Lab Assignment

8086-Microprocessor Kit Microprocessor-Based

System Design (UCS617) By

Richika	102103398
Hardik Sharma	102103402
Ishan Mathur	102103408
Aparna Singh	102103414
Akshay Khanna	102103415

Submitted to

Dr. Tanvi

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

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

PATIALA, PUNJAB (INDIA)

JAN – JUNE 2024

INDEX

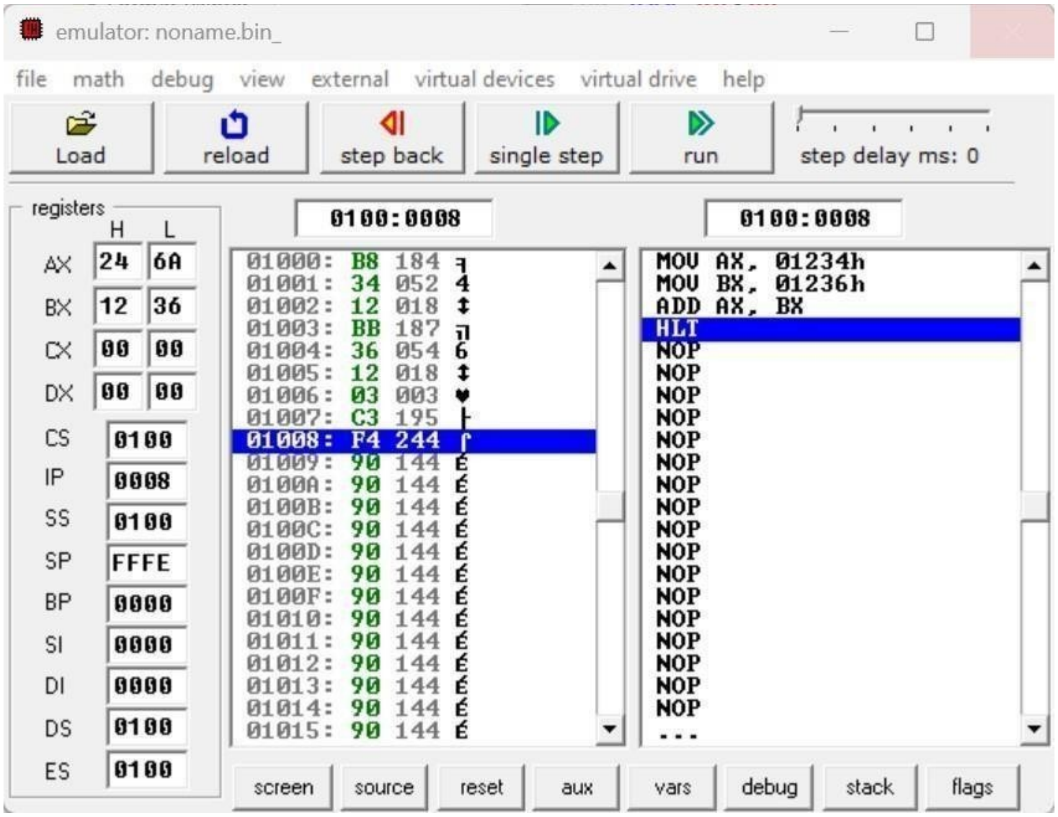
Experiment 1

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

Soln. -

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

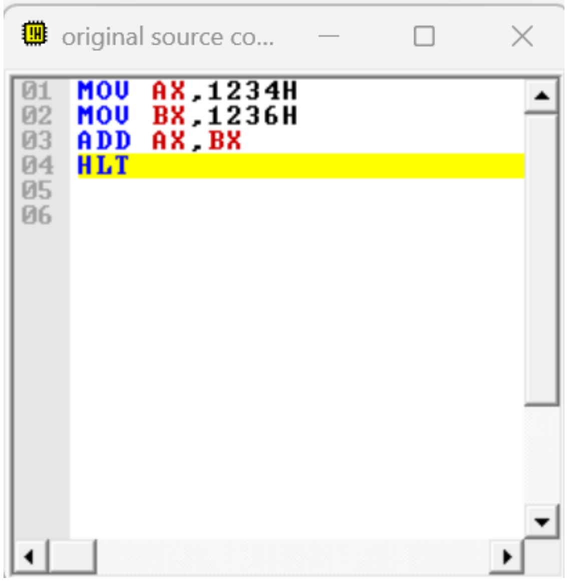
Output:



```
registers
      H   L
AX    24  6A
BX    12  36
CX    00  00
DX    00  00
CS    0100
IP    0008
SS    0100
SP    FFFE
BP    0000
SI    0000
DI    0000
DS    0100
ES    0100
```

```
0100:0008

01000: B8 184 ╕
01001: 34 052 4
01002: 12 018 ↕
01003: BB 187 ╗
01004: 36 054 6
01005: 12 018 ↕
01006: 03 003 ♥
01007: C3 195 ├
01008: F4 244 ⌠
01009: 90 144 É
0100A: 90 144 É
0100B: 90 144 É
0100C: 90 144 É
0100D: 90 144 É
0100E: 90 144 É
0100F: 90 144 É
01010: 90 144 É
01011: 90 144 É
01012: 90 144 É
01013: 90 144 É
01014: 90 144 É
01015: 90 144 É
```

```
0100:0008

MOV  AX, 01234h
MOV  BX, 01236h
ADD  AX, BX
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```



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

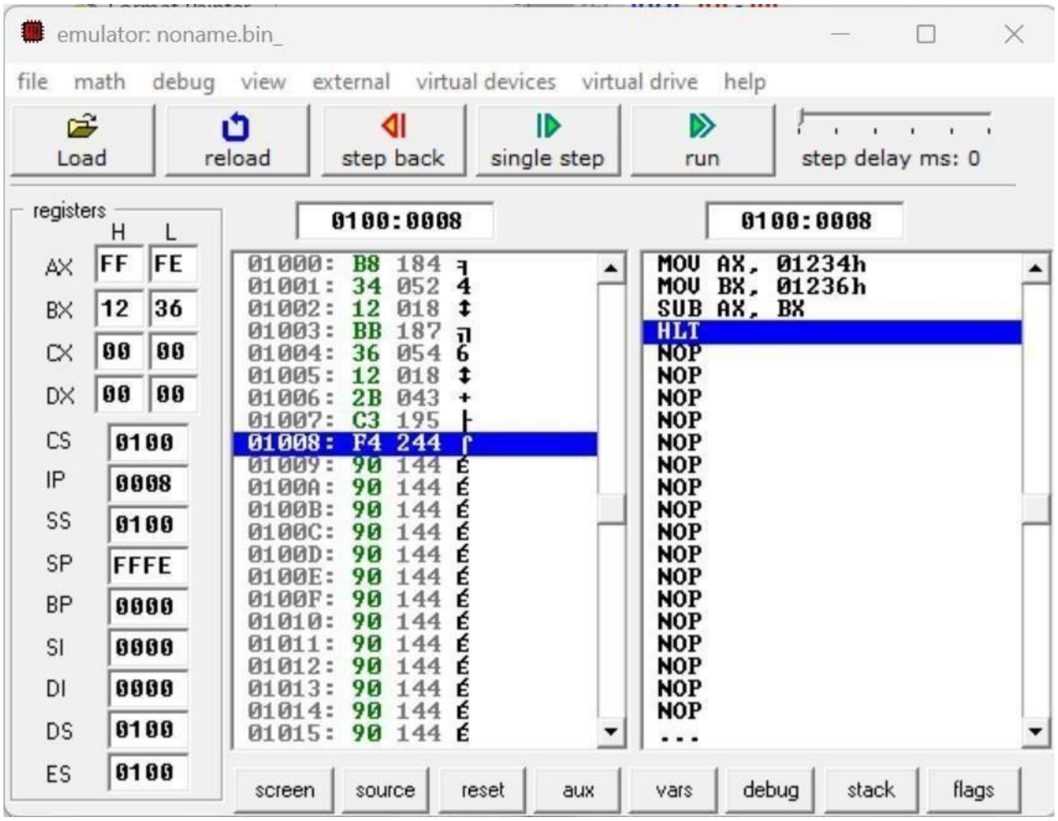
Experiment 2

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

Soln. -

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

Output:

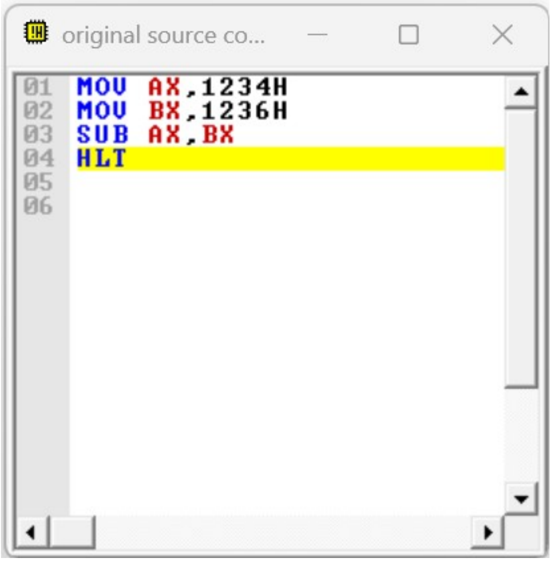


registers	H	L
AX	FF	FE
BX	12	36
CX	00	00
DX	00	00

CS 0100
IP 0008
SS 0100
SP FFFE
BP 0000
SI 0000
DI 0000
DS 0100
ES 0100

```
0100:0008

01000: B8 184 ╕
01001: 34 052 4
01002: 12 018 ↕
01003: BB 187 ╗
01004: 36 054 6
01005: 12 018 ↕
01006: 2B 043 +
01007: C3 195 ├
01008: F4 244 ⌠
01009: 90 144 É
0100A: 90 144 É
0100B: 90 144 É
0100C: 90 144 É
0100D: 90 144 É
0100E: 90 144 É
0100F: 90 144 É
01010: 90 144 É
01011: 90 144 É
01012: 90 144 É
01013: 90 144 É
01014: 90 144 É
01015: 90 144 É
```

```
0100:0008

MOV AX, 01234h
MOV BX, 01236h
SUB AX, BX
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```



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

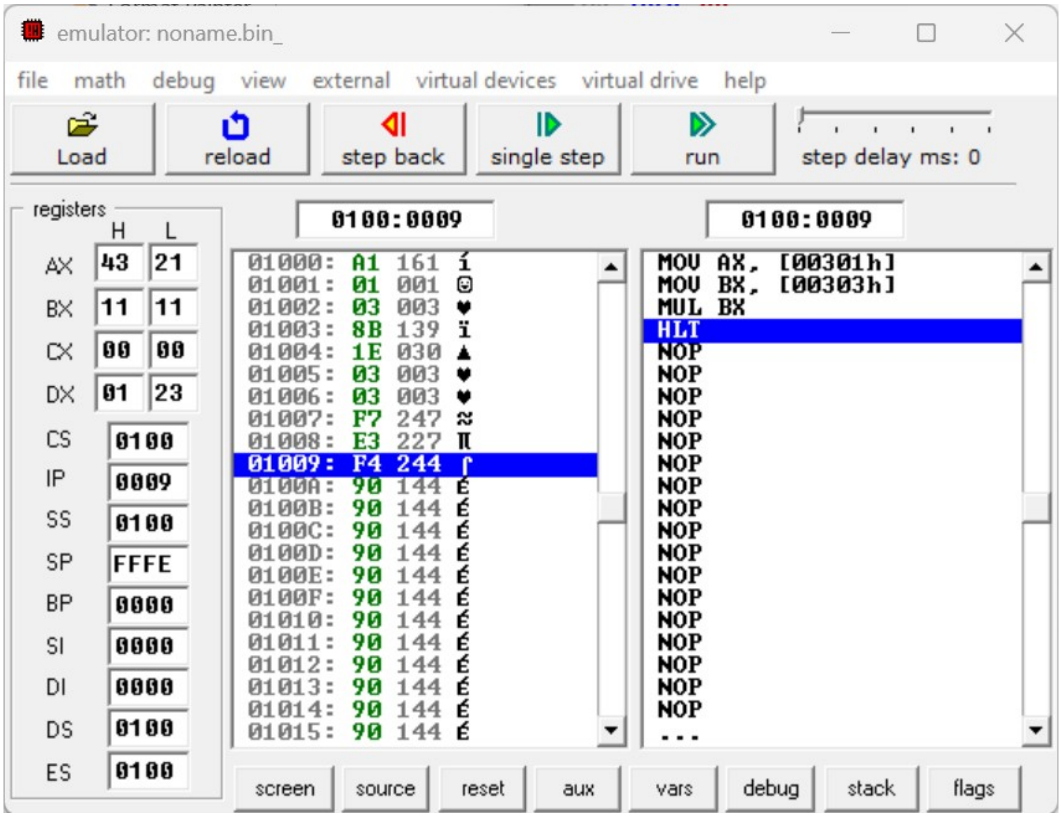
Experiment 3

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

Soln. -

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

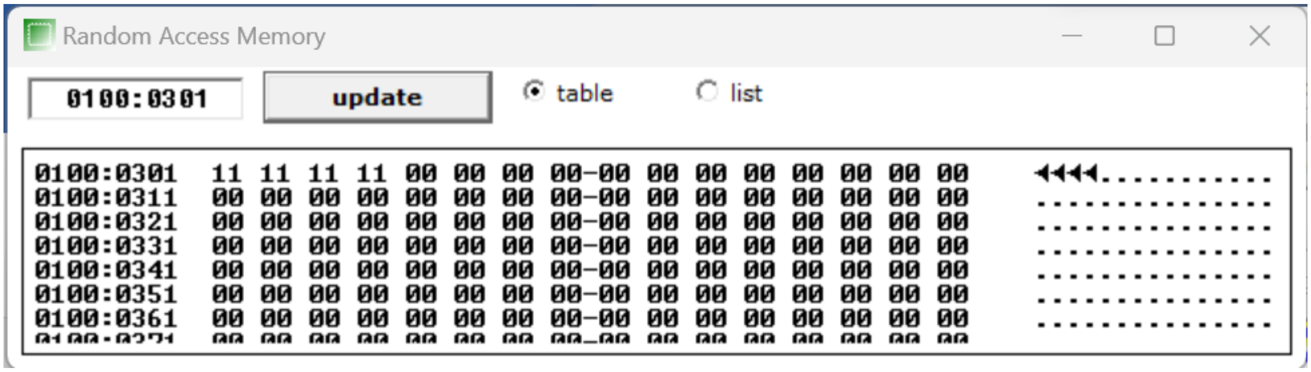
Output:



	H	L
AX	43	21
BX	11	11
CX	00	00
DX	01	23
CS	0100	
IP	0009	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

```
0100:0009

01000: A1 161 í
01001: 01 001 ☺
01002: 03 003 ♥
01003: 8B 139 ï
01004: 1E 030 ▲
01005: 03 003 ♥
01006: 03 003 ♥
01007: F7 247 ≈
01008: E3 227 π
01009: F4 244 ⌠
0100A: 90 144 É
0100B: 90 144 É
0100C: 90 144 É
0100D: 90 144 É
0100E: 90 144 É
0100F: 90 144 É
01010: 90 144 É
01011: 90 144 É
01012: 90 144 É
01013: 90 144 É
01014: 90 144 É
01015: 90 144 É
```

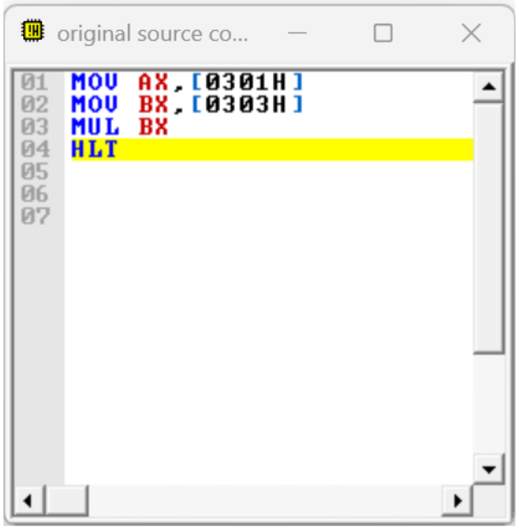
```
0100:0009

MOV AX, [00301h]
MOV BX, [00303h]
MUL BX
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```



0100:0301

● 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   ................
0100:0371  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```



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

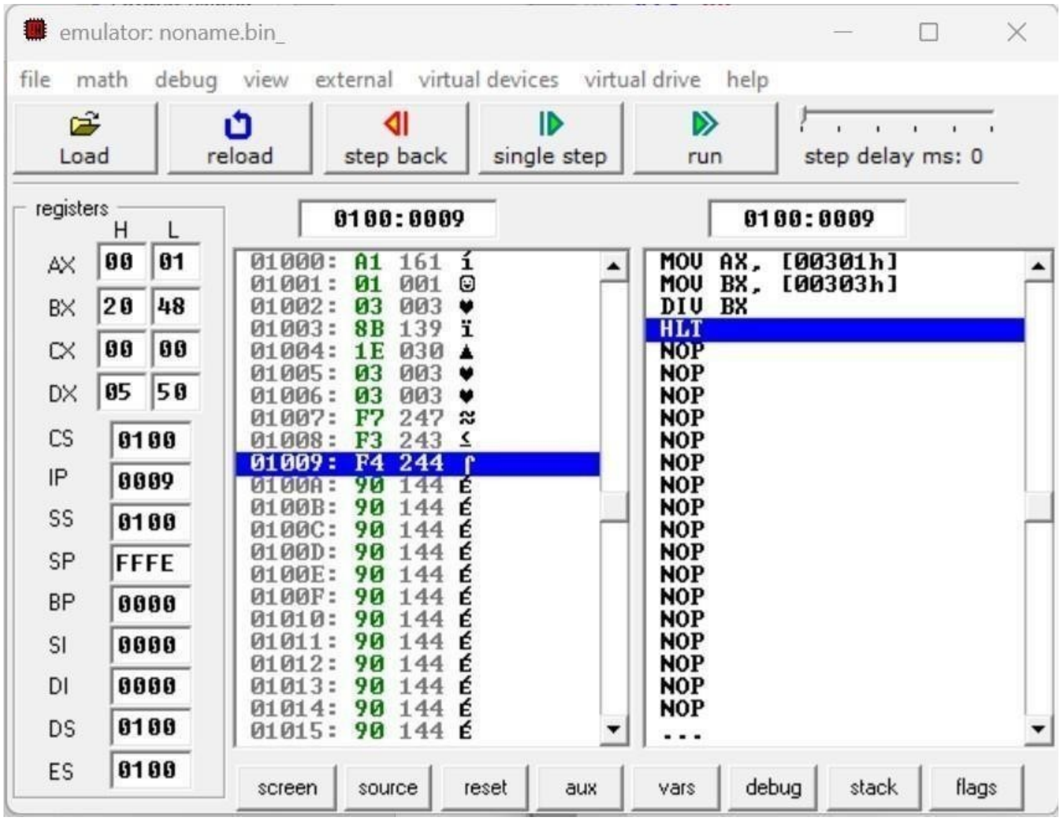
Experiment 4

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

Soln. -

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

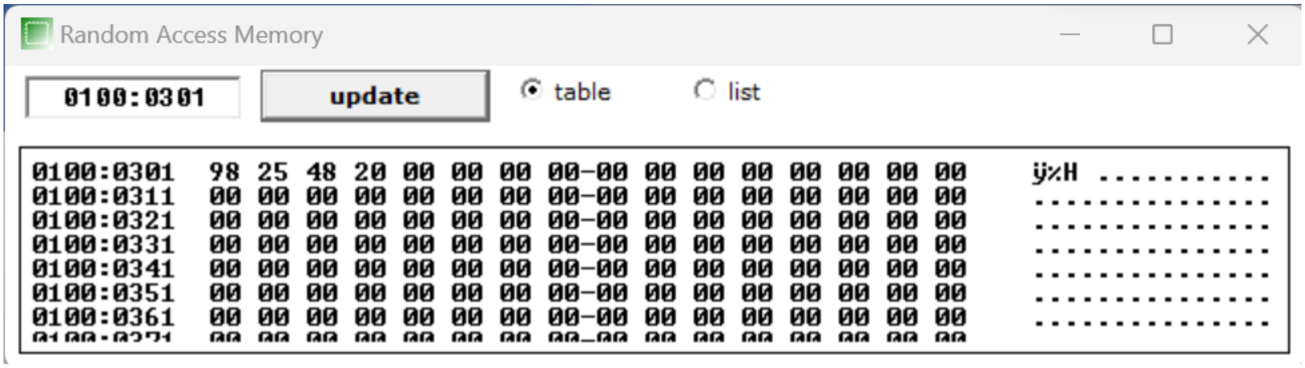
Output:



```
      H  L
AX   00 01
BX   20 48
CX   00 00
DX   05 50
CS   0100
IP   0009
SS   0100
SP   FFFE
BP   0000
SI   0000
DI   0000
DS   0100
ES   0100
```

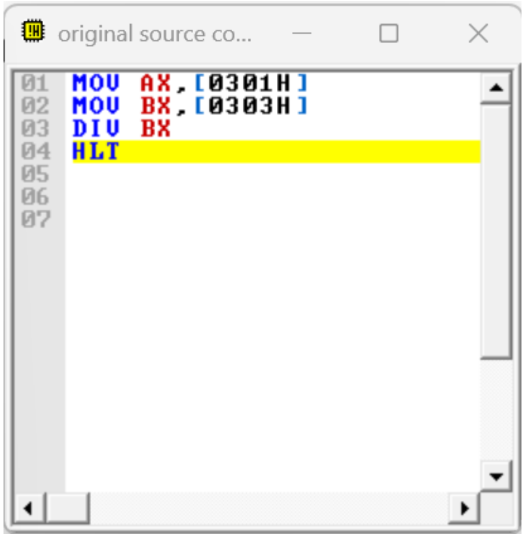
```
0100:0009
01000: A1 161 í
01001: 01 001 ☺
01002: 03 003 ♥
01003: 8B 139 ï
01004: 1E 030 ▲
01005: 03 003 ♥
01006: 03 003 ♥
01007: F7 247 ≈
01008: F3 243 ≤
01009: F4 244 ⌠
0100A: 90 144 É
0100B: 90 144 É
0100C: 90 144 É
0100D: 90 144 É
0100E: 90 144 É
0100F: 90 144 É
01010: 90 144 É
01011: 90 144 É
01012: 90 144 É
01013: 90 144 É
01014: 90 144 É
01015: 90 144 É
```

```
0100:0009
MOV  AX, [00301h]
MOV  BX, [00303h]
DIV  BX
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```



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     ................
0100:0371   00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```



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

Experiment 5

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

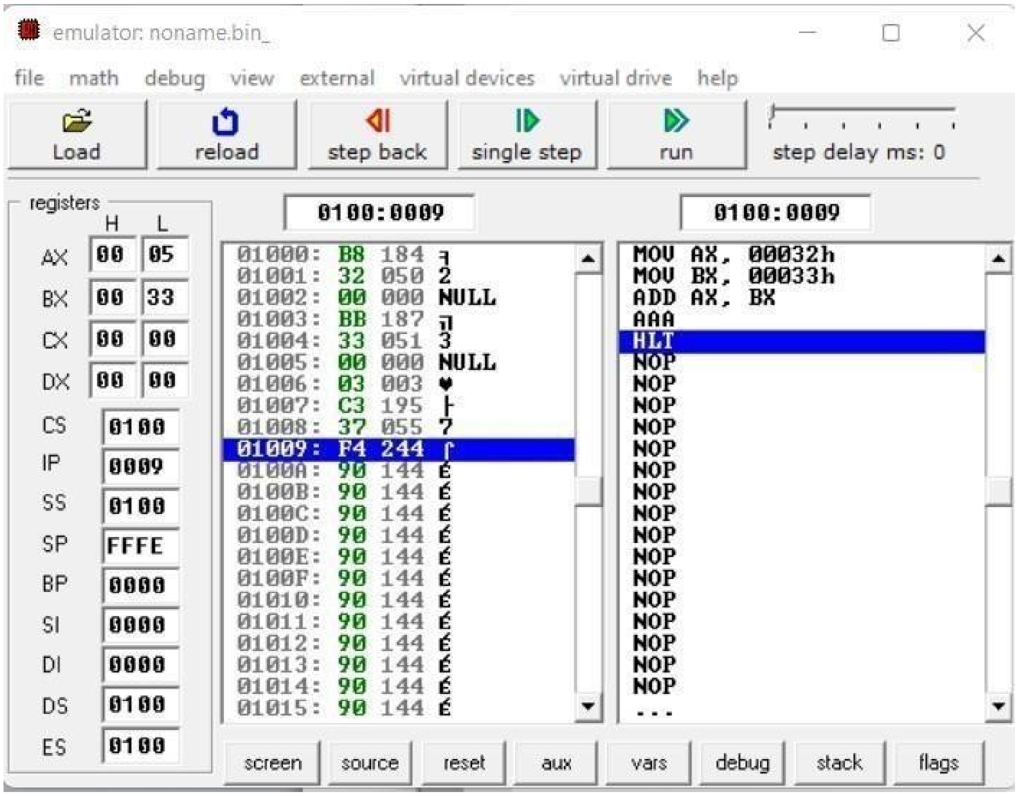
Soln.

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

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

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

AAA Instruction

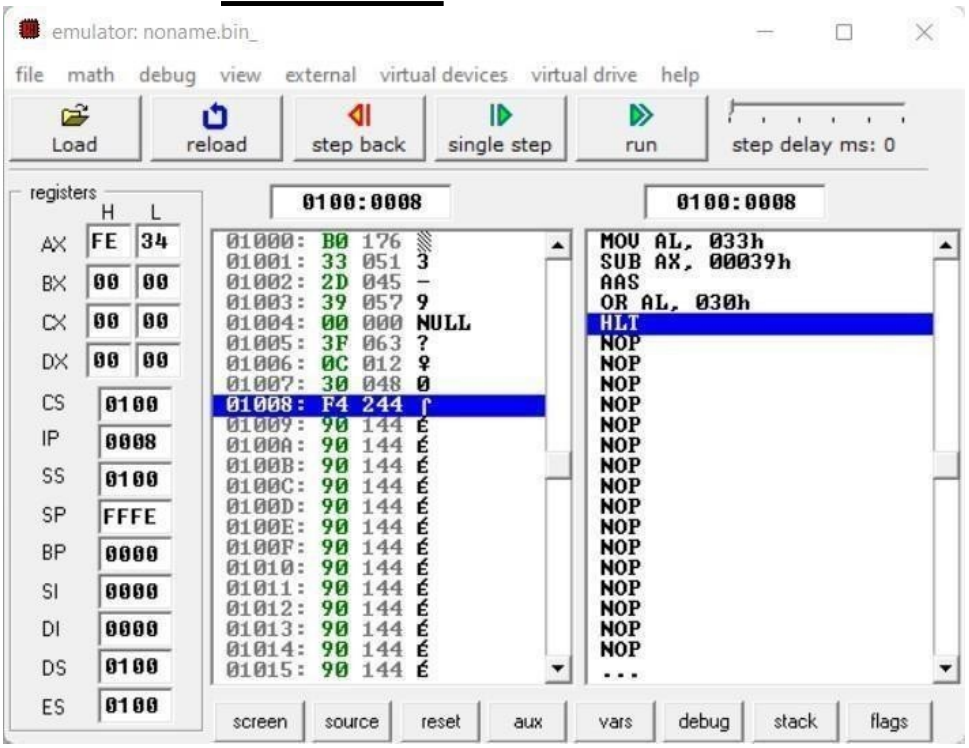


```
      H   L
AX   00  05
BX   00  33
CX   00  00
DX   00  00
CS   0100
IP   0009
SS   0100
SP   FFFE
BP   0000
SI   0000
DI   0000
DS   0100
ES   0100
```

```
0100:0009
01000: B8 184 ╕
01001: 32 050 2
01002: 00 000 NULL
01003: BB 187 ╗
01004: 33 051 3
01005: 00 000 NULL
01006: 03 003 ♥
01007: C3 195 ├
01008: 37 055 7
01009: F4 244 ⌠
0100A: 90 144 É
0100B: 90 144 É
0100C: 90 144 É
0100D: 90 144 É
0100E: 90 144 É
0100F: 90 144 É
01010: 90 144 É
01011: 90 144 É
01012: 90 144 É
01013: 90 144 É
01014: 90 144 É
01015: 90 144 É
```

```
0100:0009
MOV AX, 00032h
MOV BX, 00033h
ADD AX, BX
AAA
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

AAS Instruction

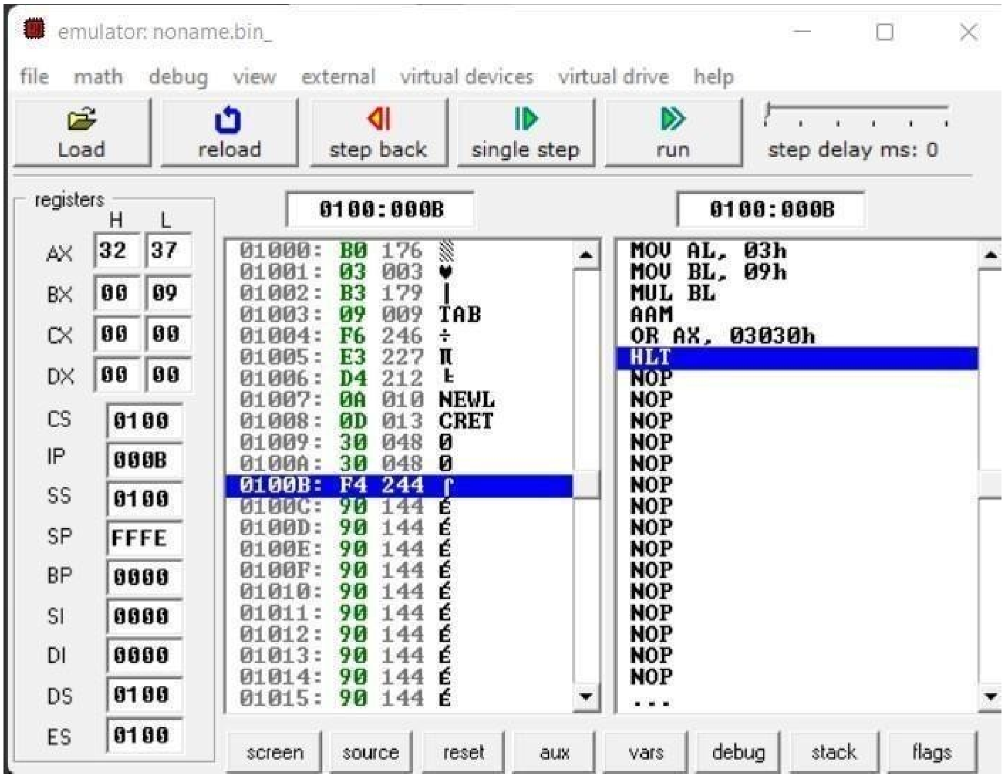


```
      H   L
AX   FE  34
BX   00  00
CX   00  00
DX   00  00
CS   0100
IP   0008
SS   0100
SP   FFFE
BP   0000
SI   0000
DI   0000
DS   0100
ES   0100
```

```
0100:0008
01000: B0 176 ░
01001: 33 051 3
01002: 2D 045 -
01003: 39 057 9
01004: 00 000 NULL
01005: 3F 063 ?
01006: 0C 012 ♀
01007: 30 048 0
01008: F4 244 ⌠
01009: 90 144 É
0100A: 90 144 É
0100B: 90 144 É
0100C: 90 144 É
0100D: 90 144 É
0100E: 90 144 É
0100F: 90 144 É
01010: 90 144 É
01011: 90 144 É
01012: 90 144 É
01013: 90 144 É
01014: 90 144 É
01015: 90 144 É
```

```
0100:0008
MOV AL, 033h
SUB AX, 00039h
AAS
OR AL, 030h
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

AAM Instruction

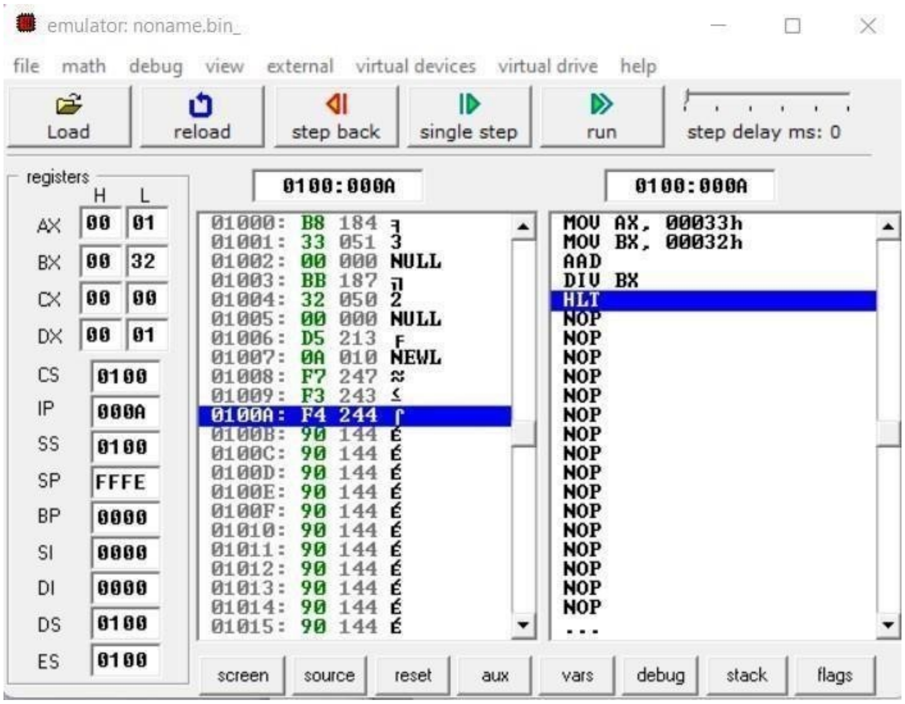


```
registers
     H   L
AX   32  37
BX   00  09
CX   00  00
DX   00  00
CS   0100
IP   000B
SS   0100
SP   FFFE
BP   0000
SI   0000
DI   0000
DS   0100
ES   0100
```

```
0100:000B

01000: B0 176 ░
01001: 03 003 ♥
01002: B3 179 │
01003: 09 009 TAB
01004: F6 246 ÷
01005: E3 227 π
01006: D4 212 ╘
01007: 0A 010 NEWL
01008: 0D 013 CRET
01009: 30 048 0
0100A: 30 048 0
0100B: F4 244 ⌠
0100C: 90 144 É
0100D: 90 144 É
0100E: 90 144 É
0100F: 90 144 É
01010: 90 144 É
01011: 90 144 É
01012: 90 144 É
01013: 90 144 É
01014: 90 144 É
01015: 90 144 É
```

```
0100:000B

MOV AL, 03h
MOV BL, 09h
MUL BL
AAM
OR AX, 03030h
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

AAD Instruction

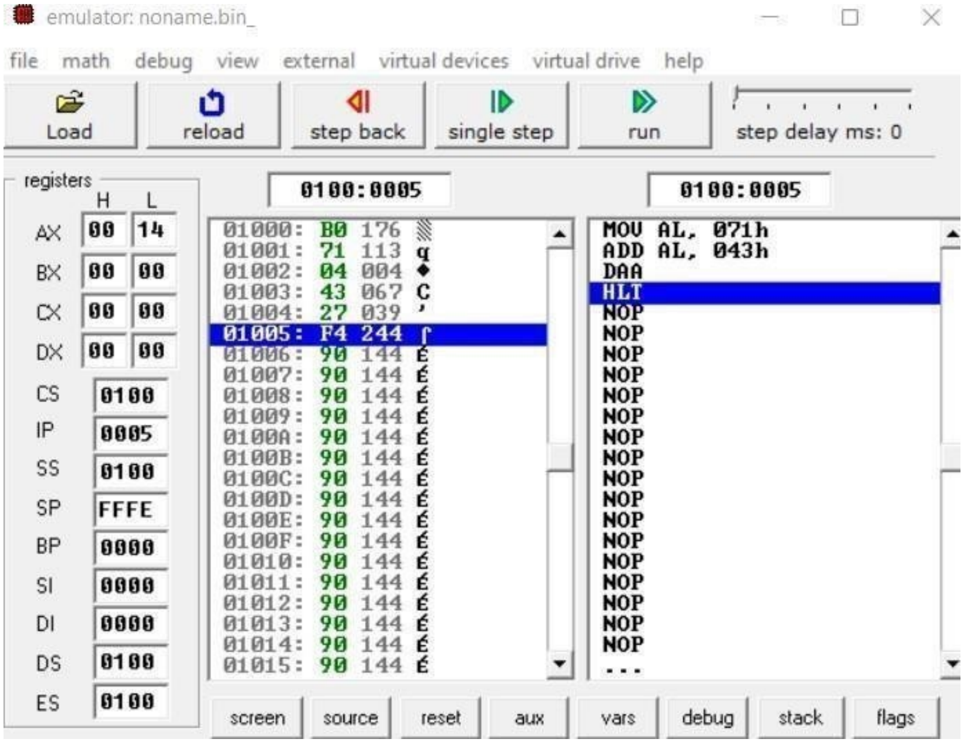


```
registers
     H   L
AX   00  01
BX   00  32
CX   00  00
DX   00  01
CS   0100
IP   000A
SS   0100
SP   FFFE
BP   0000
SI   0000
DI   0000
DS   0100
ES   0100
```

```
0100:000A

01000: B8 184 ╕
01001: 33 051 3
01002: 00 000 NULL
01003: BB 187 ╗
01004: 32 050 2
01005: 00 000 NULL
01006: D5 213 ╒
01007: 0A 010 NEWL
01008: F7 247 ≈
01009: F3 243 ≤
0100A: F4 244 ⌠
0100B: 90 144 É
0100C: 90 144 É
0100D: 90 144 É
0100E: 90 144 É
0100F: 90 144 É
01010: 90 144 É
01011: 90 144 É
01012: 90 144 É
01013: 90 144 É
01014: 90 144 É
01015: 90 144 É
```

```
0100:000A

MOV AX, 00033h
MOV BX, 00032h
AAD
DIV BX
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

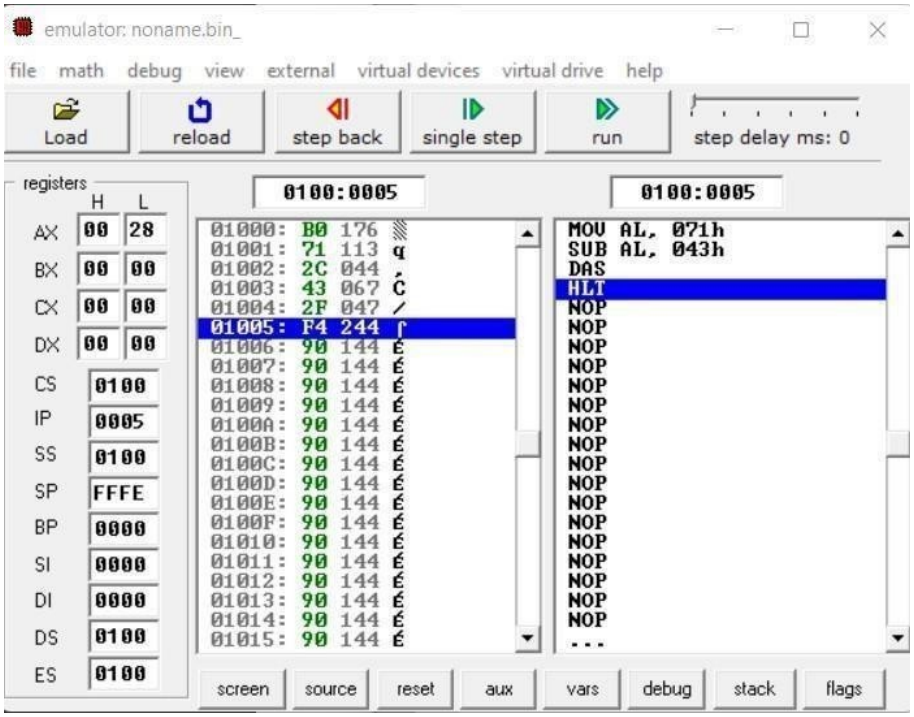
DAA Instruction



```
registers
      H    L
AX   00   14
BX   00   00
CX   00   00
DX   00   00
CS   0100
IP   0005
SS   0100
SP   FFFE
BP   0000
SI   0000
DI   0000
DS   0100
ES   0100
```

```
0100:0005                       0100:0005

01000: B0 176 ▒                 MOV AL, 071h
01001: 71 113 q                 ADD AL, 043h
01002: 04 004 ♦                 DAA
01003: 43 067 C                 HLT
01004: 27 039 '                 NOP
01005: F4 244 ⌠                 NOP
01006: 90 144 É                 NOP
01007: 90 144 É                 NOP
01008: 90 144 É                 NOP
01009: 90 144 É                 NOP
0100A: 90 144 É                 NOP
0100B: 90 144 É                 NOP
0100C: 90 144 É                 NOP
0100D: 90 144 É                 NOP
0100E: 90 144 É                 NOP
0100F: 90 144 É                 NOP
01010: 90 144 É                 NOP
01011: 90 144 É                 NOP
01012: 90 144 É                 NOP
01013: 90 144 É                 NOP
01014: 90 144 É                 NOP
01015: 90 144 É                 ...
```

DAS Instruction



```
registers
      H    L
AX   00   28
BX   00   00
CX   00   00
DX   00   00
CS   0100
IP   0005
SS   0100
SP   FFFE
BP   0000
SI   0000
DI   0000
DS   0100
ES   0100
```

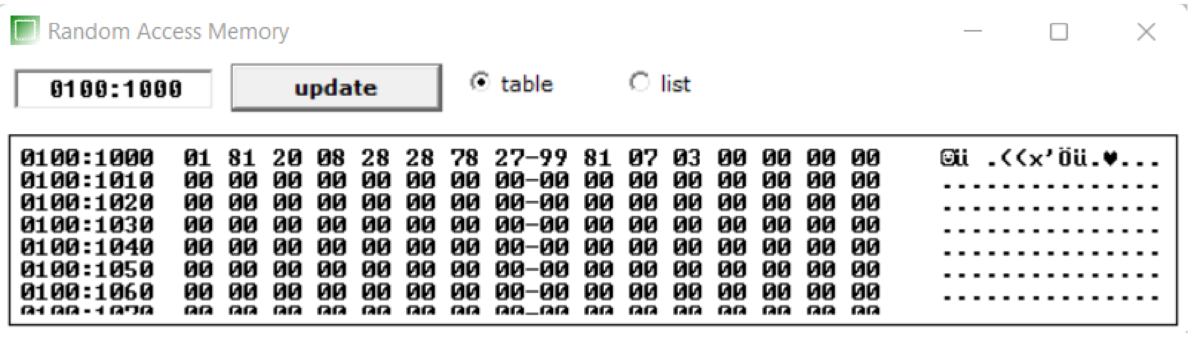
```
0100:0005                       0100:0005

01000: B0 176 ▒                 MOV AL, 071h
01001: 71 113 q                 SUB AL, 043h
01002: 2C 044 ,                 DAS
01003: 43 067 C                 HLT
01004: 2F 047 /                 NOP
01005: F4 244 ⌠                 NOP
01006: 90 144 É                 NOP
01007: 90 144 É                 NOP
01008: 90 144 É                 NOP
01009: 90 144 É                 NOP
0100A: 90 144 É                 NOP
0100B: 90 144 É                 NOP
0100C: 90 144 É                 NOP
0100D: 90 144 É                 NOP
0100E: 90 144 É                 NOP
0100F: 90 144 É                 NOP
01010: 90 144 É                 NOP
01011: 90 144 É                 NOP
01012: 90 144 É                 NOP
01013: 90 144 É                 NOP
01014: 90 144 É                 NOP
01015: 90 144 É                 ...
```

Experiment 6

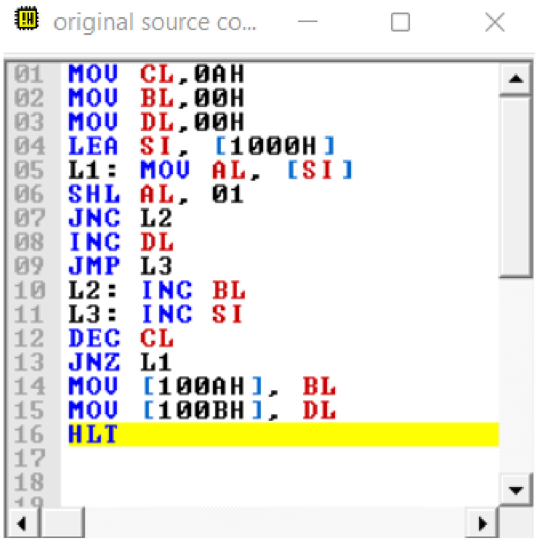
Q. Write an assembly language program to find out the count of positive numbers and negative numbers from a series of signed numbers in 8086.

Soln. -

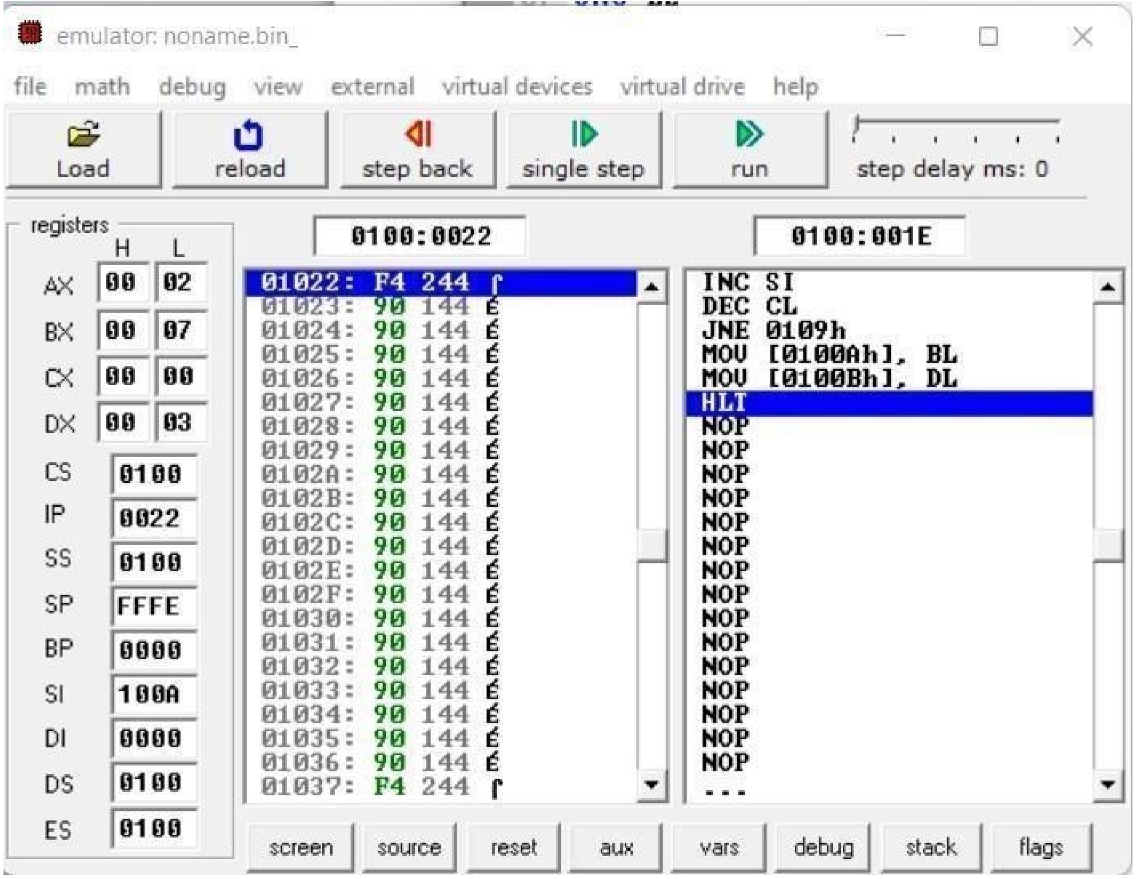
```
MOV CL,0AH
MOV BL,00H
MOV DL,00H
LEA SI, [1000H]
L1: MOV AL, [SI]
SHL AL, 01
JNC L2
INC DL
JMP  L3
L2: INC
BL L3:
INC SI
DEC CL
JNZ L1
MOV [100AH], BL
MOV [100BH], DL
HLT
```



```
0100:1000

0100:1000  01 81 20 08 28 28 78 27-99 81 07 03 00 00 00 00    ☺ü .((x'Öü.♥...
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    ................
0100:1070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```



```
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
```



```
    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 ⌠
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
0102C: 90 144 É
0102D: 90 144 É
0102E: 90 144 É
0102F: 90 144 É
01030: 90 144 É
01031: 90 144 É
01032: 90 144 É
01033: 90 144 É
01034: 90 144 É
01035: 90 144 É
01036: 90 144 É
01037: F4 244 ⌠
```

```
0100:001E
INC  SI
DEC  CL
JNE  0109h
MOV  [0100Ah], BL
MOV  [0100Bh], DL
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

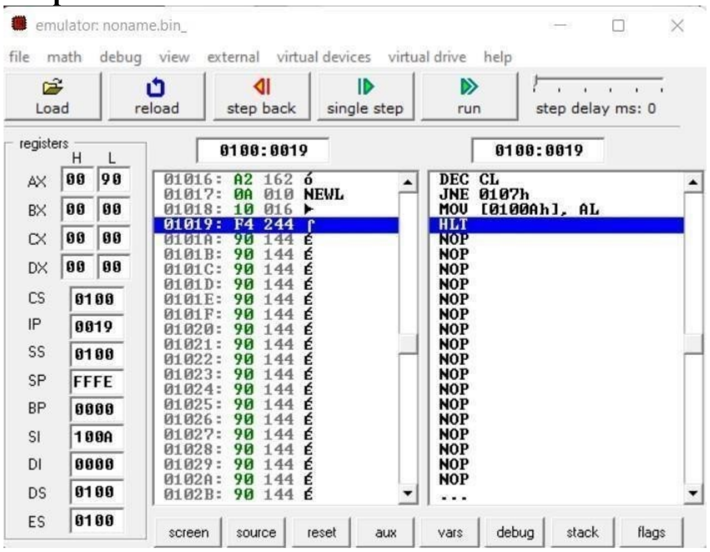
Experiment 7

Q. 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.

Soln. -

```
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
```

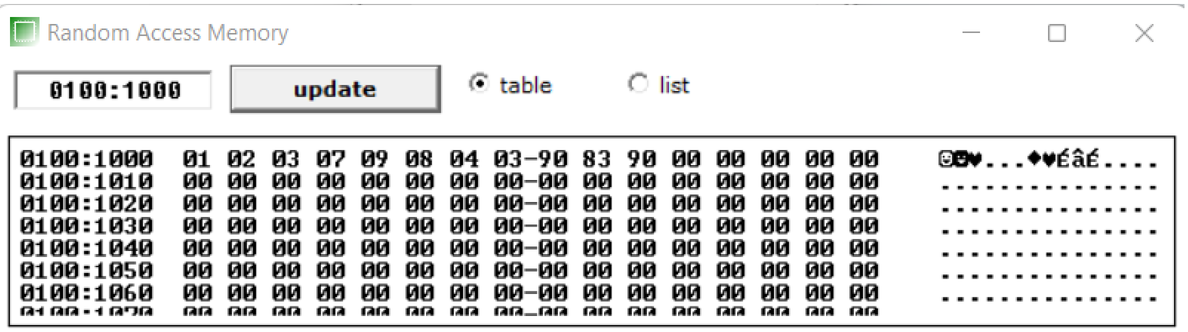
Output:-



```
     H  L
AX  00 90
BX  00 00
CX  00 00
DX  00 00
CS  0100
IP  0019
SS  0100
SP  FFFE
BP  0000
SI  100A
DI  0000
DS  0100
ES  0100
```

```
0100:0019

01016: A2 162 ó
01017: 0A 010 NEWL
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

DEC CL
JNE 0107h
MOV [0100Ah], AL
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```



0100:1000

● table ○ list

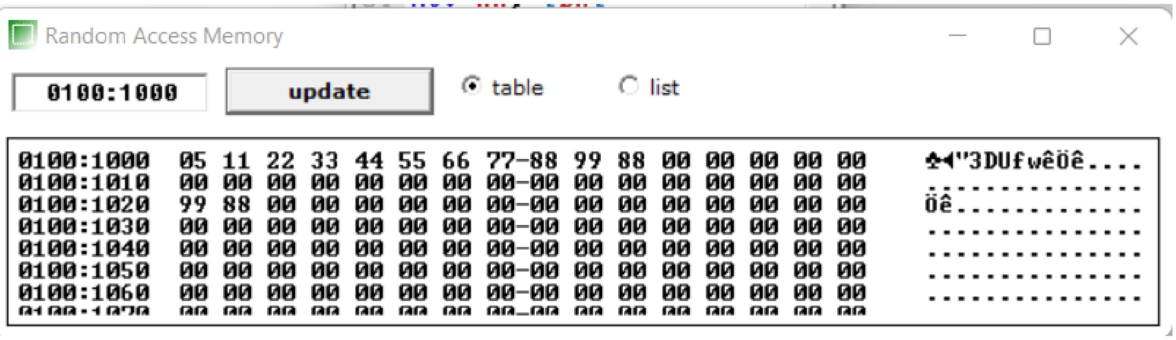
```
0100:1000  01 02 03 07 09 08 04 03-90 83 90 00 00 00 00 00    ☺☻♥...♦♥ÉâÉ....
0100:1010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    ...............
0100:1020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    ...............
0100:1030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00    ...............
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    ...............
0100:1070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

Experiment 8

Q. 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.

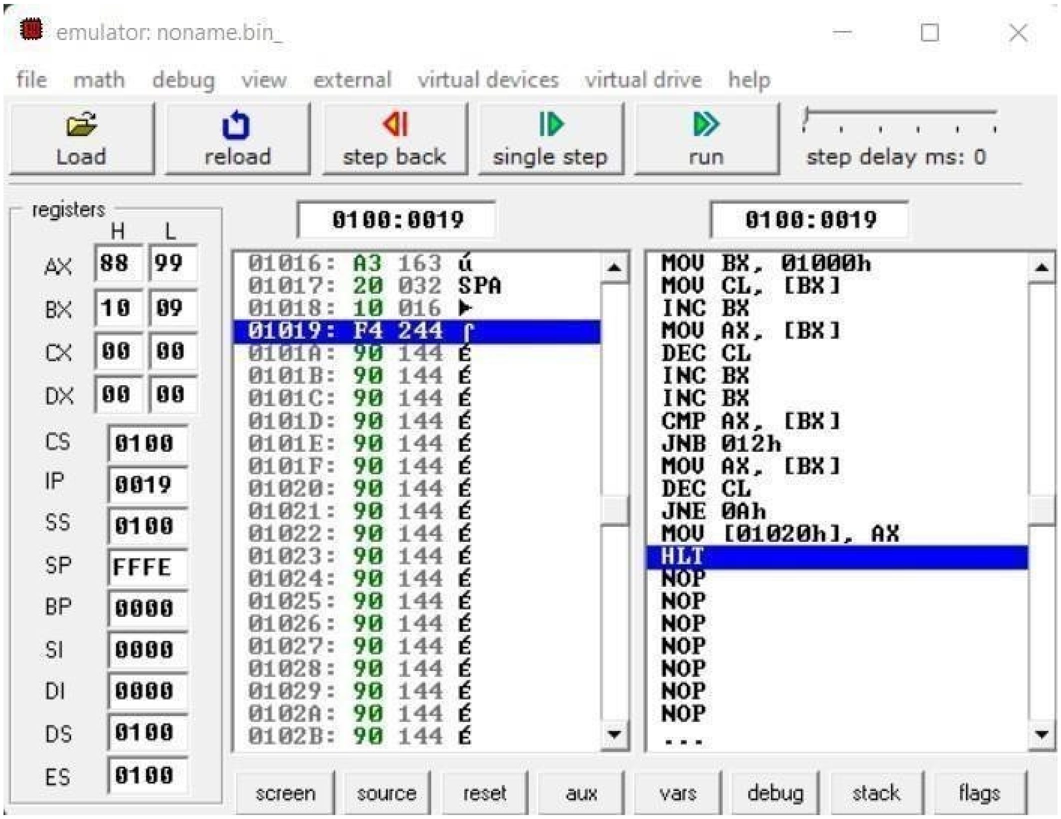
Soln. -

```
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
```



```
0100:1000
```

```
0100:1000  05 11 22 33 44 55 66 77-88 99 88 00 00 00 00 00   ♣◄"3DUfwêÖê....
0100:1010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0100:1020  99 88 00 00 00 00 00 00-00 00 00 00 00 00 00 00   Öê..............
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   ................
0100:1070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```



```
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
...
```

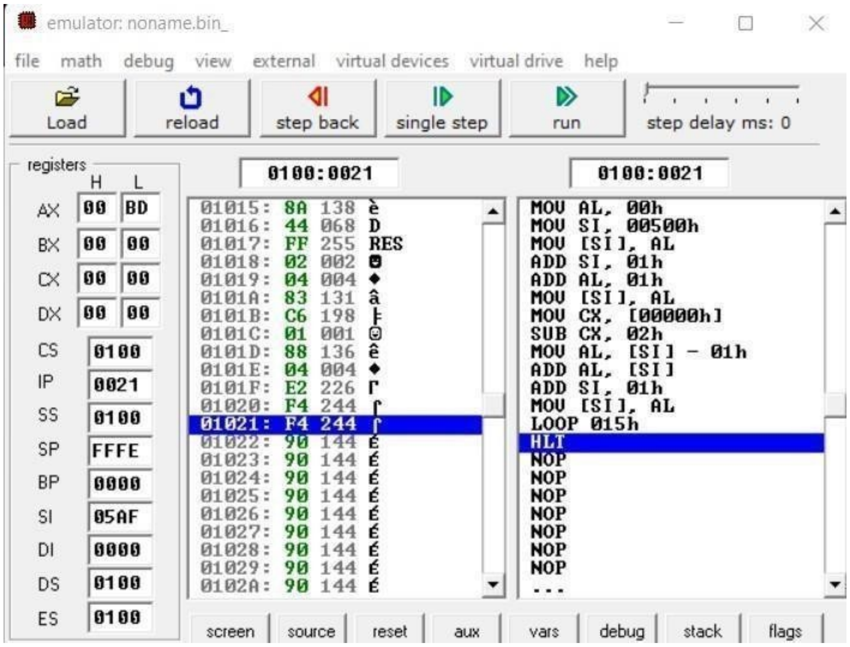
Experiment 9

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

Soln. -

```
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:

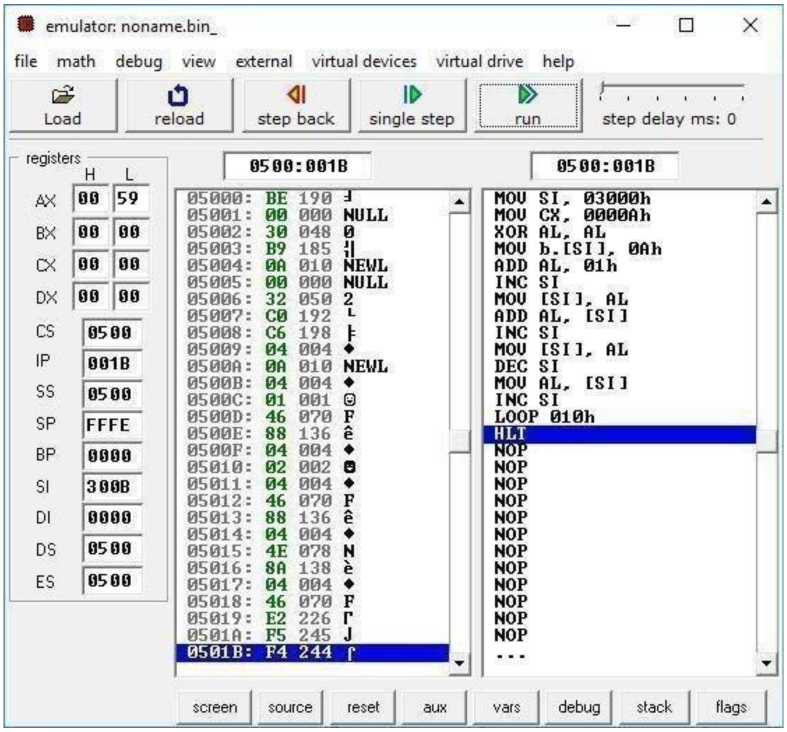


	H	L
AX	00	BD
BX	00	00
CX	00	00
DX	00	00

CS	0100
IP	0021
SS	0100
SP	FFFE
BP	0000
SI	05AF
DI	0000
DS	0100
ES	0100

```
0100:0021
01015: 8A 138 è
01016: 44 068 D
01017: FF 255 RES
01018: 02 002 ☻
01019: 04 004 ♦
0101A: 83 131 â
0101B: C6 198 ╞
0101C: 01 001 ☺
0101D: 88 136 ê
0101E: 04 004 ♦
0101F: E2 226 Γ
01020: F4 244 ⌠
01021: F4 244 ⌠
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
```

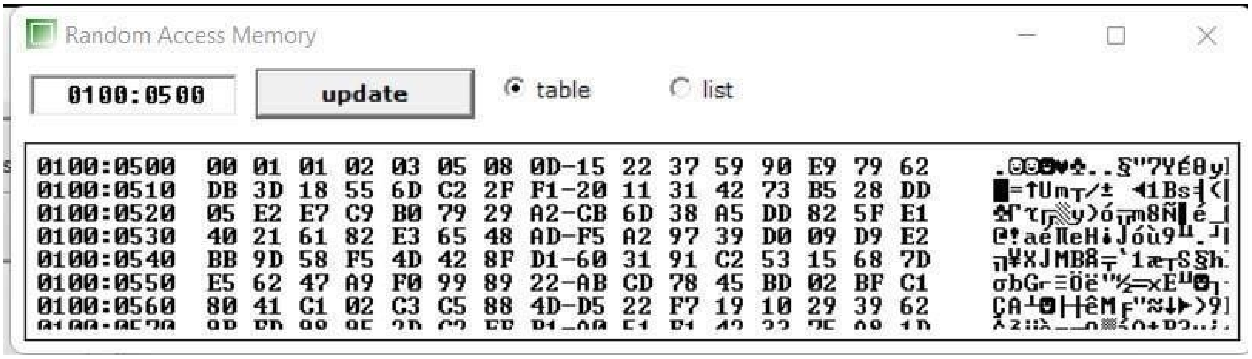
```
0100:0021
MOV AL, 00h
MOV SI, 00500h
MOV [SI], AL
ADD SI, 01h
ADD AL, 01h
MOV [SI], AL
MOV CX, [00000h]
SUB CX, 02h
MOV AL, [SI] - 01h
ADD AL, [SI]
ADD SI, 01h
MOV [SI], AL
LOOP 015h
HLT
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```



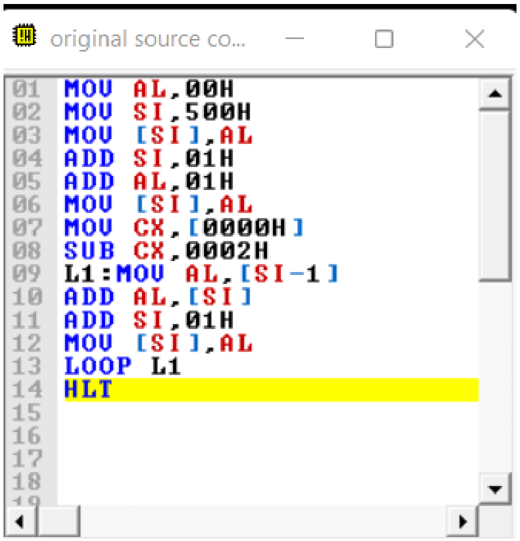
	H	L
AX	00	59
BX	00	00
CX	00	00
DX	00	00
CS	0500	
IP	001B	
SS	0500	
SP	FFFE	
BP	0000	
SI	300B	
DI	0000	
DS	0500	
ES	0500	

```
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 ê
0500F: 04 004 ♦
05010: 02 002 ☻
05011: 04 004 ♦
05012: 46 070 F
05013: 88 136 ê
05014: 04 004 ♦
05015: 4E 078 N
05016: 8A 138 è
05017: 04 004 ♦
05018: 46 070 F
05019: E2 226 Γ
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
...
```



```
0100:0500

0100:0500  00 01 01 02 03 05 08 0D-15 22 37 59 90 E9 79 62   .☺☺☻♥♣..§"7YÉθy]
0100:0510  DB 3D 18 55 6D C2 2F F1-20 11 31 42 73 B5 28 DD   █=↑Um┬/± ◄1Bs╡(▌
0100:0520  05 E2 E7 C9 B0 79 29 A2-CB 6D 38 A5 DD 82 5F E1   ♣Γτ╔░y)ó╦m8Ñ▌é_ß
0100:0530  40 21 61 82 E3 65 48 AD-F5 A2 97 39 D0 09 D9 E2   @!aéπeH¡JóùΘ╨.┘Γ
0100:0540  BB 9D 58 F5 4D 42 8F D1-60 31 91 C2 53 15 68 7D   ╗¥XJMBÅ╤`1æ┬S§h}
0100:0550  E5 62 47 A9 F0 99 89 22-AB CD 78 45 BD 02 BF C1   σbG⌐≡Öë"½═xE╜☻┐┴
0100:0560  80 41 C1 02 C3 C5 88 4D-D5 22 F7 19 10 29 39 62   ÇA┴☻├┼êMF"≈↓►)9b
```



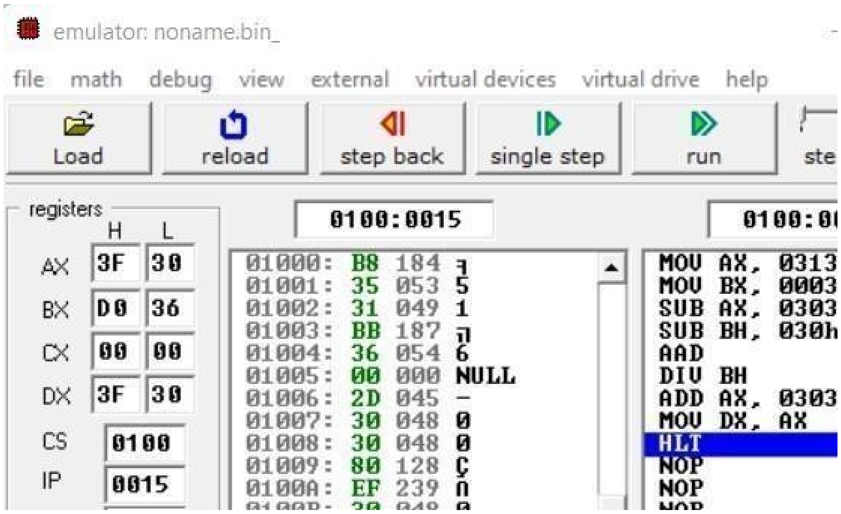
```
MOV AL,00H
MOV SI,500H
MOV [SI],AL
ADD SI,01H
ADD AL,01H
MOV [SI],AL
MOV CX,[0000H]
SUB CX,0002H
L1:MOV AL,[SI-1]
ADD AL,[SI]
ADD SI,01H
MOV [SI],AL
LOOP L1
HLT
```

Experiment 10

Q. 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.

Soln. -

```
MOV AX,“15”
MOV BX,“6”
SUB AX, 3030H
SUB BH, 30H
AAD
DIV BH
ADD AX, 3030H
MOV [SI], AX
HL
```



```
AX  3F 30
BX  D0 36
CX  00 00
DX  3F 30
CS  0100
IP  0015
```

```
0100:0015
01000: B8 184 ╕
01001: 35 053 5
01002: 31 049 1
01003: BB 187 ╗
01004: 36 054 6
01005: 00 000 NULL
01006: 2D 045 -
01007: 30 048 0
01008: 30 048 0
01009: 80 128 Ç
0100A: EF 239 ∩
```

```
0100:0(
MOV AX, 0313
MOV BX, 0003
SUB AX, 0303
SUB BH, 030h
AAD
DIV BH
ADD AX, 0303
MOV DX, AX
HLT
NOP
NOP
```