# Laboratory Manual

## Microprocessor & Assembly Language (CS-330)

**5th Semester**


Name: Munib-ul-hassan

Roll No: _____

Section: 2019-CS-037 "A"

Batch: 2019

Department of Computer Science & Information Technology
Sir Syed University of Engineering & Technology
University Road, Karachi
https://www.ssuet.edu.pk

# List of Laboratory Experiments

| Lab No. | Date | TITLES | Page No. | Signature |
|---|---|---|---|---|
| 1. | 23/2 | To familiarize the students with the 8086/88 Assembler, the Central Processing Uit (CPU), and refresh the hexadecimal number systems. | 1-5 | |
| 2. | 23/2 | To introduce System commands using DEBUG programming utility (at command prompt using PC). | 6-11 | |
| 3. | 9/3 | To introduce Program control commands using DEBUG programming utility (at command prompt using PC). | 12-15 | |
| 4. | 9/3 | To learn how to create and assemble an executable Assembly Language Programming using Assembler and Linker utilities. | 16-17 | |
| 5. | | Develop understanding to use different BIOS and DOS Interrupt Services using MASM/TASM utility. | 18-19 | |
| 6. | | Implementation of the Program Flow Control using "JMP" and "LOOP" instructions in Assembly language. | 20-24 | |
| 7. | | Implementation of the basic arithmetic operations using Assembly instructions, such as Half-Adder, Full Adder, Half-Subtractor, Full Subtractor, Multiplier and Divisor. | 25-27 | |
| 8. | | Implementation of Logical instructions using Assembly language. | 28-30 | |
| 9. | | Implementation of ASCII and BCD arithmetic instructions with the help of Logical operation using 8086/88 Assembly language. | 31-33 | |
| 10. | | Implementation of procedures using CALL instruction in Assembly language. | 34-35 | |
| 11. | | To perform computation by using Assembly Language Programming.<br>• To generate Fibonacci series<br>• To generate Factorial of a number | 36-37 | |
| 12. | | To implement the string manipulation instructions provided in the 8086/88 Assembly language. | 38-39 | |

**c) Conversion from decimal to hexadecimal**

- Divide the given number by sixteen, keeping track of the remainder
- First remainder is bit 0 with the weight 1 (LSB, least-Significant Bit)
- Second remainder is bit 1
- Do the same till the last digit
-

**d) Conversion from binary to decimal**

- Multiply each bit by $2^n$, the "weight" of the bit
- n is the sequence number of bits from LSB which is the right most bit
- Add the results

**e) Conversion from binary to octal**

- Group bits in threes, from right to left
- Convert to octal digits

**f) Conversion from binary to hexadecimal**

- Group bits in four, from right to left
- Convert to hexadecimal digits

**g) Conversion from octal hexadecimal**

- Use binary as an intermediary

**h) Conversion from hexadecimal to octal**

- Use binary as an intermediary

**A Quick Example**

Following is a quick example of such conversions.

$$25_{10} = \underset{3}{\underset{1}{1}}\underset{1}{\underset{9}{1001}}_2 = 31_8 = 19_{16}$$

## V. Activity:

Convert and fill the table give below. Use your roll number to fill the last (blank) row:

| Decimal | Binary | Octal | Hexadecimal |
|---|---|---|---|
| 33 | 100001 | 41 | 21 |
| 117 | 1110101 | 165 | 75 |
| 451 ~~#10000##~~ | 11100011 | 703 | 1C3 |
| 431 | 110101111 | 657 | 1AF |
| | | | |

# LAB 2

- Display the memory contents starting from memory location 0100h

```
Z:\>C:

C:\>debug
-d 0100
073F:0100  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
073F:0110  00 00 8F E9 00 F0 87 74-B2 00 8C 00 2E 07 2E 07   .......t........
073F:0120  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
073F:0130  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
073F:0140  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
073F:0150  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
073F:0160  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
073F:0170  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-
```

Display the memory contents of first 10 bytes starting from memory location 0100h

```
-d 0100 0110
073F:0100  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
073F:0110  00                                                .
-
```

- Fill the memory location starting from 0200h with your name and 3-digit Roll #

```
-F 0200 "Munib ul hassan 037"
-d 0200
073F:0200  4D 75 6E 69 62 20 75 6C-20 68 61 73 73 61 6E 20   Munib ul hassan
073F:0210  30 33 37 4D 75 6E 69 62-20 75 6C 20 68 61 73 73   037Munib ul hass
073F:0220  61 6E 20 30 33 37 4D 75-6E 69 62 20 75 6C 20 68   an 037Munib ul h
073F:0230  61 73 73 61 6E 20 30 33-37 4D 75 6E 69 62 20 75   assan 037Munib u
073F:0240  6C 20 68 61 73 73 61 6E-20 30 33 37 4D 75 6E 69   l hassan 037Muni
073F:0250  62 20 75 6C 20 68 61 73-73 61 6E 20 30 33 37 4D   b ul hassan 037M
073F:0260  75 6E 69 62 20 75 6C 20-68 61 73 73 61 6E 20 30   unib ul hassan 0
073F:0270  33 37 4D 75 6E 69 62 20-75 6C 20 68 61 73 73 61   37Munib ul hassa
```

- Note the last offset (memory) value of the last digit of your roll #

---

- Move the block of memory where you filled your name and Roll # to the memory location starting from 0400h

```
-M 0200 0230 0400
-d 0400
073F:0400  4D 75 6E 69 62 20 75 6C-20 68 61 73 73 61 6E 20   Munib ul hassan
073F:0410  30 33 37 4D 75 6E 69 62-20 75 6C 20 68 61 73 73   037Munib ul hass
073F:0420  61 6E 20 30 33 37 4D 75-6E 69 62 20 75 6C 20 68   an 037Munib ul h
073F:0430  61 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   a...............
073F:0440  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
073F:0450  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
073F:0460  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
073F:0470  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
```

Compare the two memory

blocks (one starting from 0100h & the other from 0400h)

```
-c 0100 0120 0400
073F:011C 34 00 073F:011C
073F:011E 2E 00 073F:011E
073F:011F 07 00 073F:011F
-
```

- Edit 2018-CS- to the memory block starting from 0400h before your Roll #

```
-E 0411 "CS-19-"
-D 0400
073F:0400 4D 75 6E 69 62 20 75 6C-20 68 61 73 73 61 6E 20  Munib ul hassan
073F:0410 30 43 53 2D 31 39 2D 62-20 75 6C 20 68 61 73 73  0CS-19-b ul hass
073F:0420 61 6E 20 30 33 37 4D 75-6E 69 62 20 75 6C 20 68  an 037Munib ul h
073F:0430 61 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  a...............
073F:0440 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
073F:0450 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
073F:0460 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
073F:0470 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ................
-
```

- Compare the two memory blocks (one starting from 0100h & the other from 0400h)

```
-C 200 230 400
073F:0211 33 43 073F:0411
073F:0212 37 53 073F:0412
073F:0213 4D 2D 073F:0413
073F:0214 75 31 073F:0414
073F:0215 6E 39 073F:0415
073F:0216 69 2D 073F:0416
```

- Calculate the sum and difference of 75 and 34 using H command

```
-H 75 34
00A9 0041
```

## Activity- 2:

Assemble and Unassemble the following code and fill the following table .

| Logical Address | Opcode | Assembly code | Comments |
|---|---|---|---|
| 0735;0100 | B80002 | MOV AX, 0200H | ; store "0200" in AX |
| 0735:0103 | BB0004 | MOV BX, 0400H | ; store "0400" in bx register |
| 0735:0106 | 01 DB | ADD AX, BX | ; store the sum value of AX and BX in AX register |

## Activity- 3:

Give answers to the following questions after unassemble the code:

*Why the values **0200H** and **0400H** are written as **0002** and **0004**, in the opcode, respectively?*

*Write the number of bytes and the value of IP register taken by each instruction.*

Ans's

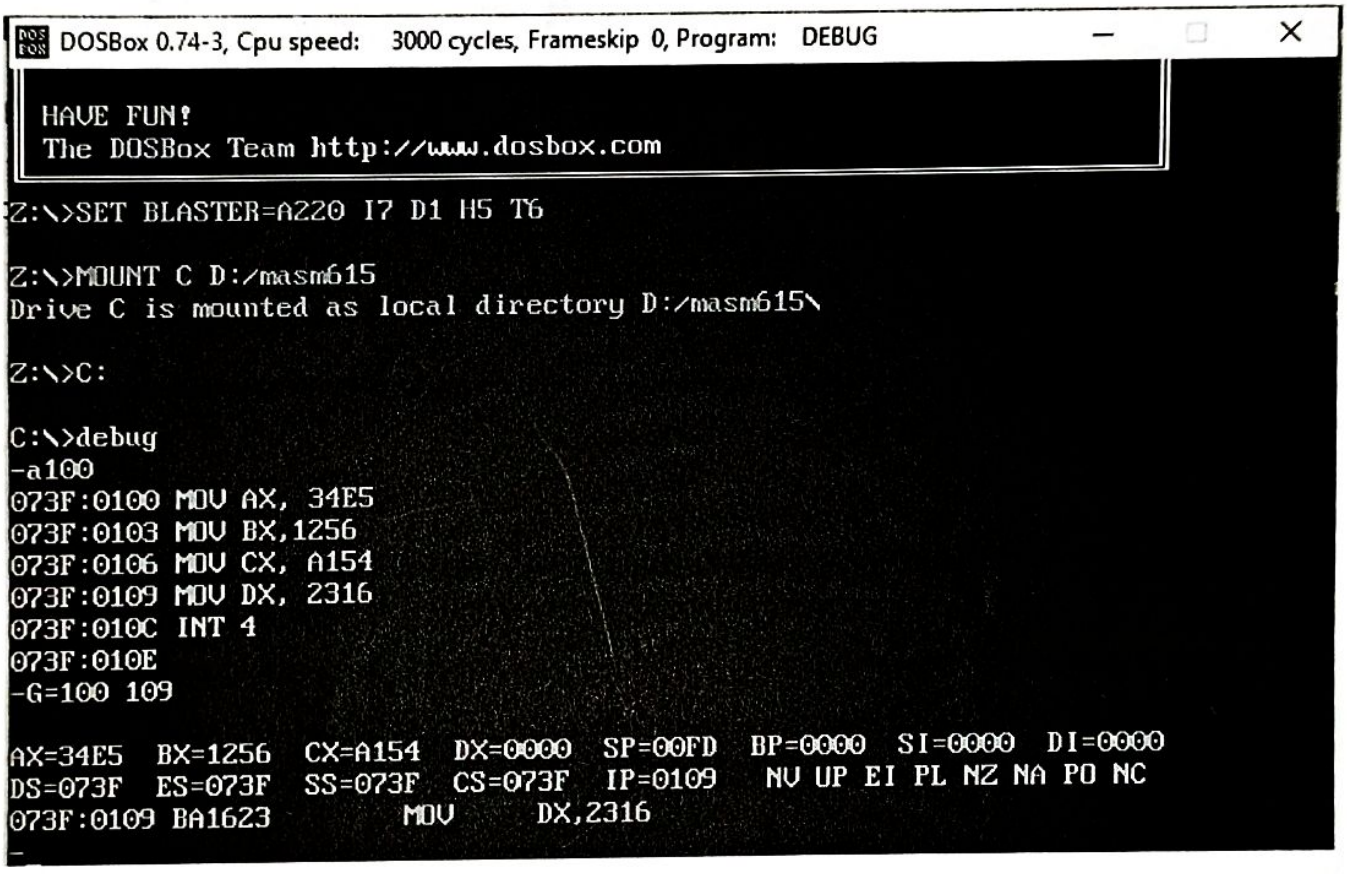
The value of 0200 and 0400 are written in 0002 and 0004 because of the byte system (Little Endian) in which the least significant byte (the "little end") of the data is placed at the byte with the lowest address.

| Instructions | no of bytes | IP address |
|---|---|---|
| | | 0100 |
| MOV AX 0200H | 3 | 0103 |
| MOVE BX 400H | 3 | 0106 |
| ~~MOVE~~ ~~AD~~ AX,BX<br>ADD | 3 | |

# LAB 03

## TASK- 1:

Assemble a program using DEBUG programming utility to move the decimal values in registers as given below AX = $5431_{10}$ BX = $9321_{10}$ CX = $45032_{10}$ DX = $23102_{10}$



# TASK- 2:

Apply the program control command **T** to execute the code. Capture the screen. Write down and analyze the values of each registers including IP and Flag registers.

```
073F:0109 BA1623        MOV     DX,2316
-T 4

AX=34E5  BX=1256  CX=A154  DX=2316  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=010C   NV UP EI PL NZ NA PO NC
073F:010C CD04          INT     04

AX=34E5  BX=1256  CX=A154  DX=2316  SP=00F7  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=0070  IP=0008   NV UP DI PL NZ NA PO NC
0070:0008 FE38          ???     [BX+SI]                         DS:1256=00

AX=34E5  BX=1256  CX=A154  DX=2316  SP=00F7  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=0070  IP=000C   NV UP DI PL NZ NA PO NC
0070:000C CF            IRET

AX=34E5  BX=1256  CX=A154  DX=2316  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=010E   NV UP EI PL NZ NA PO NC
073F:010E AE            SCASB
-
```

# TASK- 3:

Apply the program control command **G** to execute the code. Capture the screen. Write down and analyze the values of each registers including IP and Flag registers.

```
-G 100 100

AX=0000  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0100   NV UP EI PL NZ NA PO NC
073F:0100 CC            INT     3
-G 103 103

AX=0000  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0100   NV UP EI PL NZ NA PO NC
073F:0100 CC            INT     3
-G106 106

AX=0000  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0100   NV UP EI PL NZ NA PO NC
073F:0100 CC            INT     3
-G 109 109

AX=0000  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0100   NV UP EI PL NZ NA PO NC
073F:0100 CC            INT     3
-
```

# LAB 04

**TASK- 1:** Write, run and analyze a program that adds 5 bytes of data (given below). Use 5 different byte variables to store the data. Save the result in a byte variable named RESULT. **25h, 12h, 15h, 1Fh, 2Bh**

```
Title lab 4 activity 1
.model small
.stack 100h
.data

A1 DB 25H
A2 DB 12H
A3 DB 15H
A4 DB 1FH
A5 DB 2BH

RESULT DB ?

.code

MAIN PROC

; initialize DS
MOV AX,@DATA
MOV DS,AX
;add the numbers

mov al,A1
mov al,A2
mov al,A3
mov al,A4
mov al,A5

mov RESULT,al

mov ah,4Ch
int 21h

MAIN ENDP

END MAIN
```

Munib

CS19-037

```
Z:\>mount C d:/masm615
Drive C is mounted as local directory d:/masm615\

Z:\>C:

C:\>degug
Illegal command: degug.

C:\>debugger
Illegal command: debugger.

C:\>Debug
-q

C:\>cd BIN

C:\BIN>ls
Illegal command: ls.

C:\BIN>debug lab4a1.eze
```

```
C:\Windows\System32\cmd.exe
(c) 2020 Microsoft Corporation. All rights reserved.

D:\>cd masm615

D:\masm615>cd Bin

D:\masm615\BIN>masm Lab4a1.asm
Microsoft (R) MASM Compatibility Driver
Copyright (C) Microsoft Corp 1993.  All rights reserved.

 Invoking: ML.EXE /I. /Zm /c /Ta Lab4a1.asm

Microsoft (R) Macro Assembler Version 6.14.8444
Copyright (C) Microsoft Corp 1981-1997.  All rights reserved.

 Assembling: Lab4a1.asm
MASM : fatal error A1000: cannot open file : Lab4a1.asm

D:\masm615\BIN>link lab4a1.obg

Microsoft (R) Segmented Executable Linker  Version 5.60.339 Dec  5 1994
Copyright (C) Microsoft Corp 1984-1993.  All rights reserved.

Run File [lab4a1.exe]:
List File [nul.map]:
Libraries [.lib]:
Definitions File [nul.def]:
```

**TASK- 2:** Write, run and analyze a program that adds 5 bytes of data (given below). Use an array of 6 bytes to store the given 5 bytes of data and save the result in the last byte of array. **25h, 12h, 15h, 1Fh, 2Bh**

LAB4a2 - Notepad

File Edit Format View Help

```
Title lab 4 activity 2
.model small
.stack 100h
.data

Array1 DB 25H,12H,15H,1FH,2BH, ?

.CODE
MAIN PROC

MOV AX,@DATA
MOV DS,AX

MOV SI, offset Array1
MOV AL, [SI]
ADD AL, [SI+1]
ADD AL, [SI+2]
ADD AL, [SI+3]
ADD AL, [SI+4]
MOV [SI+5], AL

MOV AX,4C00H
INT 21H

MAIN ENDP
END MAIN
```

```
C:\>cd masm615

C:\masm615>cd bin

C:\masm615\BIN>masm LAB4a2.asm
Microsoft (R) MASM Compatibility Driver
Copyright (C) Microsoft Corp 1993.  All rights

 Invoking: ML.EXE /I. /Zm /c /Ta LAB4a2.asm

Microsoft (R) Macro Assembler Version 6.14.8444
Copyright (C) Microsoft Corp 1981-1997.  All ri

 Assembling: LAB4a2.asm

C:\masm615\BIN>link LAB4a2.obj

Microsoft (R) Segmented Executable Linker  Versi
Copyright (C) Microsoft Corp 1984-1993.  All rig

Run File [LAB4a2.exe]:
List File [nul.map]:
Libraries [.lib]:
Definitions File [nul.def]:

C:\masm615\BIN>
```

```
HAVE FUN!
The DOSBox Team http://www.dosb

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c c:\masm615
Drive C is mounted as local direc

Z:\>c:

C:\>cd bin

C:\BIN>LAB4a2

C:\BIN>debug LAB4a2.exe
-
```

Munib

CS19-037

13/04/21

# LAB 05

**ACTIVITY 01**

lab5A1.asm - Notepad

File Edit Format View Help

```
Title lab  5 activity 1
.MODEL small
.STACK 100h
.DATA
        STRING DB 'We are the students of computer Sience department$'
.CODE
  Main PROC
        MOV AX, @DATA
        MOV DS, AX

        MOV DX,OFFSET String

        MOV AH,9H
        INT 21H

        MOV AH,4CH
        INT 21H

   MAIN ENDP
END MAIN
```

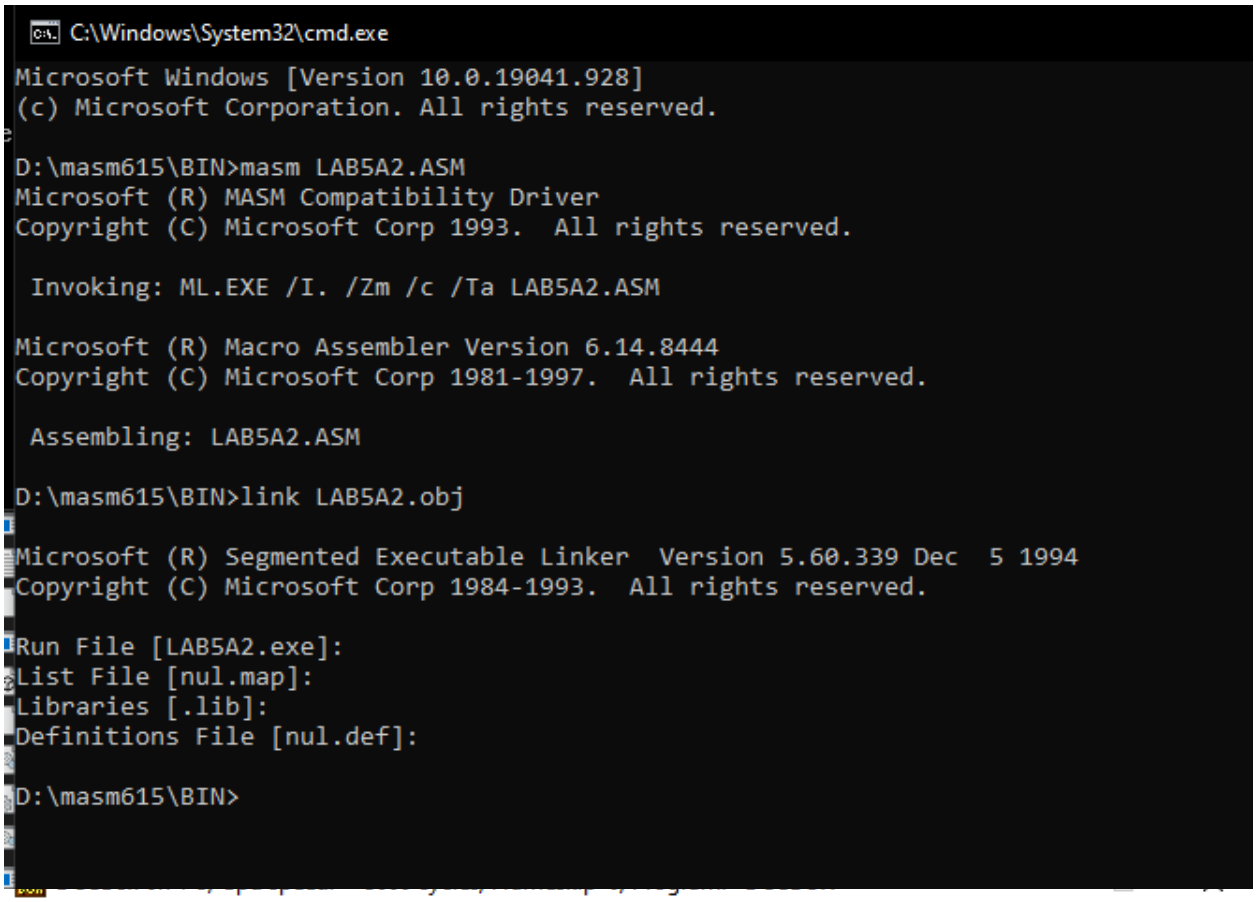
```
C:\BIN>lab5a1.exe
We are the students of computer Sience department
```

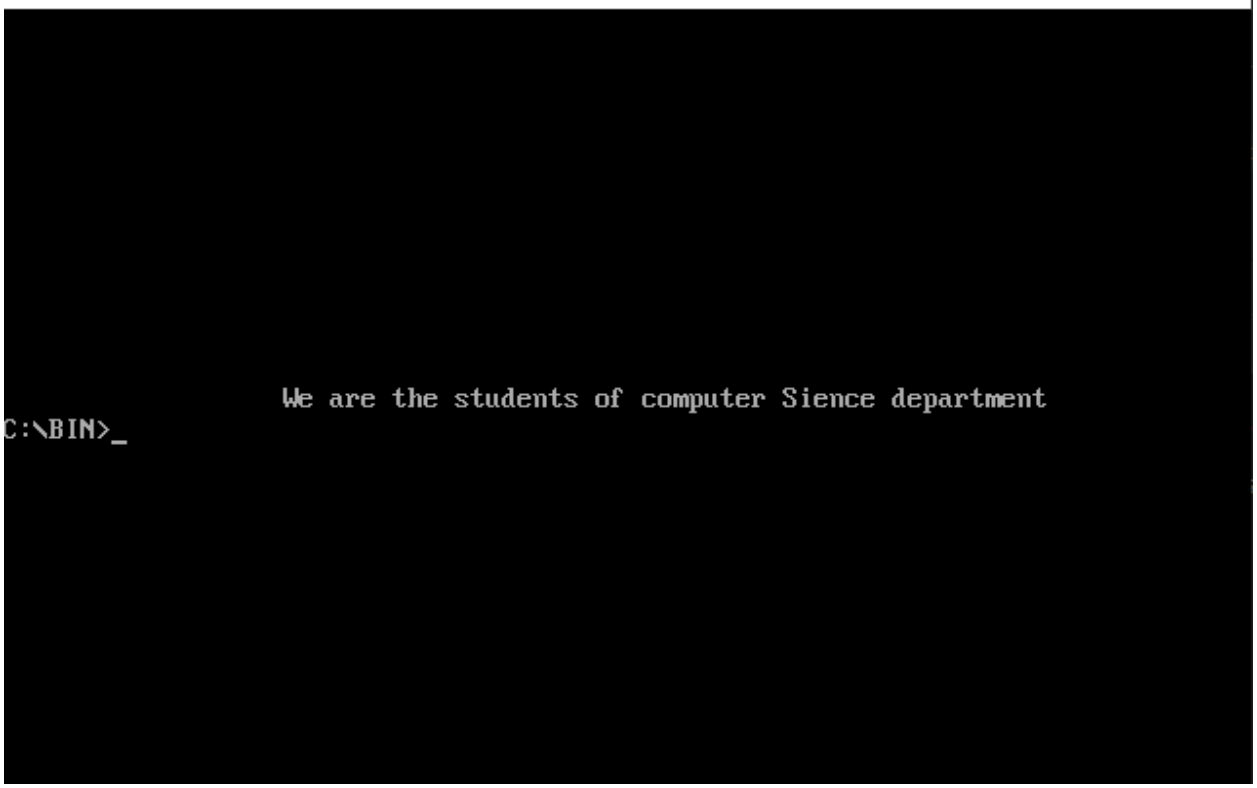
**ACTIVITY 2**

LAB5A2.ASM - Notepad

File Edit Format View Help

```
Title lab  5 activity 2
.MODEL small
.STACK 100h
.DATA
        STRING DB 'We are the students of computer Sience department$'
.CODE
  Main PROC
        MOV AX, @DATA
        MOV DS, AX


        MOV AH,6H
        MOV AL,0H
        MOV BH,7H
        MOV CX,0H
        MOV DX,184FH
        INT 10H

        MOV AH,2
        MOV BH,0H
        MOV DL,12H
        MOV DH,0CH
        INT 10H

        MOV DX,OFFSET String

        MOV AH,9H
        INT 21H

        MOV AH,4CH
        INT 21H

   MAIN ENDP
END MAIN
```


C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19041.928]
(c) Microsoft Corporation. All rights reserved.
D:\masm615\BIN>masm LAB5A2.ASM
Microsoft (R) MASM Compatibility Driver
Copyright (C) Microsoft Corp 1993. All rights reserved.
Invoking: ML.EXE /I. /Zm /c /Ta LAB5A2.ASM
Microsoft (R) Macro Assembler Version 6.14.8444
Copyright (C) Microsoft Corp 1981-1997. All rights reserved.
Assembling: LAB5A2.ASM
D:\masm615\BIN>link LAB5A2.obj
Microsoft (R) Segmented Executable Linker Version 5.60.339 Dec 5 1994
Copyright (C) Microsoft Corp 1984-1993. All rights reserved.
Run File [LAB5A2.exe]:
List File [nul.map]:
Libraries [.lib]:
Definitions File [nul.def]:
D:\masm615\BIN>


We are the students of computer Sience department
C:\BIN>_

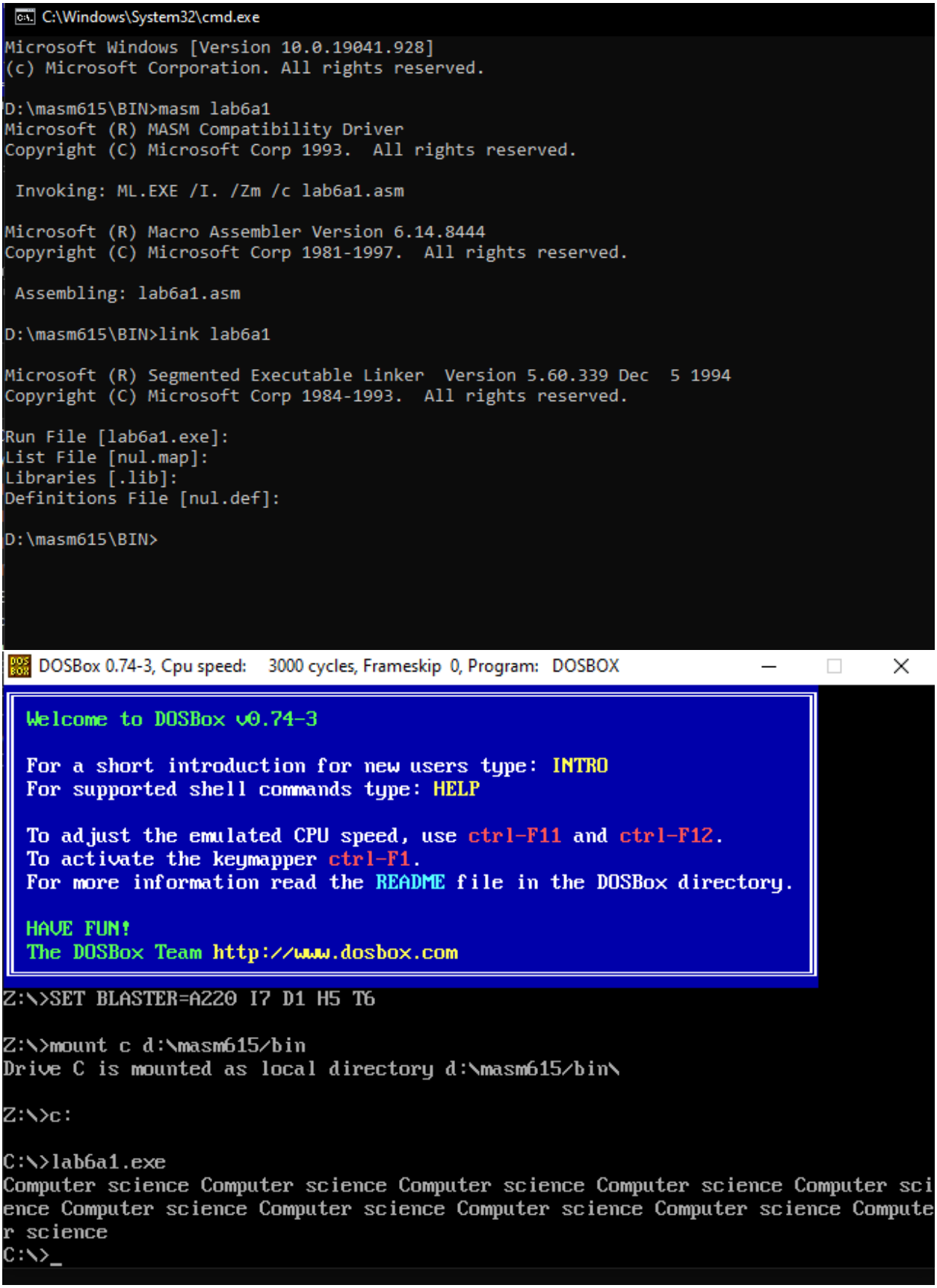
# LAB 6

**ACTIVITY 01**

lab6a1.asm - Notepad

File Edit Format View Help

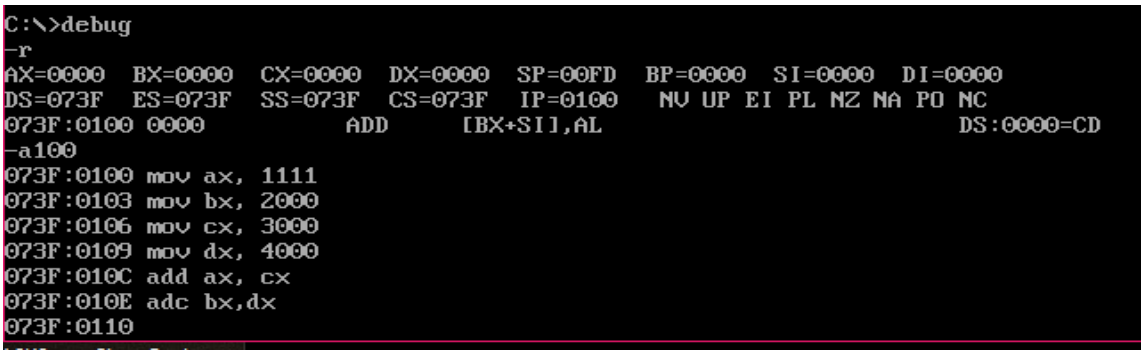
```
Title lab 6 activity 1
.model small
.stack 100h
.Data
msg db "Computer science $"
.code
main proc
        mov ax,@data
        mov ds,ax


        mov cx,10
Sum:    lea dx,msg
        mov ah,9
        int 21h
        loop sum


        mov ah,4ch
        int 21h
main endp
end main
```


C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19041.928]
(c) Microsoft Corporation. All rights reserved.
D:\masm615\BIN>masm lab6a1
Microsoft (R) MASM Compatibility Driver
Copyright (C) Microsoft Corp 1993. All rights reserved.
Invoking: ML.EXE /I. /Zm /c lab6a1.asm
Microsoft (R) Macro Assembler Version 6.14.8444
Copyright (C) Microsoft Corp 1981-1997. All rights reserved.
Assembling: lab6a1.asm
D:\masm615\BIN>link lab6a1
Microsoft (R) Segmented Executable Linker Version 5.60.339 Dec 5 1994
Copyright (C) Microsoft Corp 1984-1993. All rights reserved.
Run File [lab6a1.exe]:
List File [nul.map]:
Libraries [.lib]:
Definitions File [nul.def]:
D:\masm615\BIN>
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Welcome to DOSBox v0.74-3
For a short introduction for new users type: INTRO
For supported shell commands type: HELP
To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.
HAVE FUN!
The DOSBox Team http://www.dosbox.com
Z:\>SET BLASTER=A220 I7 D1 H5 T6
Z:\>mount c d:\masm615/bin
Drive C is mounted as local directory d:\masm615/bin\
Z:\>c:
C:\>lab6a1.exe
Computer science Computer science Computer science Computer science Computer sci
ence Computer science Computer science Computer science Computer science Compute
r science
C:\>

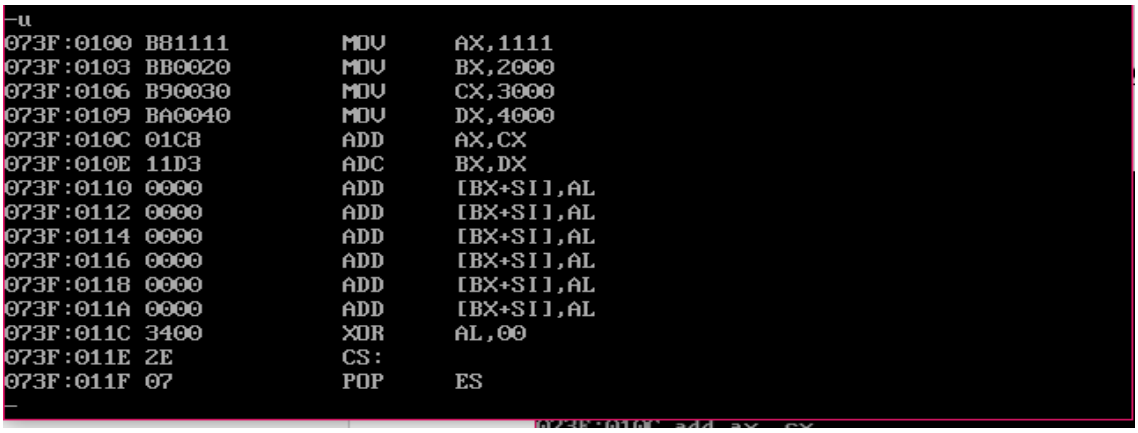
**ACTIVITY 02**

## Lab 7 Activity 1:

## Before Assembling:




## Unassembled code:




## After Assembling:

```
AX=1111  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0103   NV UP EI PL NZ NA PO NC
073F:0103 BB0020        MOV     BX,2000
-t

AX=1111  BX=2000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0106   NV UP EI PL NZ NA PO NC
073F:0106 B90030        MOV     CX,3000
-t

AX=1111  BX=2000  CX=3000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0109   NV UP EI PL NZ NA PO NC
073F:0109 BA0040        MOV     DX,4000
-t

AX=1111  BX=2000  CX=3000  DX=4000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=010C   NV UP EI PL NZ NA PO NC
073F:010C 01C8          ADD     AX,CX
-t

AX=4111  BX=2000  CX=3000  DX=4000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=010E   NV UP EI PL NZ NA PE NC
073F:010E 11D3          ADC     BX,DX
-
```

- ***"NO CHANGE IN FLAG STATUSES"***

## Lab 7 Activity 2:

### Before Assembling:

```
Z:\>c:

C:\>debug
-r
AX=0000  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0100   NV UP EI PL NZ NA PO NC
073F:0100 0000          ADD     [BX+SI],AL                      DS:0000=CD
-a100
073F:0100 mov ax, 3000
073F:0103 mov bx, 2000
073F:0106 mov cx, 1000
073F:0109 mov dx, 4000
073F:010C sub ax, dx
073F:010E sbb bx, cx
073F:0110 _
```

### Unassembled code:

```
-u
073F:0100 B80030        MOV     AX,3000
073F:0103 BB0020        MOV     BX,2000
073F:0106 B90010        MOV     CX,1000
073F:0109 BA0040        MOV     DX,4000
073F:010C 29D0          SUB     AX,DX
073F:010E 19CB          SBB     BX,CX
073F:0110 00F0          ADD     AL,DH
073F:0112 46            INC     SI
073F:0113 7400          JZ      0115
073F:0115 00B200B2      ADD     [BP+SI+B200],DH
073F:0119 0999002E      OR      [BX+DI+2E00],BX
073F:011D 07            POP     ES
073F:011E 2E            CS:
073F:011F 07            POP     ES
```

### After Assembling:

```
AX=3000  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0103   NV UP EI PL NZ NA PO NC
073F:0103 BB0020        MOV     BX,2000
-t

AX=3000  BX=2000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0106   NV UP EI PL NZ NA PO NC
073F:0106 B90010        MOV     CX,1000
-t

AX=3000  BX=2000  CX=1000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0109   NV UP EI PL NZ NA PO NC
073F:0109 BA0040        MOV     DX,4000
-t

AX=3000  BX=2000  CX=1000  DX=4000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=010C   NV UP EI PL NZ NA PO NC
073F:010C 29D0          SUB     AX,DX
-t

AX=F000  BX=2000  CX=1000  DX=4000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=010E   NV UP EI NG NZ NA PE CY
073F:010E 19CB          SBB     BX,CX
```

## Lab 7 Activity 3:

### Before Assembling:

```
C:\>debug
-r
AX=0000  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0100   NV UP EI PL NZ NA PO NC
073F:0100 0000          ADD     [BX+SI],AL                      DS:0000=CD
-a100
073F:0100 mov bl, 05
073F:0102 mov cl, 10
073F:0104 mov al, cl
073F:0106 mul bl
073F:0108 mov dx, ax
073F:010A
```

### Unassembled code:

```
-u
073F:0100 B305          MOV     BL,05
073F:0102 B110          MOV     CL,10
073F:0104 88C8          MOV     AL,CL
073F:0106 F6E3          MUL     BL
073F:0108 89C2          MOV     DX,AX
073F:010A 0000          ADD     [BX+SI],AL
073F:010C 0000          ADD     [BX+SI],AL
073F:010E 0000          ADD     [BX+SI],AL
073F:0110 0000          ADD     [BX+SI],AL
073F:0112 0000          ADD     [BX+SI],AL
073F:0114 0000          ADD     [BX+SI],AL
073F:0116 0000          ADD     [BX+SI],AL
073F:0118 0000          ADD     [BX+SI],AL
073F:011A 0000          ADD     [BX+SI],AL
073F:011C 3400          XOR     AL,00
073F:011E 2E            CS:
073F:011F 07            POP     ES
-_
```

### After Assembling:

```
AX=0000  BX=0005  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0102   NV UP EI PL NZ NA PO NC
073F:0102 B110          MOV     CL,10
-t

AX=0000  BX=0005  CX=0010  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0104   NV UP EI PL NZ NA PO NC
073F:0104 88C8          MOV     AL,CL
-t

AX=0010  BX=0005  CX=0010  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0106   NV UP EI PL NZ NA PO NC
073F:0106 F6E3          MUL     BL
-t

AX=0050  BX=0005  CX=0010  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0108   NV UP EI PL NZ NA PO NC
073F:0108 89C2          MOV     DX,AX
-t

AX=0050  BX=0005  CX=0010  DX=0050  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=010A   NV UP EI PL NZ NA PO NC
073F:010A 0000          ADD     [BX+SI],AL                  DS:0005=EA
-
```

- ***"NO CHANGE IN FLAG STATUSES"***

## Lab 7 Activity 4:

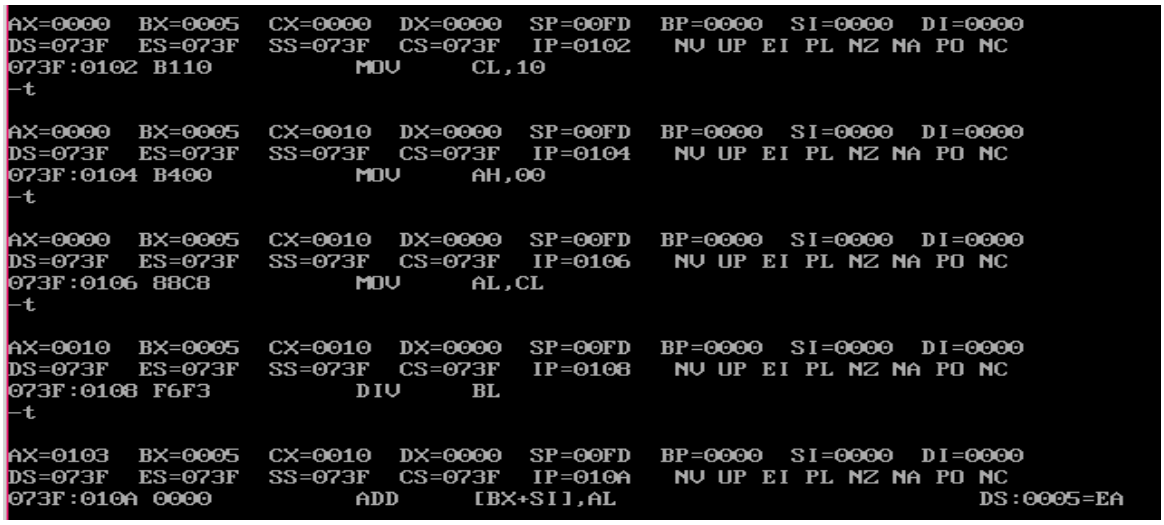
### Before Assembling:

```
C:\>debug
-r
AX=0000  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0100   NV UP EI PL NZ NA PO NC
073F:0100 B305          MOV     BL,05
-a0100
073F:0100 mov bl, 05
073F:0102 mov cl, 10
073F:0104 mov ah, 00
073F:0106 mov al, cl
073F:0108 div bl
073F:010A _
```

### Unassembled code:

```
-u
073F:0100 B305          MOV     BL,05
073F:0102 B110          MOV     CL,10
073F:0104 B400          MOV     AH,00
073F:0106 88C8          MOV     AL,CL
073F:0108 F6F3          DIV     BL
073F:010A 0000          ADD     [BX+SI],AL
073F:010C 0000          ADD     [BX+SI],AL
073F:010E 0000          ADD     [BX+SI],AL
073F:0110 0000          ADD     [BX+SI],AL
073F:0112 0000          ADD     [BX+SI],AL
073F:0114 0000          ADD     [BX+SI],AL
073F:0116 0000          ADD     [BX+SI],AL
073F:0118 0000          ADD     [BX+SI],AL
073F:011A 0000          ADD     [BX+SI],AL
073F:011C 3400          XOR     AL,00
073F:011E 2E            CS:
073F:011F 07            POP     ES
```

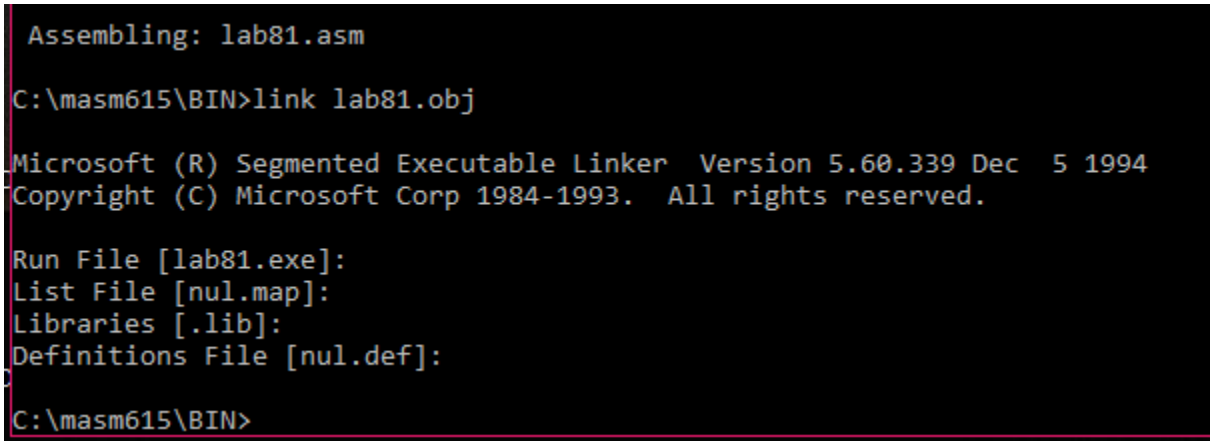
### After Assembling:




## Lab 8 Activity 1:

```
TITLE LAB8A1

.MODEL SMALL

.STACK 100H

.DATA

GRADES DB 69H, 87H, 96H, 45H, 75H

HIGHEST DB ?

.CODE

MAIN PROC

MOV AX,@DATA      ;to initialize DS

MOV DS,AX

MOV CX,4

MOV BX, OFFSET GRADES

MOV AL, [BX]

LBACK:

CMP AL, [BX+1]

JC SWAP

SBACK:
```

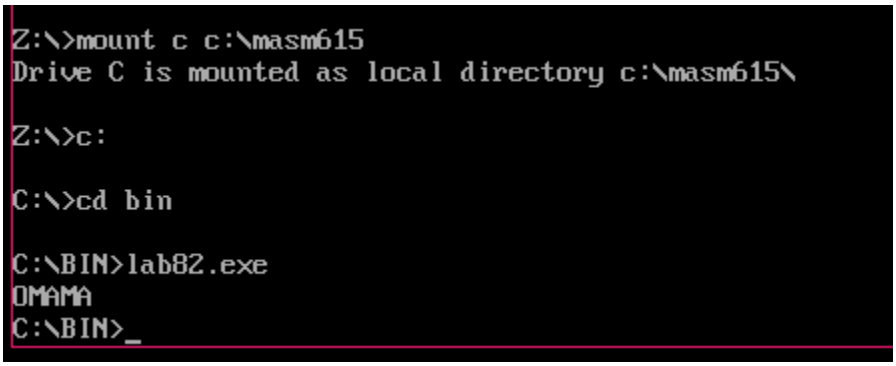
```
INC BX
LOOP LBACK
JMP TER
SWAP:
MOV AL, [BX+1]
JMP SBACK
TER:
MOV HIGHEST, AL

MOV AH,4CH
INT 21H
MAIN ENDP
END MAIN
```




## Lab 8 Activity 2:

```
TITLE LAB8A1
.MODEL SMALL
.STACK 100H
.DATA
```

```
MSG DB ‘omama’
.CODE
MAIN PROC
MOV AX,@DATA      ;to initialize DS
MOV DS, AX
AGAIN:
MOV CX, 5
MOV BX, OFFSET MSG
DISPLAY:
MOV DL, [BX]
SUB DL, 32
INC BX
MOV AH, 2
INT 21H
LOOP DISPLAY
MOV AH, 4CH
INT 21H
MAIN ENDP
END MAIN
```



# Lab 8 Activity 3

```
Title Lab 8 activity 3
.Model Small
.Stack 100h

.Data
num db 76h
ones db 0

.Code
Main Proc
mov ax,@data
mov ds,ax
mov al,num
mov cx,7
mov bl,ones

SBACK:
ROR al,1
JC COUNT


LBACK:
LOOP SBACK
JMP TER

COUNT:
```

Inc BL

Jmp LBACK

TER:

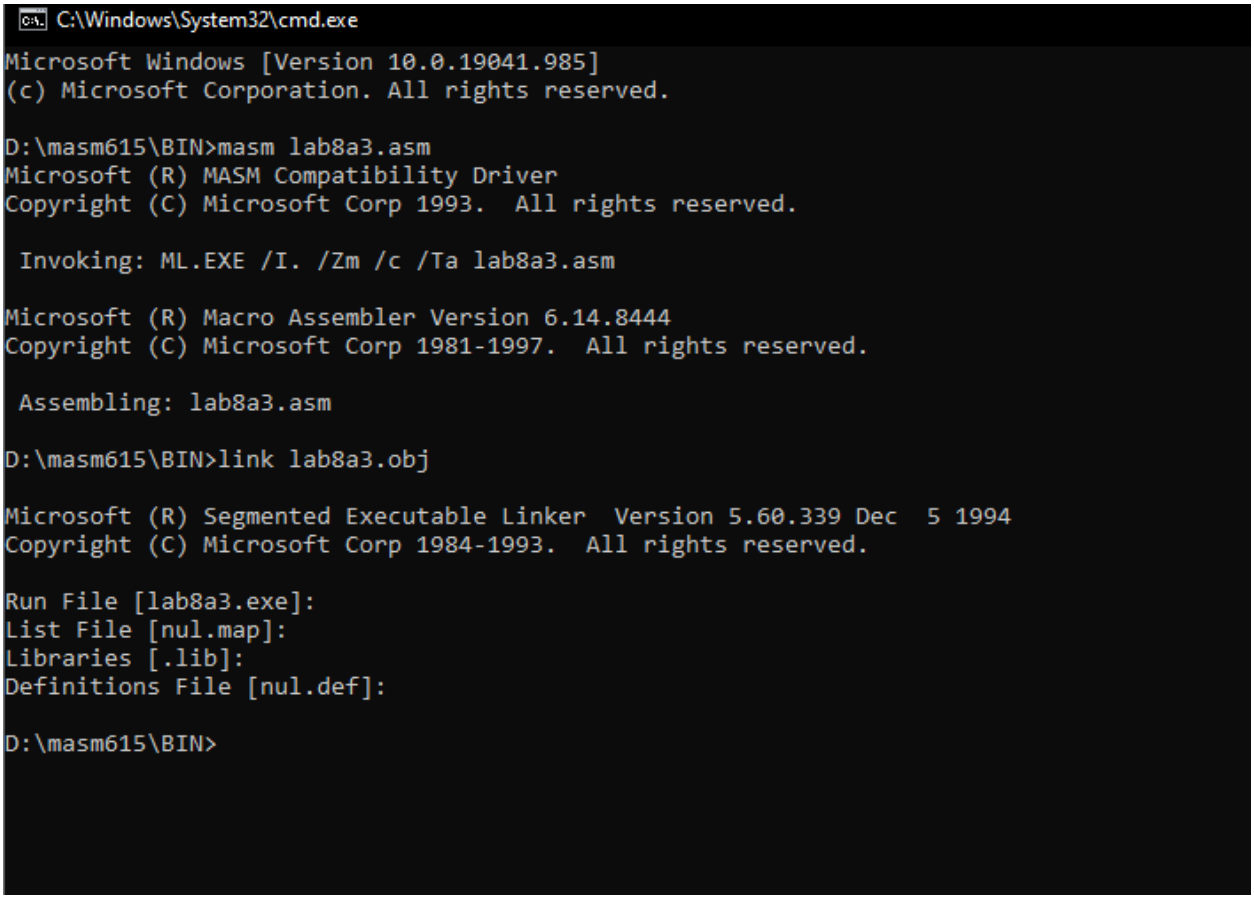
mov ones,bl

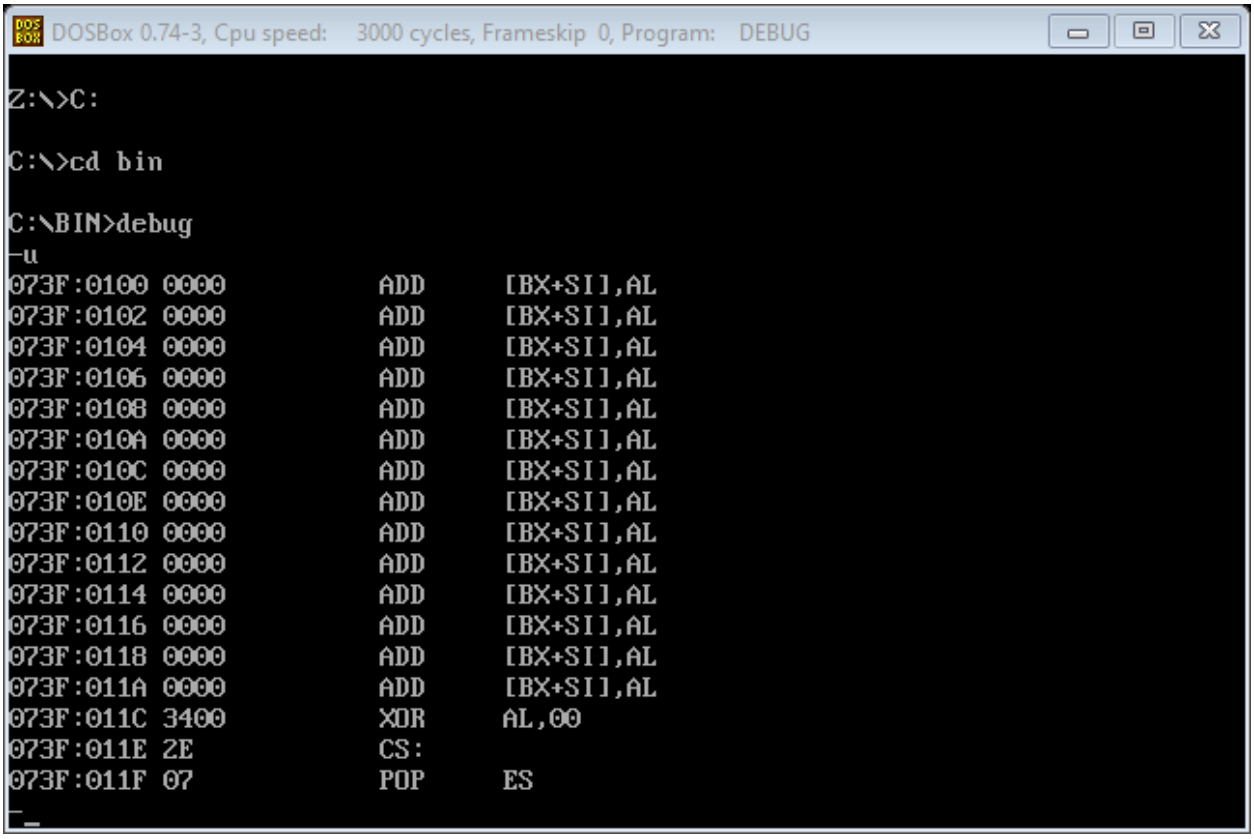
mov ah,4ch

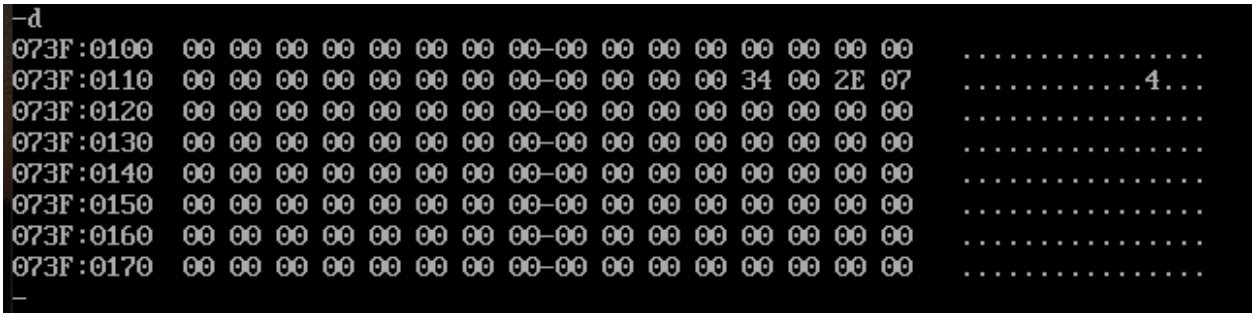
int 21h

Main endp

end Main




DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
Z:\>C:
C:\>cd bin
C:\BIN>debug
-u
073F:0100 0000 ADD [BX+SI],AL
073F:0102 0000 ADD [BX+SI],AL
073F:0104 0000 ADD [BX+SI],AL
073F:0106 0000 ADD [BX+SI],AL
073F:0108 0000 ADD [BX+SI],AL
073F:010A 0000 ADD [BX+SI],AL
073F:010C 0000 ADD [BX+SI],AL
073F:010E 0000 ADD [BX+SI],AL
073F:0110 0000 ADD [BX+SI],AL
073F:0112 0000 ADD [BX+SI],AL
073F:0114 0000 ADD [BX+SI],AL
073F:0116 0000 ADD [BX+SI],AL
073F:0118 0000 ADD [BX+SI],AL
073F:011A 0000 ADD [BX+SI],AL
073F:011C 3400 XOR AL,00
073F:011E 2E CS:
073F:011F 07 POP ES
-

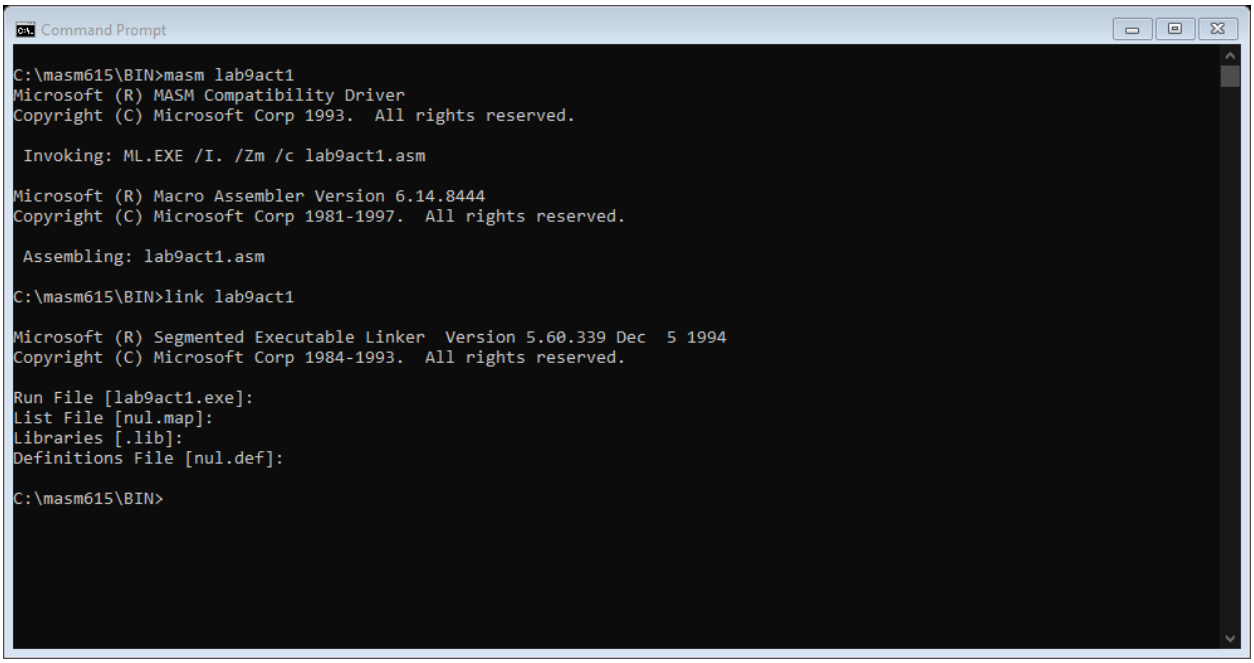

-d
073F:0100 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
073F:0110 00 00 00 00 00 00 00 00-00 00 00 00 34 00 2E 07 ............4...
073F:0120 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
073F:0130 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
073F:0140 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
073F:0150 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
073F:0160 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
073F:0170 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
-

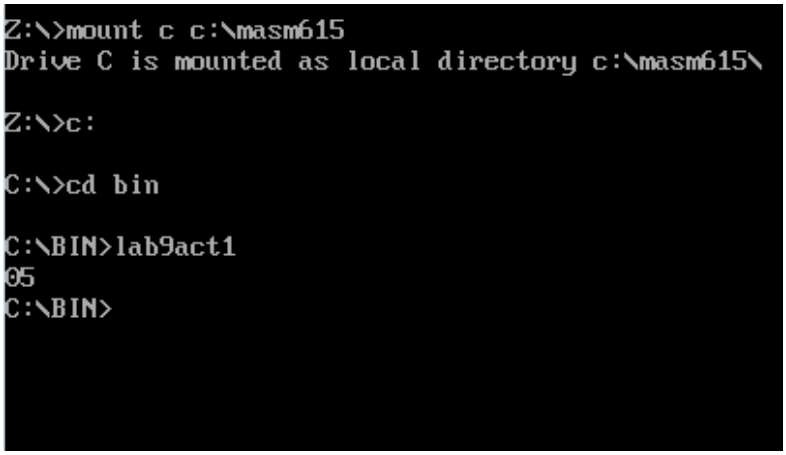
# Lab 9 Activity 1

```
TITLE LAB9ACT1
.MODEL SMALL
.STACK 100H
.DATA
DB 64 DUP (?)
NUM DB 76H
ONES DB 00

.CODE
MAIN PROC
MOV AX,@DATA
MOV DS,AX
;COUNTING NO OF ONES
MOV AL,NUM
MOV CX,07
MOV BH,00
MOV BL,ONES

SBACK:
ROR AL,1
JC COUNT

LBACK:
LOOP SBACK
```

```
JMP TER
COUNT:
INC BL
JMP LBACK

TER:

;CONVERTING INTO UNPACKED BCD

MOV AL,BL
MOV DL,01
MUL DL
AAM
OR AX,3030H
MOV BX,AX

;PRINTING BCD VALUES
MOV AH,2
MOV DL,BH
INT 21H

MOV AH,2
MOV DL,BL
INT 21H

MOV AH,4CH
INT 21H
```

MAIN ENDP




## Lab 9 Activity 2

Title Lab 10 Activity 2

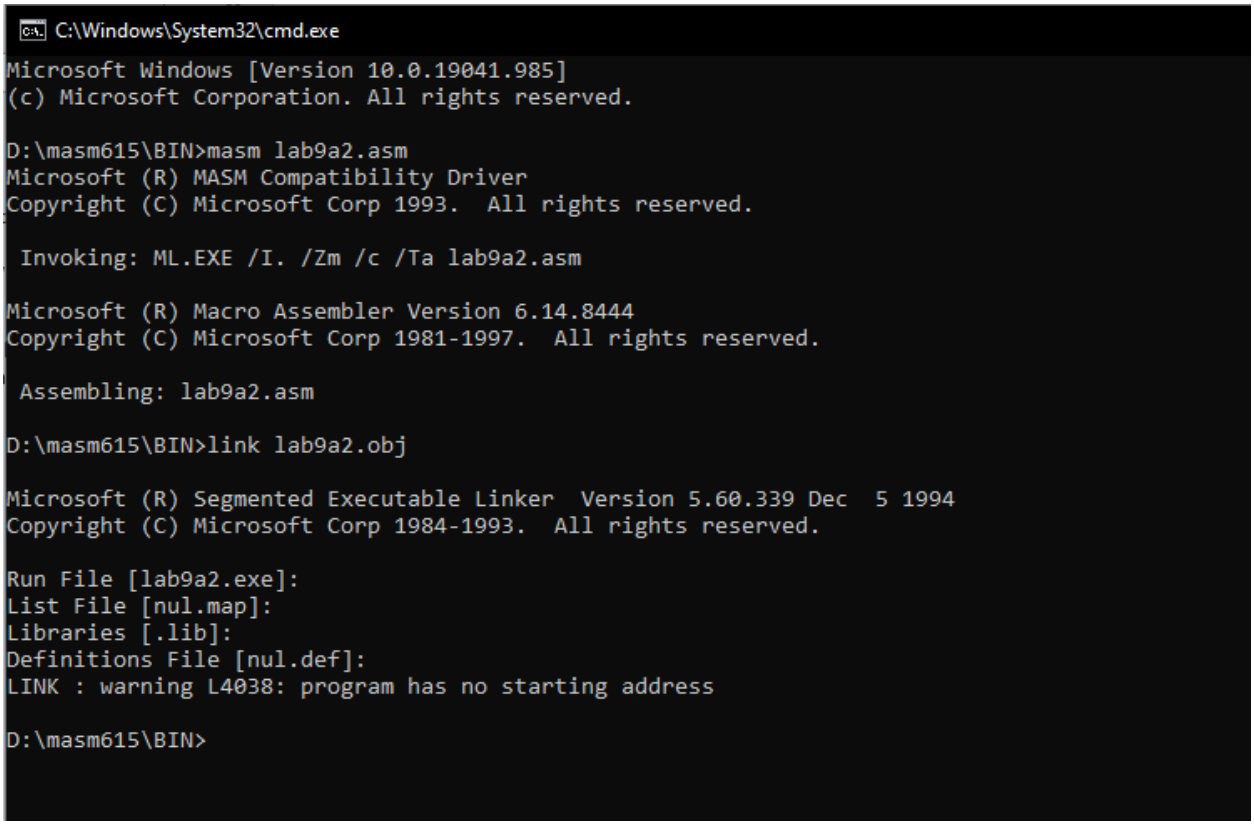
.MODEL SMALL

.STACK 100H

.DATA

```
.CODE
MAIN PROC

        CALL PROCEDURE

        MOV AX,0
        MOV BX,0
MOV CX,0
MOV DX,0

MOV CX,10

MOV AL,3
MOV DH,1

AGAIN:
MOV AH,2
MOV DL,20H
INT 21H

MOV AL,3
MUL DH

AAM
ADD AX,3030H
MOV BX,AX
```

```
MOV AH,2
MOV DL,BH
INT 21H

INC DH

LOOP AGAIN

MOV AH,4CH
INT 21H

MAIN ENDP

PROCEDURE PROC

MOV AH,6
MOV AL,0
MOV CX,0
MOV DX,184FH
MOV BH,7
INT 10H
RET
PROCEDURE ENDP
END
```




## Lab 9 Activity 3(a)

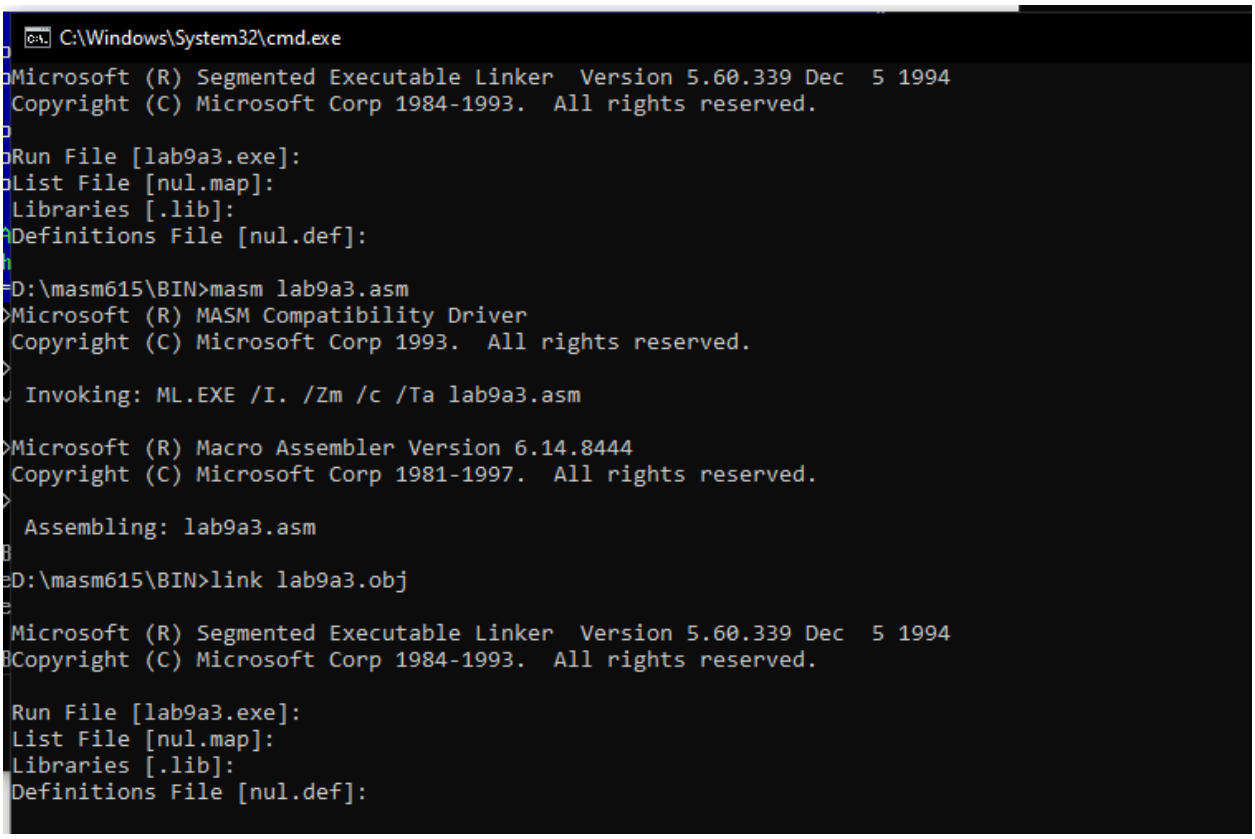
**ADDITION**

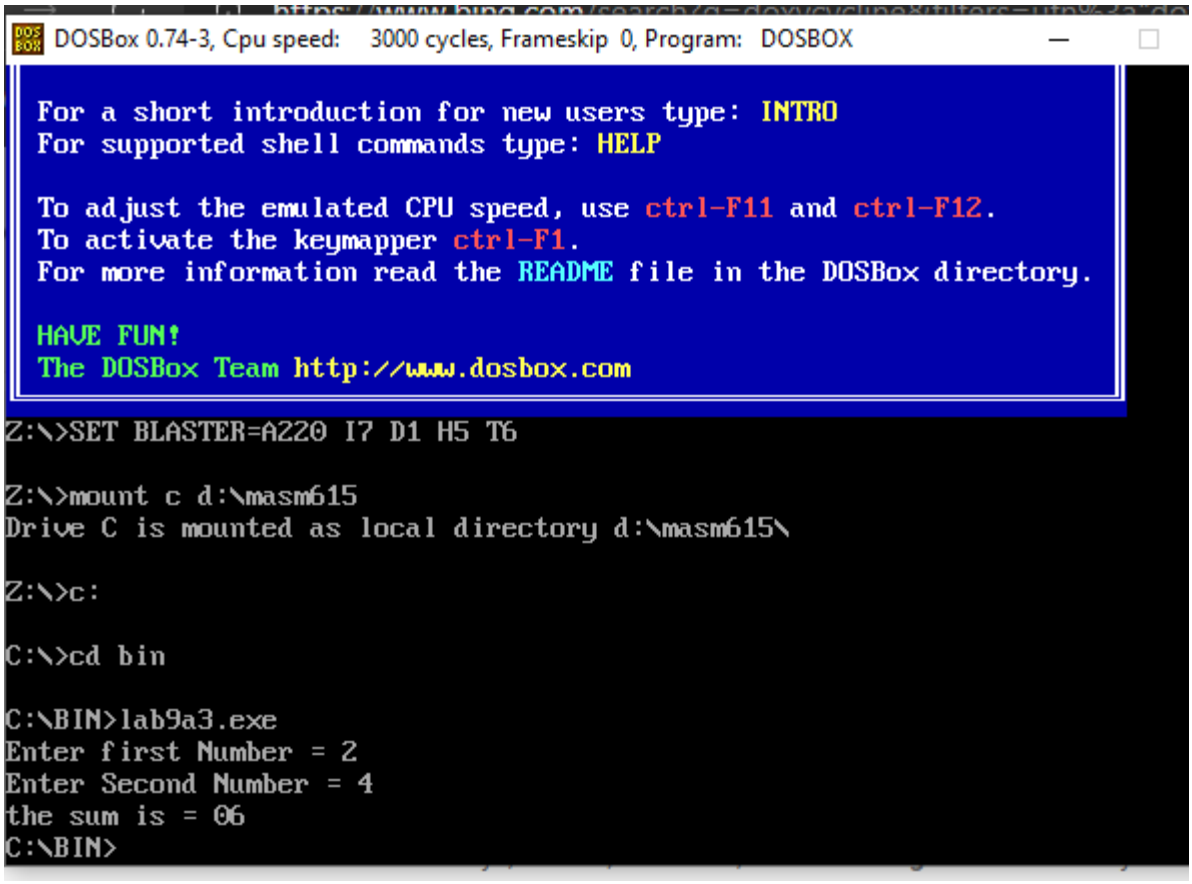
```
Title Lab9 Activity 3
.Model Small
.Stack 100h
.Data
        Msg DB 'Enter first Number = $'
        Msg1 DB 0ah,0dh, 'Enter Second Number = $'
        Msg2 DB 0ah,0dh, 'the sum is = $'
```

```
.Code
Main Proc
        MOV AX,@DATA
        MOV DS,AX

        MOV DX, OFFSET MSG
        MOV AH,9
        INT 21H

        MOV AH,1
        INT 21H

        MOV BL,AL

        MOV DX, OFFSET MSG1
        MOV AH,9
        INT 21H

        MOV AH,1
        INT 21H

        MOV AH,00H
        ADD AL,BL
```

```
        AAA
        OR AX,3030H
        MOV BX,AX

        MOV DX,OFFSET MSG2
        MOV AH,9
        INT 21H

        MOV DL,BH
        MOV AH,2
        INT 21H

        MOV DL,BL
        MOV AH,2
        INT 21H

        MOV AH,4CH
        INT 21H
MAIN ENDP
```

END MAIN


C:\Windows\System32\cmd.exe
Microsoft (R) Segmented Executable Linker  Version 5.60.339 Dec  5 1994
Copyright (C) Microsoft Corp 1984-1993.  All rights reserved.
Run File [lab9a3.exe]:
List File [nul.map]:
Libraries [.lib]:
Definitions File [nul.def]:
D:\masm615\BIN>masm lab9a3.asm
Microsoft (R) MASM Compatibility Driver
Copyright (C) Microsoft Corp 1993.  All rights reserved.
Invoking: ML.EXE /I. /Zm /c /Ta lab9a3.asm
Microsoft (R) Macro Assembler Version 6.14.8444
Copyright (C) Microsoft Corp 1981-1997.  All rights reserved.
Assembling: lab9a3.asm
D:\masm615\BIN>link lab9a3.obj
Microsoft (R) Segmented Executable Linker  Version 5.60.339 Dec  5 1994
Copyright (C) Microsoft Corp 1984-1993.  All rights reserved.
Run File [lab9a3.exe]:
List File [nul.map]:
Libraries [.lib]:
Definitions File [nul.def]:




## Lab 9 Activity 3(b)

## SUBSTRACTION

Title LAB 9 Activity 3b

.Model Small

.Stack 100H

.Data

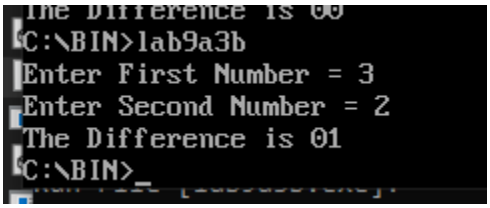
Msg DB 'Enter First Number = $'

Msg1 DB 0ah,0dh,'Enter Second Number = $'

Msg2 DB 0ah,0dh,'The Difference is $'

```
.Code
MAIN PROC

Mov AX,@Data
Mov DS,AX

LEA DX,Msg
Mov AH,9
Int 21h

Mov AH,1
Int 21h

Mov BL,AL

LEA DX,Msg1
Mov AH,9
Int 21h

Mov AH,1
Int 21h

Mov AH,00H
Sub BL,AL

AAS
OR BX,3030H
```

```
Mov CX,BX

LEA DX,Msg2
Mov AH,9
Int 21h

Mov DL,CH

Mov AH,2
Int 21h

Mov DL,CL

Mov AH,2
Int 21h

Mov AH,4CH
Int 21h

Main ENDP
END MAIN
```



```
D:\masm615\BIN>masm lab9a3b.asm
Microsoft (R) MASM Compatibility Driver
Copyright (C) Microsoft Corp 1993.  All rights reserved.

 Invoking: ML.EXE /I. /Zm /c /Ta lab9a3b.asm

Microsoft (R) Macro Assembler Version 6.14.8444
Copyright (C) Microsoft Corp 1981-1997.  All rights reserved.

 Assembling: lab9a3b.asm

D:\masm615\BIN>link lab9a3b.obj

Microsoft (R) Segmented Executable Linker  Version 5.60.339 Dec  5 1994
Copyright (C) Microsoft Corp 1984-1993.  All rights reserved.

Run File [lab9a3b.exe]:
List File [nul.map]:
Libraries [.lib]:
Definitions File [nul.def]:

D:\masm615\BIN>
```

## Lab 9 Activity 3(c)

```
Title LAB 9 Activity 3c

.Model Small

.Stack 100H


.Data

Msg DB 'Enter First Number = $'

Msg1 DB 0ah,0dh,'Enter Second Number = $'

Msg2 DB 0ah,0dh,'The Product of Two Numbers =  $'


.Code

MAIN PROC


Mov AX,@Data

Mov DS,AX
```

```
LEA DX,Msg
Mov AH,9
Int 21h

Mov AH,1
Int 21h

AND AL,0FH
Mov BL,AL

LEA DX,Msg1
Mov AH,9
Int 21h

Mov AH,1
Int 21h

Mov AH,00H
AND AL,0FH
Mul BL

AAM
OR AX,3030H
Mov CX,AX

LEA DX,Msg2
```

```
Mov AH,9
Int 21h

Mov DL,CH

Mov AH,2
Int 21h

Mov DL,CL

Mov AH,2
Int 21h

Mov AH,4CH
Int 21h

Main ENDP
END MAIN
```

```
Z:\>mount c d:\masm615\bin
Drive C is mounted as local directory d:\masm615\bin\

Z:\>c:

C:\>lab9a3c.exe
Enter First Number = 3
Enter Second Number =

The Product of Two Numbers =  39
C:\>lab9a3c.exe
Enter First Number = 2
Enter Second Number = 2
The Product of Two Numbers =  04
C:\>
```

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19041.985]
(c) Microsoft Corporation. All rights reserved.

D:\masm615\BIN>masm615 lab9a3c.asm
'masm615' is not recognized as an internal or external command,
operable program or batch file.

D:\masm615\BIN>masm lab9a3c.asm
Microsoft (R) MASM Compatibility Driver
Copyright (C) Microsoft Corp 1993.  All rights reserved.

 Invoking: ML.EXE /I. /Zm /c /Ta lab9a3c.asm

Microsoft (R) Macro Assembler Version 6.14.8444
Copyright (C) Microsoft Corp 1981-1997.  All rights reserved.

 Assembling: lab9a3c.asm

D:\masm615\BIN>link lab9a3c.obj

Microsoft (R) Segmented Executable Linker  Version 5.60.339 Dec  5 1994
Copyright (C) Microsoft Corp 1984-1993.  All rights reserved.

Run File [lab9a3c.exe]:
List File [nul.map]:
Libraries [.lib]:
Definitions File [nul.def]:

D:\masm615\BIN>
D:\masm615\BIN>
```

# Lab 9 Activity 3(d)

```
Title LAB 9 Activity 3d
.Model Small
.Stack 100H

.Data
Msg DB 'Enter First Number = $'
Msg1 DB 0ah,0dh,'Enter Second Number = $'

Msg2 db 0ah,0dh,'Remainder is = $'
Msg3 db 0ah,0dh,'Quotient is = $'

.Code
MAIN PROC

Mov AX,@Data
Mov DS,AX

LEA DX,Msg
Mov AH,9
Int 21h

Mov AH,1
Int 21h

AND AL,0FH
Mov BL,AL
```

```
LEA DX,Msg1
Mov AH,9
Int 21h

Mov AH,1
Int 21h

Mov AH,00H
AND AL,0FH

XCHG AL,BL

AAD

DIV BL
OR AX,3030H
Mov CX,AX

LEA DX,Msg2
Mov AH,9
Int 21h

Mov DL,CH

Mov AH,2
Int 21h
```

```
LEA DX,Msg3
Mov AH,9
Int 21h


Mov DL,CL

Mov AH,2
Int 21h

Mov AH,4CH
Int 21h

Main ENDP
END MAIN
```

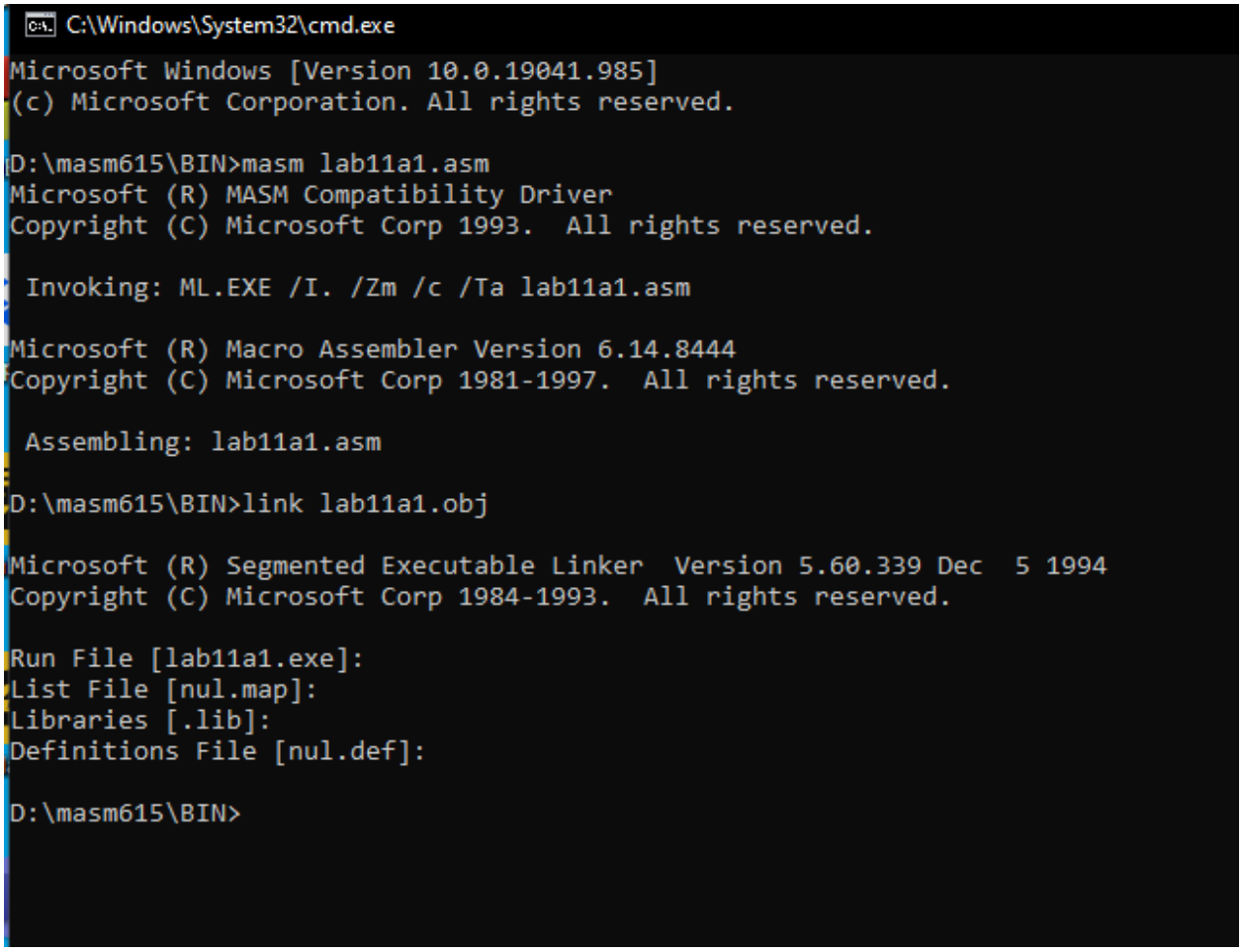
```
C:\>lab9a3d.exe
Enter First Number = 3
Enter Second Number = 4
Remainder is = 3
Quotient is = 0
C:\>lab9a3d.exe
Enter First Number = 4
Enter Second Number = 2
Remainder is = 0
Quotient is = 2
C:\>
```

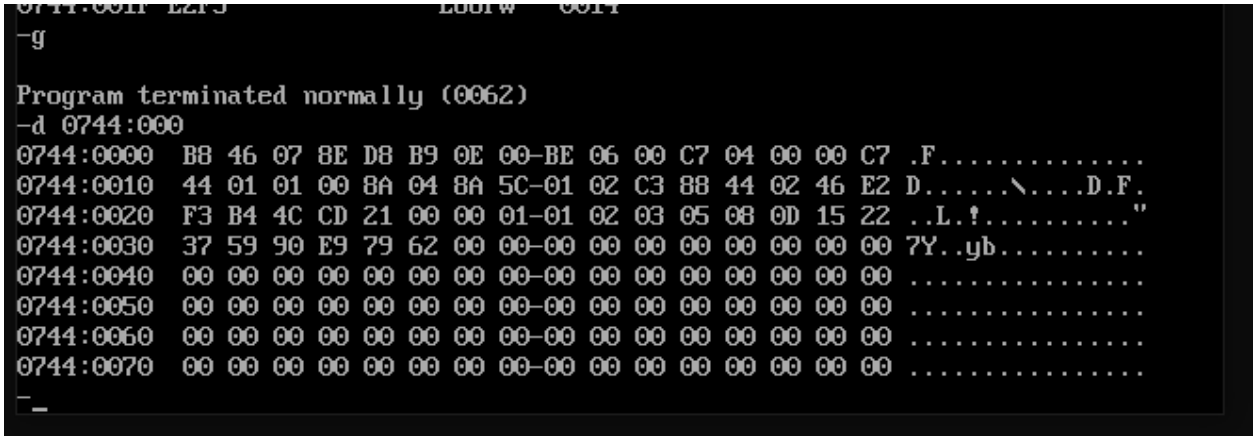
```
D:\masm615\BIN>masm lab9a3d.asm
Microsoft (R) MASM Compatibility Driver
Copyright (C) Microsoft Corp 1993.  All rights reserved.

 Invoking: ML.EXE /I. /Zm /c /Ta lab9a3d.asm

Microsoft (R) Macro Assembler Version 6.14.8444
Copyright (C) Microsoft Corp 1981-1997.  All rights reserved.

 Assembling: lab9a3d.asm

D:\masm615\BIN>link lab9a3d.obj

Microsoft (R) Segmented Executable Linker  Version 5.60.339 Dec  5 1994
Copyright (C) Microsoft Corp 1984-1993.  All rights reserved.

Run File [lab9a3d.exe]:
List File [nul.map]:
Libraries [.lib]:
Definitions File [nul.def]:

D:\masm615\BIN>
```

## Lab 11 Activity 1

```
TITLE LAB 11 ACTIVITY 1

.MODEL SMALL

.STACK 100H

.DATA


array db 13 dup (?)


.CODE

MAIN PROC
```

```
MOV AX,@DATA
MOV DS,AX

mov cx,14

mov SI,offset array

mov [SI],00h
mov [SI+1],01h

again:

mov al,[SI]
mov bl,[SI+1]

add al,bl
mov [SI+2],al

inc SI
loop again

MOV AH,4CH
INT 21H

MAIN ENDP
END MAIN
```


C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19041.985]
(c) Microsoft Corporation. All rights reserved.
D:\masm615\BIN>masm lab11a1.asm
Microsoft (R) MASM Compatibility Driver
Copyright (C) Microsoft Corp 1993. All rights reserved.
Invoking: ML.EXE /I. /Zm /c /Ta lab11a1.asm
Microsoft (R) Macro Assembler Version 6.14.8444
Copyright (C) Microsoft Corp 1981-1997. All rights reserved.
Assembling: lab11a1.asm
D:\masm615\BIN>link lab11a1.obj
Microsoft (R) Segmented Executable Linker Version 5.60.339 Dec 5 1994
Copyright (C) Microsoft Corp 1984-1993. All rights reserved.
Run File [lab11a1.exe]:
List File [nul.map]:
Libraries [.lib]:
Definitions File [nul.def]:
D:\masm615\BIN>


-g
Program terminated normally (0062)
-d 0744:000
0744:0000 B8 46 07 8E D8 B9 0E 00-BE 06 00 C7 04 00 00 C7 .F..............
0744:0010 44 01 01 00 8A 04 8A 5C-01 02 C3 88 44 02 46 E2 D......\....D.F.
0744:0020 F3 B4 4C CD 21 00 00 01-01 02 03 05 08 0D 15 22 ..L.!..........
0744:0030 37 59 90 E9 79 62 00 00-00 00 00 00 00 00 00 00 7Y..yb..........
0744:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
0744:0050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
0744:0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
0744:0070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................

```
C:\BIN>debug lab11a1.exe
-u
0744:0000 B84607          MOV     AX,0746
0744:0003 8ED8            MOV     DS,AX
0744:0005 B90E00          MOV     CX,000E
0744:0008 BE0600          MOV     SI,0006
0744:000B C7040000        MOV     WORD PTR [SI],0000
0744:000F C744010100      MOV     WORD PTR [SI+01],0001
0744:0014 8A04            MOV     AL,[SI]
0744:0016 8A5C01          MOV     BL,[SI+01]
0744:0019 02C3            ADD     AL,BL
0744:001B 884402          MOV     [SI+02],AL
0744:001E 46              INC     SI
0744:001F E2F3            LOOPW   0014
-
```

## Lab 11 Activity 2

```
TITLE LAB 11 ACTIVITY 2


.MODEL SMALL

.STACK 100H

.DATA

MSG DB 'ENETR A DIGIT TO PRINT THE FACTORIALS  = $'


.CODE

MAIN PROC


MOV AX,@DATA

MOV DS,AX


MOV AH,9

LEA DX,MSG

INT 21H
```

**MOV AH,1**

**INT 21H**

**SUB AL,30H**

**MOV CL,AL**

**SUB CL,1**

**MOV BL,AL**

**SUB BL,1**

**AGAIN:**

**MUL BL**

**SUB BL,1**

**LOOP AGAIN**

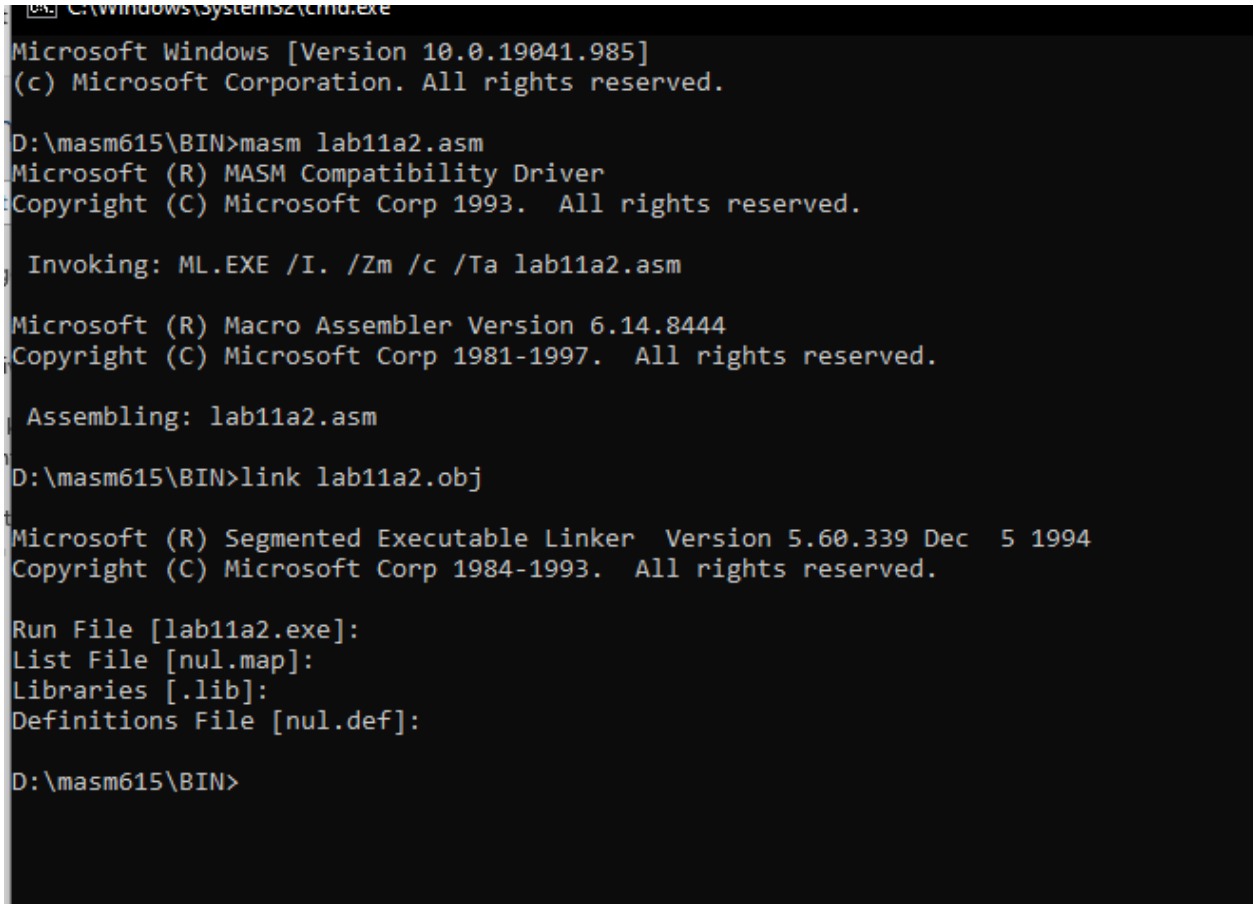
**MOV AH,4CH**

**INT 21H**

**MAIN ENDP**

**END MAIN**

```
C:\>debug lab11a2.exe
-g
ENETR A DIGIT TO PRINT THE FACTORIALS  = 4
Program terminated normally (0018)
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=072A ES=072A SS=072A CS=072A IP=0100 NV UP EI NG NZ NA PO NC
072A:0100 C3                RET
```


C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19041.985]
(c) Microsoft Corporation. All rights reserved.
D:\masm615\BIN>masm lab11a2.asm
Microsoft (R) MASM Compatibility Driver
Copyright (C) Microsoft Corp 1993. All rights reserved.
Invoking: ML.EXE /I. /Zm /c /Ta lab11a2.asm
Microsoft (R) Macro Assembler Version 6.14.8444
Copyright (C) Microsoft Corp 1981-1997. All rights reserved.
Assembling: lab11a2.asm
D:\masm615\BIN>link lab11a2.obj
Microsoft (R) Segmented Executable Linker Version 5.60.339 Dec 5 1994
Copyright (C) Microsoft Corp 1984-1993. All rights reserved.
Run File [lab11a2.exe]:
List File [nul.map]:
Libraries [.lib]:
Definitions File [nul.def]:
D:\masm615\BIN>