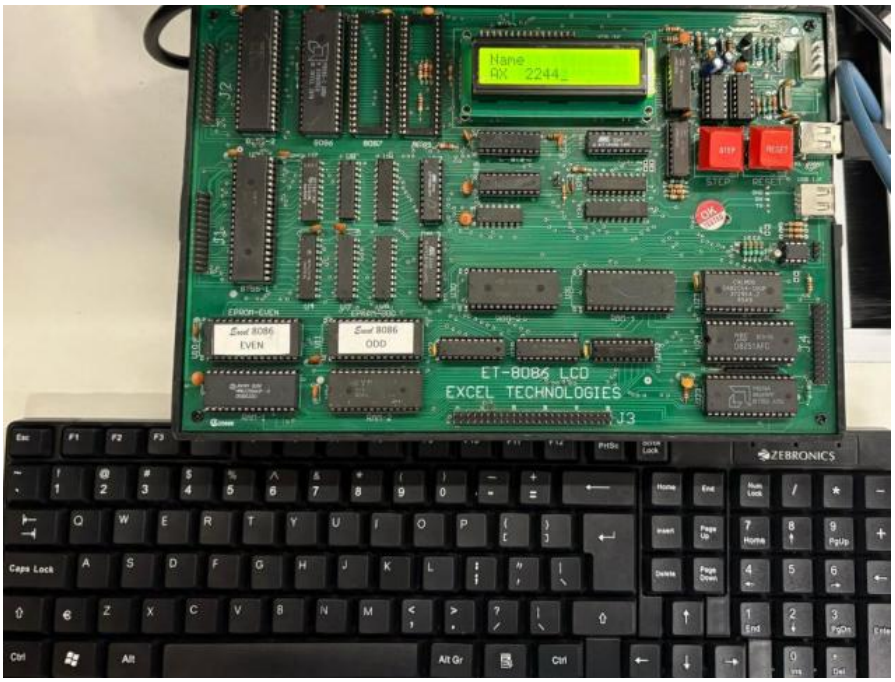
Steps for execution on the 8086 kit:

1. Press Reset
2. Press E from the keyboard
3. Press Enter
4. Write A1000:1000 (Assembly language Segment Address : Offset Address)
5. Press Enter
6. Now write your code line by line
7. In the last line of code write INT A5. INT A5 is similar to RST 5 in 8085 microprocessor.
8. Press Enter
9. Press Reset
10. Press G
11. Press Enter
12. It will show Burst mode it means that compile runs the code in one turn.
13. Press Enter
14. Now it will asks the segment address i.e. 1000
15. Press Enter
16. Now it will asks the offset address i.e. 1000
17. Press Enter
18. Now it has executed the program
19. It will show Cmd_word
20. Press S
21. Press Enter
22. Now you can check the contents from any place either from memory/register/IO
23. From whichever place you want to see the contents Press Enter at there.
24. For example I want to check the contents from register then I press Enter at Register.
25. Now it will asks the name of the register i.e. AX (It will show the contents at AX).
26. Finally you have successfully write the program in assembly language and execute it

```
MOV AX,1122
MOV BX,1122
ADD AX,BX
INTA5

MOV AX, [0301H]
MOV BX, [0303H]
ADD AX, BX
HLT
```

Output


Name
AX 2244
STEP
RESET
EVEN
ODD
ET-8086 LCD
EXCEL TECHNOLOGIES
J3
ZEBRONICS