DEPARTMENT OF COMPUTER &
SOFTWARE ENGINEERING

COLLEGE OF E&ME, NUST, RAWALPINDI

Microprocessor and Microcontroller Based Design

Lab 01

SUBMITTED TO:
Dr Taimoor Zahid

SUBMITTED BY:
AMINA QADEER
Reg # 359607
DE-42 (C&SE)-A

Submission Date: 30/9/2022

Objectives:

Making us familiar with the emu8086 interface. Compiling assembly language on its GUI.

Related Topic/Chapter in theory class:

None

Hardware/Software required:

Hardware: PC
Software Tool: emu8086 v2.57

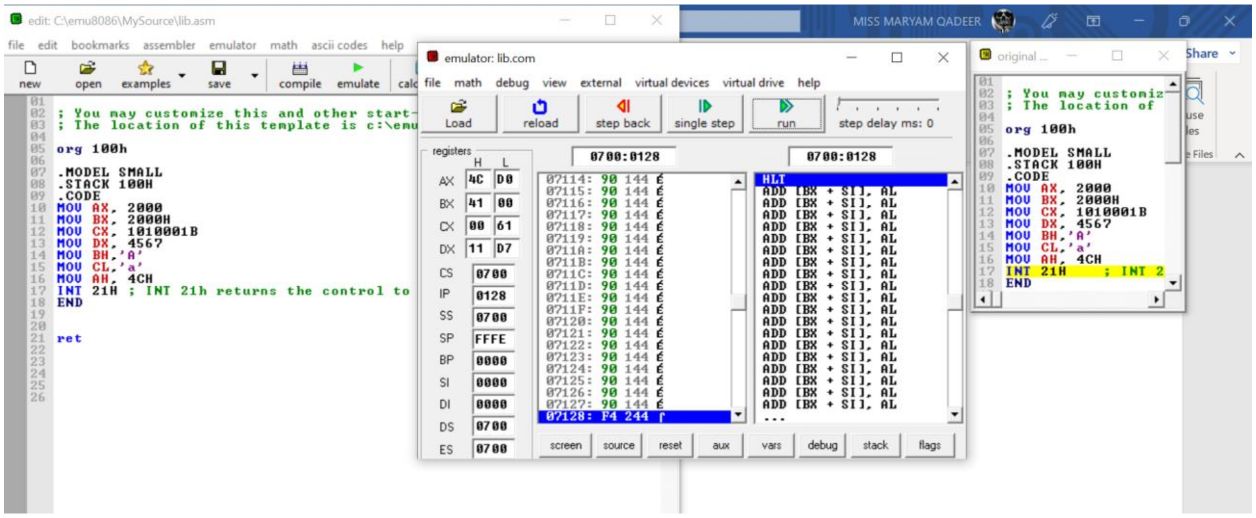
Tasks:

1) **Observe and write down the contents of registers AX, BX, CX, and DX after the complete code is run?**

Solution:

```
.MODEL SMALL
.STACK 100H
.CODE
MOV AX, 2000
MOV BX, 2000H
MOV CX, 1010001B
MOV DX, 4567
MOV BH,'A'
MOV CL,'a'
MOV AH, 4CH
INT 21H   ; INT 21h returns the control to DOS if AH=4CH
END
```

Output:



edit: C:\emu8086\MySource\lib.asm

```
; You may customize this and other start-
; The location of this template is c:\emu

org 100h

.MODEL SMALL
.STACK 100H
.CODE
MOV AX, 2000
MOV BX, 2000H
MOV CX, 1010001B
MOV DX, 4567
MOV BH,'A'
MOV CL,'a'
MOV AH, 4CH
INT 21H ; INT 21h returns the control to
END


ret
```

emulator: lib.com

	H	L
AX	4C	D0
BX	41	00
CX	00	61
DX	11	D7

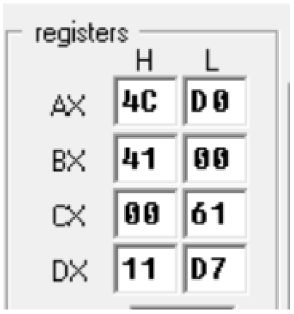
CS	0700
IP	0128
SS	0700
SP	FFFE
BP	0000
SI	0000
DI	0000
DS	0700
ES	0700

```
0700:0128
07114: 90 144 É
07115: 90 144 É
07116: 90 144 É
07117: 90 144 É
07118: 90 144 É
07119: 90 144 É
0711A: 90 144 É
0711B: 90 144 É
0711C: 90 144 É
0711D: 90 144 É
0711E: 90 144 É
0711F: 90 144 É
07120: 90 144 É
07121: 90 144 É
07122: 90 144 É
07123: 90 144 É
07124: 90 144 É
07125: 90 144 É
07126: 90 144 É
07127: 90 144 É
07128: F4 244 ſ
```

```
0700:0128
HLT
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
...
```

file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
screen source reset aux vars debug stack flags
MISS MARYAM QADEER
original ...
Share

Following values we get from data registers:



	H	L
AX	4C	D0
BX	41	00
CX	00	61
DX	11	D7

2) Do the contents of any register change as the code is run step by step? If yes, what change is observed and in which registers?

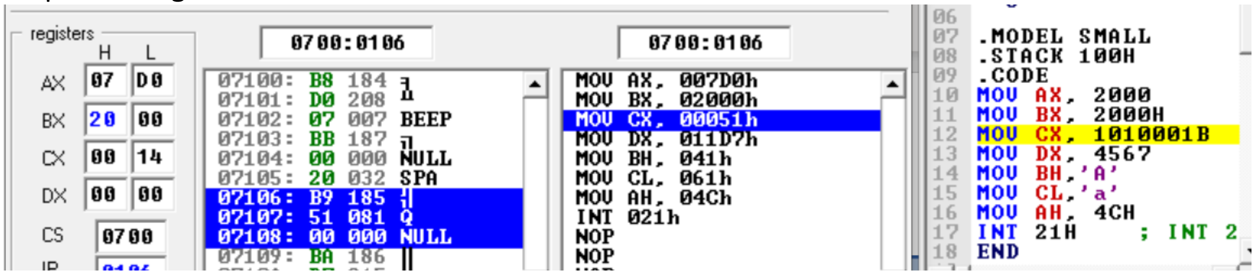
Solution:

.MODEL SMALL
.STACK 100H
.CODE
MOV AX, 2000
MOV BX, 2000H
MOV CX, 1010001B
MOV DX, 4567
MOV BH,'A'
MOV CL,'a'

MOV AH, 4CH
INT 21H   ; INT 21h returns the control to DOS if AH=4CH
END

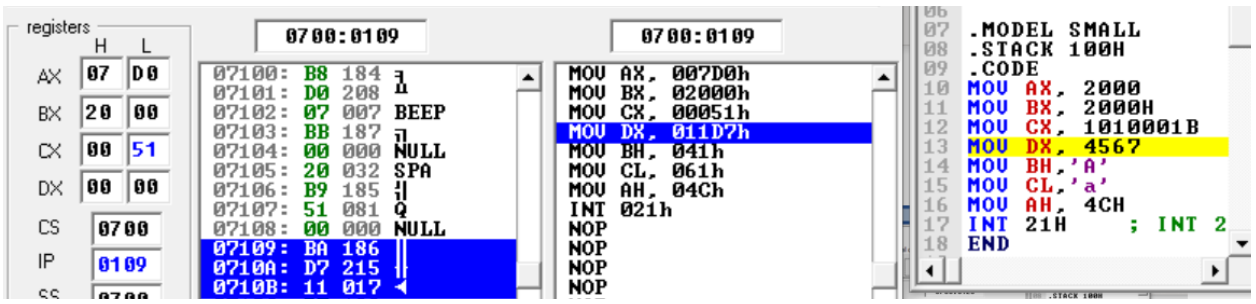
Output:

Step #1 change :



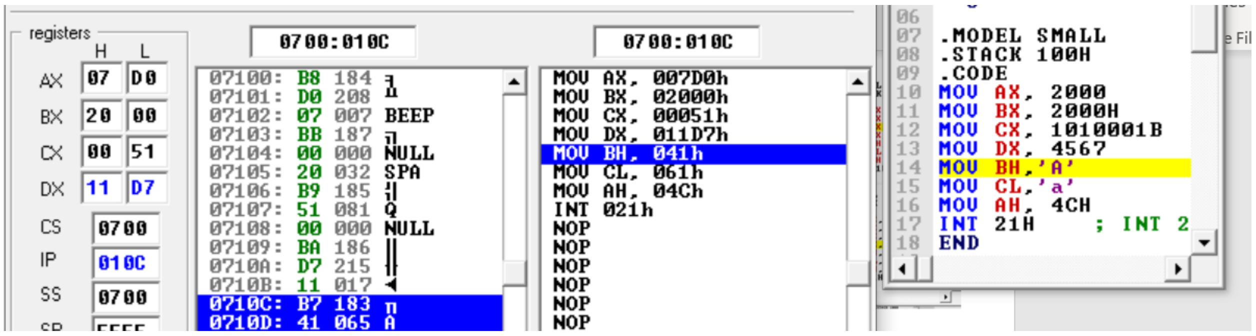
```
registers
     H    L
AX   07   D0
BX   20   00
CX   00   14
DX   00   00
CS   0700
IP   0106

0700:0106
07100: B8 184 ╕
07101: D0 208 ╨
07102: 07 007 BEEP
07103: BB 187 ╗
07104: 00 000 NULL
07105: 20 032 SPA
07106: B9 185 ╣
07107: 51 081 Q
07108: 00 000 NULL
07109: BA 186 ║

0700:0106
MOV  AX, 007D0h
MOV  BX, 02000h
MOV  CX, 00051h
MOV  DX, 011D7h
MOV  BH, 041h
MOV  CL, 061h
MOV  AH, 04Ch
INT  021h
NOP
NOP

.MODEL SMALL
.STACK 100H
.CODE
MOV AX, 2000
MOV BX, 2000H
MOV CX, 1010001B
MOV DX, 4567
MOV BH,'A'
MOV CL,'a'
MOV AH, 4CH
INT 21H      ; INT 2
END
```

Step #2 change :



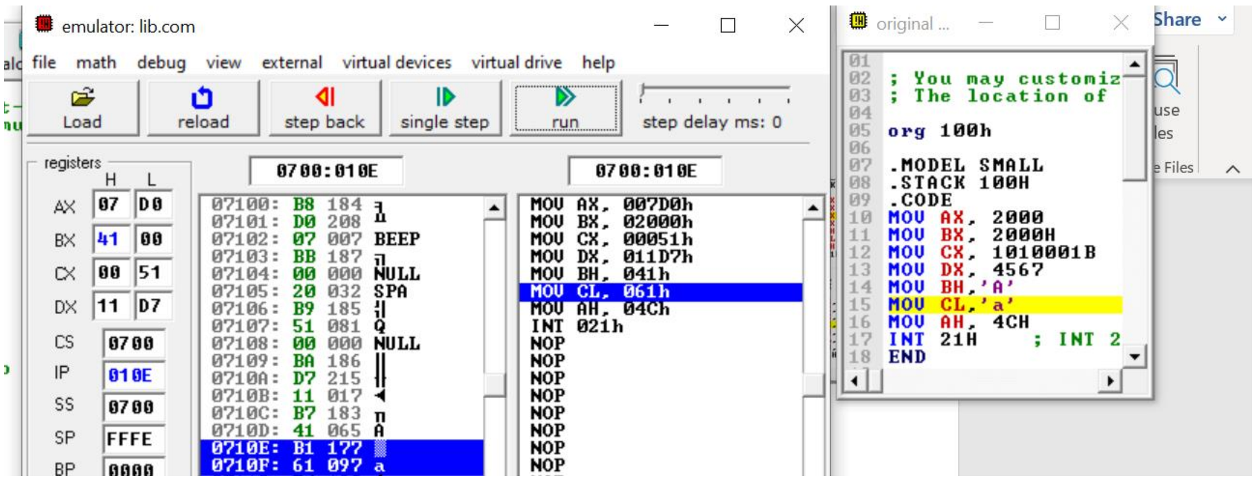
```
registers
     H    L
AX   07   D0
BX   20   00
CX   00   51
DX   00   00
CS   0700
IP   0109

0700:0109
07100: B8 184 ╕
07101: D0 208 ╨
07102: 07 007 BEEP
07103: BB 187 ╗
07104: 00 000 NULL
07105: 20 032 SPA
07106: B9 185 ╣
07107: 51 081 Q
07108: 00 000 NULL
07109: BA 186 ║
0710A: D7 215 ╫
0710B: 11 017 ◄

0700:0109
MOV  AX, 007D0h
MOV  BX, 02000h
MOV  CX, 00051h
MOV  DX, 011D7h
MOV  BH, 041h
MOV  CL, 061h
MOV  AH, 04Ch
INT  021h
NOP
NOP
NOP
NOP

.MODEL SMALL
.STACK 100H
.CODE
MOV AX, 2000
MOV BX, 2000H
MOV CX, 1010001B
MOV DX, 4567
MOV BH,'A'
MOV CL,'a'
MOV AH, 4CH
INT 21H      ; INT 2
END
```

Step #3 change :



```
registers
     H    L
AX   07   D0
BX   20   00
CX   00   51
DX   11   D7
CS   0700
IP   010C
SS   0700

0700:010C
07100: B8 184 ╕
07101: D0 208 ╨
07102: 07 007 BEEP
07103: BB 187 ╗
07104: 00 000 NULL
07105: 20 032 SPA
07106: B9 185 ╣
07107: 51 081 Q
07108: 00 000 NULL
07109: BA 186 ║
0710A: D7 215 ╫
0710B: 11 017 ◄
0710C: B7 183 ╖
0710D: 41 065 A

0700:010C
MOV  AX, 007D0h
MOV  BX, 02000h
MOV  CX, 00051h
MOV  DX, 011D7h
MOV  BH, 041h
MOV  CL, 061h
MOV  AH, 04Ch
INT  021h
NOP
NOP
NOP
NOP
NOP
NOP

.MODEL SMALL
.STACK 100H
.CODE
MOV AX, 2000
MOV BX, 2000H
MOV CX, 1010001B
MOV DX, 4567
MOV BH,'A'
MOV CL,'a'
MOV AH, 4CH
INT 21H      ; INT 2
END
```

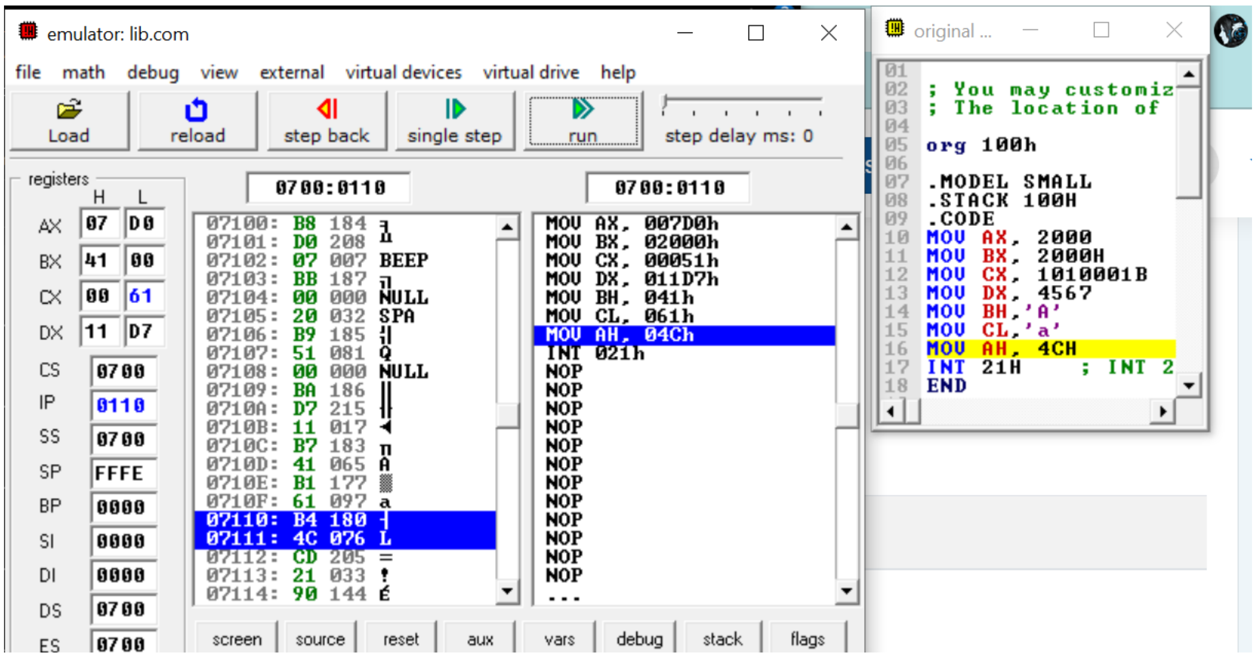
Step #4 change :



```
emulator: lib.com
registers   H  L
AX  07 D0
BX  41 00
CX  00 51
DX  11 D7
CS  0700
IP  010E
SS  0700
SP  FFFE
BP  0000

0700:010E                      0700:010E
07100: B8 184 ╕                MOV AX, 007D0h
07101: D0 208 ╨                MOV BX, 02000h
07102: 07 007 BEEP             MOV CX, 00051h
07103: BB 187 ╗                MOV DX, 011D7h
07104: 00 000 NULL             MOV BH, 041h
07105: 20 032 SPA              MOV CL, 061h
07106: B9 185 ╣                MOV AH, 04Ch
07107: 51 081 Q                INT 021h
07108: 00 000 NULL             NOP
07109: BA 186 ║                NOP
0710A: D7 215 ╫                NOP
0710B: 11 017 ◄                NOP
0710C: B7 183 ╖                NOP
0710D: 41 065 A                NOP
0710E: B1 177 ▒                NOP
0710F: 61 097 a                NOP
```

```
original ...
; You may customiz
; The location of

org 100h

.MODEL SMALL
.STACK 100H
.CODE
MOV AX, 2000
MOV BX, 2000H
MOV CX, 1010001B
MOV DX, 4567
MOV BH,'A'
MOV CL,'a'
MOV AH, 4CH
INT 21H      ; INT 2
END
```

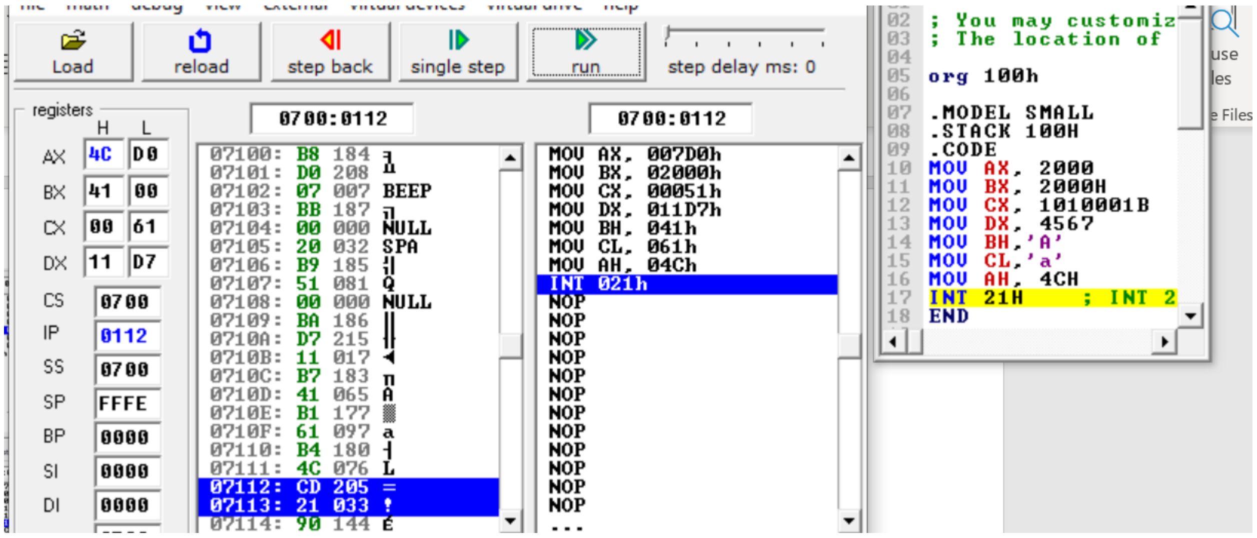
Step #5 change :



```
emulator: lib.com
registers   H  L
AX  07 D0
BX  41 00
CX  00 61
DX  11 D7
CS  0700
IP  0110
SS  0700
SP  FFFE
BP  0000
SI  0000
DI  0000
DS  0700
ES  0700

0700:0110                      0700:0110
07100: B8 184 ╕                MOV AX, 007D0h
07101: D0 208 ╨                MOV BX, 02000h
07102: 07 007 BEEP             MOV CX, 00051h
07103: BB 187 ╗                MOV DX, 011D7h
07104: 00 000 NULL             MOV BH, 041h
07105: 20 032 SPA              MOV CL, 061h
07106: B9 185 ╣                MOV AH, 04Ch
07107: 51 081 Q                INT 021h
07108: 00 000 NULL             NOP
07109: BA 186 ║                NOP
0710A: D7 215 ╫                NOP
0710B: 11 017 ◄                NOP
0710C: B7 183 ╖                NOP
0710D: 41 065 A                NOP
0710E: B1 177 ▒                NOP
0710F: 61 097 a                NOP
07110: B4 180 ┤                NOP
07111: 4C 076 L                NOP
07112: CD 205 ═                NOP
07113: 21 033 !                NOP
07114: 90 144 É                ...
```

```
original ...
; You may customiz
; The location of

org 100h

.MODEL SMALL
.STACK 100H
.CODE
MOV AX, 2000
MOV BX, 2000H
MOV CX, 1010001B
MOV DX, 4567
MOV BH,'A'
MOV CL,'a'
MOV AH, 4CH
INT 21H      ; INT 2
END
```

Step #6 change :



```
registers   H   L
AX         4C  D0
BX         41  00
CX         00  61
DX         11  D7
CS         0700
IP         0112
SS         0700
SP         FFFE
BP         0000
SI         0000
DI         0000
```

```
0700:0112
07100: B8 184 ╕
07101: D0 208 ╨
07102: 07 007 BEEP
07103: BB 187 ╗
07104: 00 000 NULL
07105: 20 032 SPA
07106: B9 185 ╣
07107: 51 081 Q
07108: 00 000 NULL
07109: BA 186 ║
0710A: D7 215 ╫
0710B: 11 017 ◄
0710C: B7 183 ╖
0710D: 41 065 A
0710E: B1 177 ▒
0710F: 61 097 a
07110: B4 180 ┤
07111: 4C 076 L
07112: CD 205 ═
07113: 21 033 !
07114: 90 144 É
```

```
0700:0112
MOV AX, 007D0h
MOV BX, 02000h
MOV CX, 00051h
MOV DX, 011D7h
MOV BH, 041h
MOV CL, 061h
MOV AH, 04Ch
INT 021h
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

```
; You may customiz
; The location of

org 100h

.MODEL SMALL
.STACK 100H
.CODE
MOV AX, 2000
MOV BX, 2000H
MOV CX, 1010001B
MOV DX, 4567
MOV BH,'A'
MOV CL,'a'
MOV AH, 4CH
INT 21H      ; INT 2
END
```

3)What happens if we replace 35H with just 35?

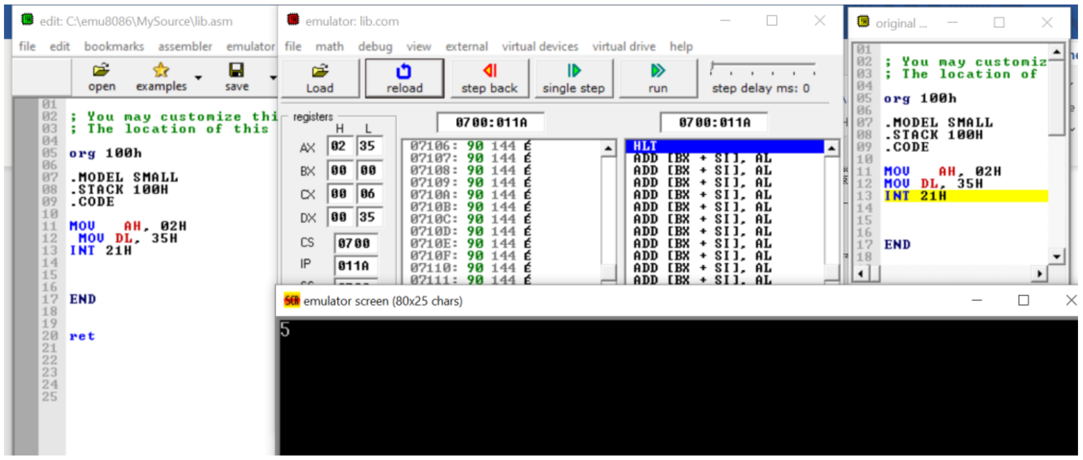
solution:

MOV AH, 02H

MOV DL, 35H

INT 21H

OUTPUT:



```
edit: C:\emu8086\MySource\lib.asm

; You may customize thi
; The location of this

org 100h

.MODEL SMALL
.STACK 100H
.CODE

MOV    AH, 02H
 MOV DL, 35H
INT 21H



END


ret
```

```
emulator: lib.com

registers   H   L
AX         02  35
BX         00  00
CX         00  06
DX         00  35
CS         0700
IP         011A
```

```
0700:011A
07106: 90 144 É
07107: 90 144 É
07108: 90 144 É
07109: 90 144 É
0710A: 90 144 É
0710B: 90 144 É
0710C: 90 144 É
0710D: 90 144 É
0710E: 90 144 É
0710F: 90 144 É
07110: 90 144 É
07111: 90 144 É
```

```
0700:011A
HLT
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
```

```
emulator screen (80x25 chars)
5
```

4)What register contains the ASCII code of the character read from the keyboard?

Solution:

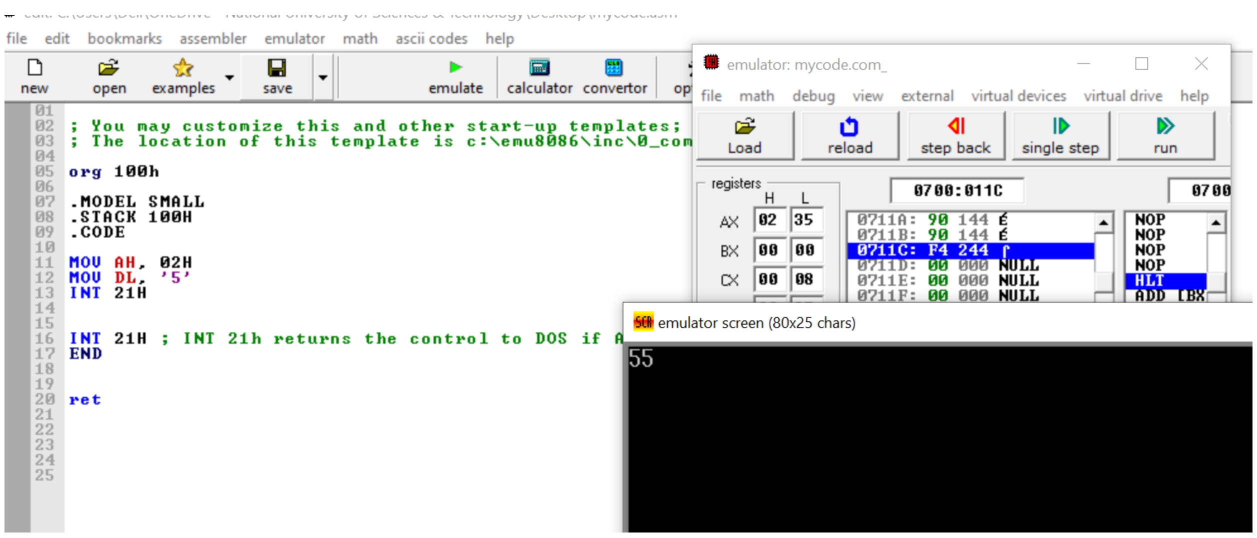
This code also displays 5:

MOV AH, 02H

MOV DL, ‘5’

INT 21H

Output

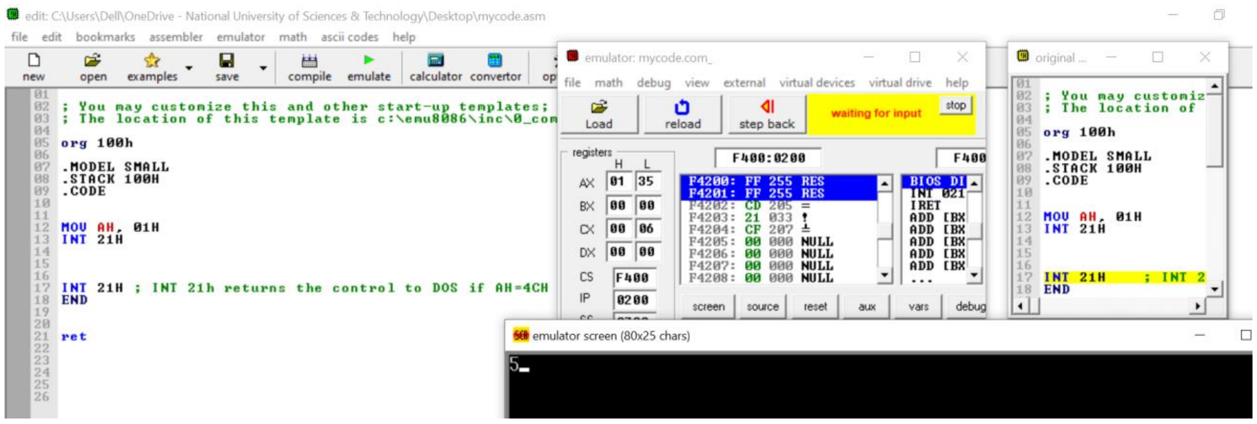


```
; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com

org 100h

.MODEL SMALL
.STACK 100H
.CODE

MOV AH, 02H
MOV DL, '5'
INT 21H


INT 21H ; INT 21h returns the control to DOS if A
END


ret
```

```
emulator: mycode.com_
registers    H   L
AX          02  35
BX          00  00
CX          00  08

0700:011C

0711A: 90 144 É        NOP
0711B: 90 144 É        NOP
0711C: F4 244 ⌠        NOP
0711D: 00 000 NULL     NOP
0711E: 00 000 NULL     HLT
0711F: 00 000 NULL     ADD [BX
```

```
emulator screen (80x25 chars)
55
```

Function 01h



```
edit: C:\Users\Dell\OneDrive - National University of Sciences & Technology\Desktop\mycode.asm

; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com

org 100h

.MODEL SMALL
.STACK 100H
.CODE

MOV AH, 01H
INT 21H


INT 21H ; INT 21h returns the control to DOS if AH=4CH
END


ret
```

```
emulator: mycode.com_
waiting for input
registers    H   L
AX          01  35
BX          00  00
CX          00  06
DX          00  00
CS          F400
IP          0200

F400:0200

F4200: FF 255 RES      BIOS DI
F4201: FF 255 RES      INT 021
F4202: CD 205 =        IRET
F4203: 21 033 !        ADD [BX
F4204: CF 207 ⊥        ADD [BX
F4205: 00 000 NULL     ADD [BX
F4206: 00 000 NULL     ADD [BX
F4207: 00 000 NULL     ADD [BX
F4208: 00 000 NULL     ...
```

```
emulator screen (80x25 chars)
5_
```

5)Does using any other register in place of DX (in function 09h) give the same result? If not, what can be the reason?

Solution:

DATA
MESSAGE **DB** ‘This is the Message to be displayed’, ‘$’

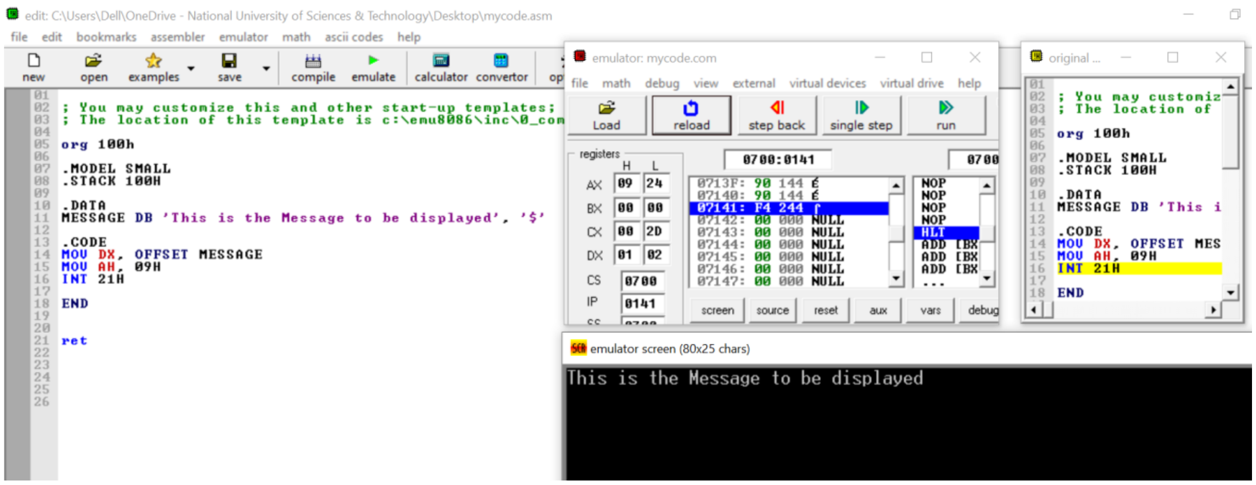
.CODE
MOV DX, OFFSET MESSAGE
MOV AH, 09H
INT 21H

Or

.DATA
MESSAGE **DB** ‘This is the Message to be displayed’, ‘$’

.CODE
LEA DX, MESSAGE
MOV AH, 09H
INT 21H

OUTPUT:

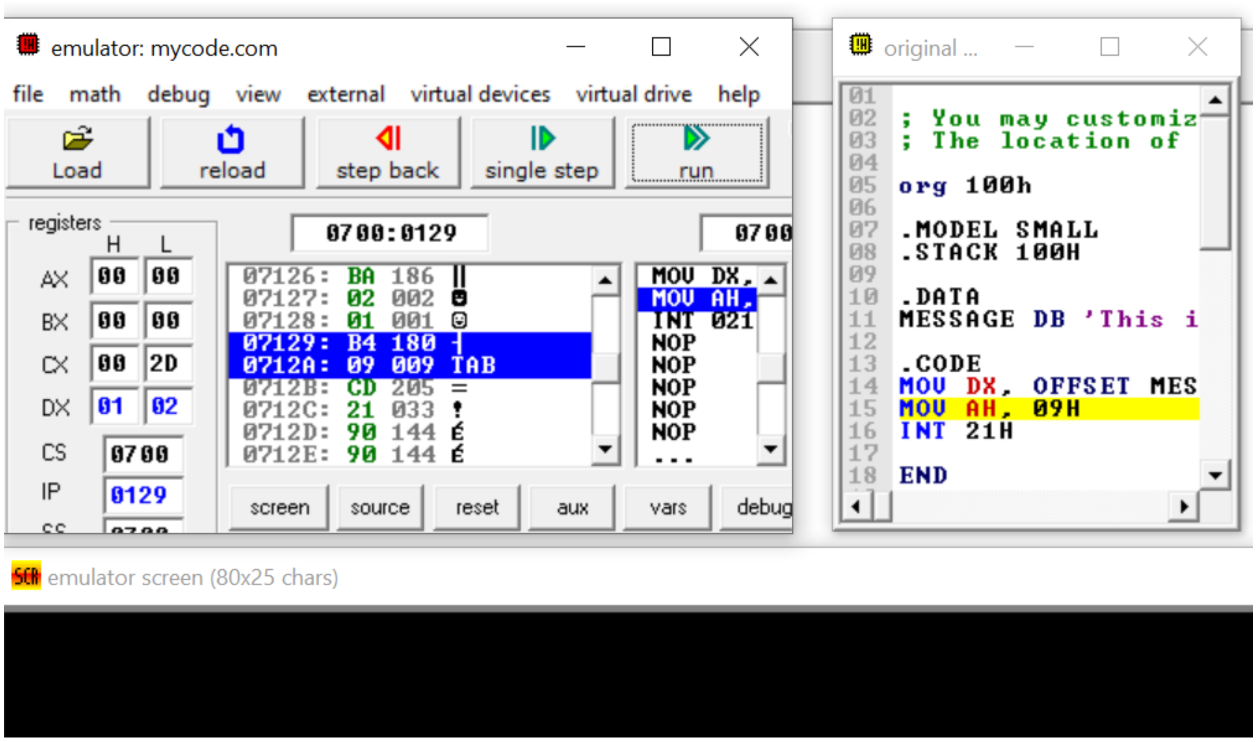

edit: C:\Users\Dell\OneDrive - National University of Sciences & Technology\Desktop\mycode.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor
emulator: mycode.com
file math debug view external virtual devices virtual drive help
Load reload step back single step run
registers
screen source reset aux vars debug
original ...
emulator screen (80x25 chars)

```
; You may customize this and other start-up templates;
; The location of this template is c:\emu8086\inc\0_com

org 100h

.MODEL SMALL
.STACK 100H

.DATA
MESSAGE DB 'This is the Message to be displayed', '$'

.CODE
MOV DX, OFFSET MESSAGE
MOV AH, 09H
INT 21H

END


ret
```

```
      H  L
AX   09 24
BX   00 00
CX   00 2D
DX   01 02
CS   0700
IP   0141

0700:0141                 0700

0713F: 90 144 É           NOP
07140: 90 144 É           NOP
07141: F4 244 ſ           NOP
07142: 00 000 NULL        NOP
07143: 00 000 NULL        HLT
07144: 00 000 NULL        ADD [BX
07145: 00 000 NULL        ADD [BX
07146: 00 000 NULL        ADD [BX
07147: 00 000 NULL        ...
```

```
; You may customiz
; The location of

org 100h

.MODEL SMALL
.STACK 100H

.DATA
MESSAGE DB 'This i

.CODE
MOV DX, OFFSET MES
MOV AH, 09H
INT 21H

END
```

```
This is the Message to be displayed
```


emulator: mycode.com
file math debug view external virtual devices virtual drive help
Load
reload
step back
single step
run
registers
screen source reset aux vars debug
original ...
emulator screen (80x25 chars)

```
      H   L
AX   00  00
BX   00  00
CX   00  2D
DX   01  02
CS   0700
IP   0129
```

```
0700:0129

07126:  BA 186 ║
07127:  02 002 ☻
07128:  01 001 ☺
07129:  B4 180 ┤
0712A:  09 009 TAB
0712B:  CD 205 ═
0712C:  21 033 !
0712D:  90 144 É
0712E:  90 144 É
```

```
MOV  DX,
MOV  AH,
INT  021
NOP
NOP
NOP
NOP
NOP
...
```

```

; You may customiz
; The location of

org 100h

.MODEL SMALL
.STACK 100H

.DATA
MESSAGE DB 'This i

.CODE
MOV DX, OFFSET MES
MOV AH, 09H
INT 21H

END
```

Upon analyzing the code by single step run. DOS first declares variables in the data segment. Just when the compiler reads a function called "MESSAGE DB "it moves to the code segment. Adding offset required to reach the desired location of memory, where we have our label defined with string," 'This is the Message to be displayed, '$'. Afterward, DOS moves to function call 09h for it displays the string characters addressed by DX to the screen. No other data register updates value because each has its own defined special purpose on a particular function call

F unction	**V alue of**	**Ou tput in**	**Functionality**
0 1h	0 1h	A L	Reads a character from keyboard, stores it in AL and
0 2h	0 2h	Sc reen	Display the content of register DL on screen in ASCII
0 9h	0 9h	Sc reen	Display the string characters addressed by DX to the screen
0 Ah	0 Ah	Of fset in D	Read a string of characters from keyboard.

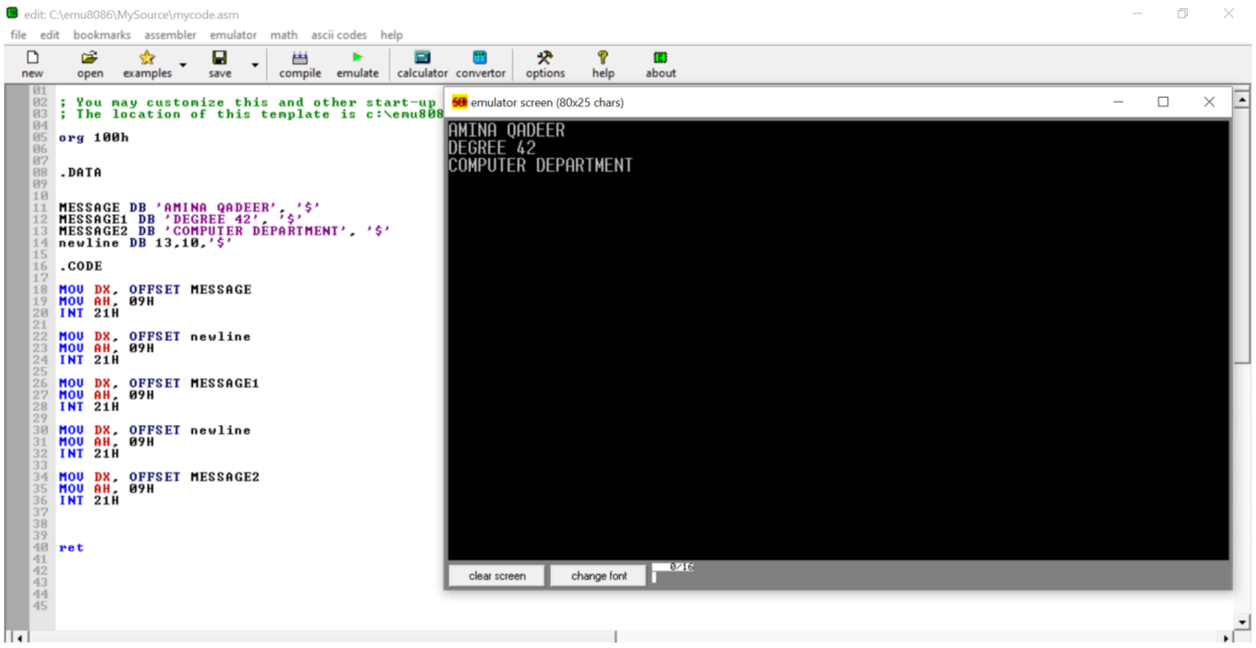
6)Define three strings containing your name, degree and department and display them on screen. The strings should be displayed on three different lines. Paste your code and the screenshot of output.

SOULTION:

```
; You may customize this and other start-up templates;

; The location of this template is c:\emu8086\inc\0_com_template.txt



org 100h



.DATA



MESSAGE DB 'AMINA QADEER', '$'

MESSAGE1 DB 'DEGREE 42', '$'

MESSAGE2 DB 'COMPUTER DEPARTMENT', '$'

newline DB 13,10,'$'

.CODE

MOV DX, OFFSET MESSAGE

MOV AH, 09H
```

```
INT 21H

MOV DX, OFFSET newline
MOV AH, 09H
INT 21H

MOV DX, OFFSET MESSAGE1
MOV AH, 09H
INT 21H

MOV DX, OFFSET newline
MOV AH, 09H
INT 21H

MOV DX, OFFSET MESSAGE2
MOV AH, 09H
INT 21H
ret
END
```

OUTPUT:



edit: C:\emu8086\MySource\mycode.asm
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
emulator screen (80x25 chars)
clear screen change font

```
; You may customize this and other start-up
; The location of this template is c:\emu808

org 100h


.DATA


MESSAGE DB 'AMINA QADEER', '$'
MESSAGE1 DB 'DEGREE 42', '$'
MESSAGE2 DB 'COMPUTER DEPARTMENT', '$'
newline DB 13,10,'$'

.CODE

MOV DX, OFFSET MESSAGE
MOV AH, 09H
INT 21H

MOV DX, OFFSET newline
MOV AH, 09H
INT 21H

MOV DX, OFFSET MESSAGE1
MOV AH, 09H
INT 21H

MOV DX, OFFSET newline
MOV AH, 09H
INT 21H

MOV DX, OFFSET MESSAGE2
MOV AH, 09H
INT 21H



ret
```

```
AMINA QADEER
DEGREE 42
COMPUTER DEPARTMENT
```

Conclusion:

1) **The 8086 is a 16-bit microprocessor**. The term "16-bit" means that its arithmetic logic unit, internal registers, and most of its instructions are designed to work with 16-bit binary words. 2) The 8086 has a 16-bit data bus, so it can read data from or write data to memory and ports either 16 bits or 8 bits at a time.
2) I saved my .asm file and constructed an executable file from it by compiling my assembly code.
3) Now I know where data registers are located on the interface.
4) Functions of special purpose and general-purpose registers.