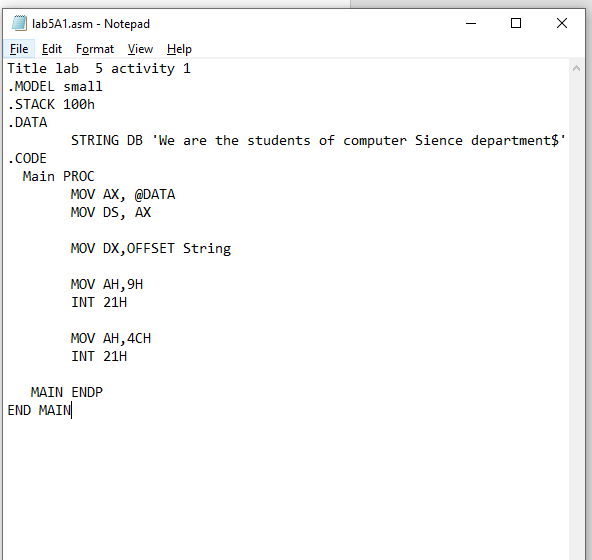
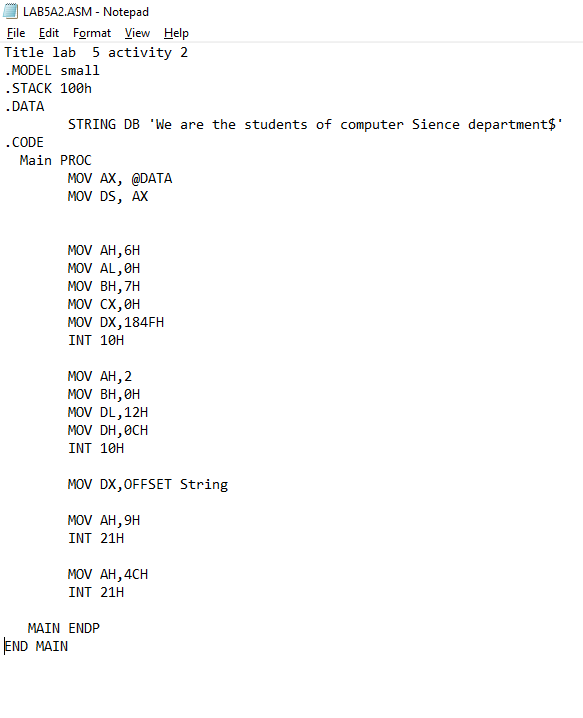
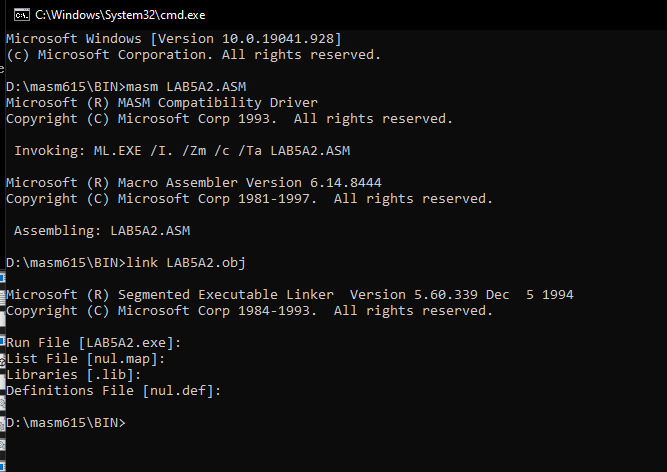
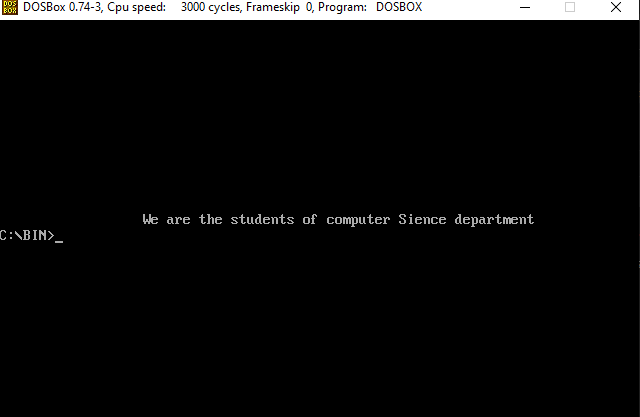
**LAB 05**

**ACTIVITY 01**

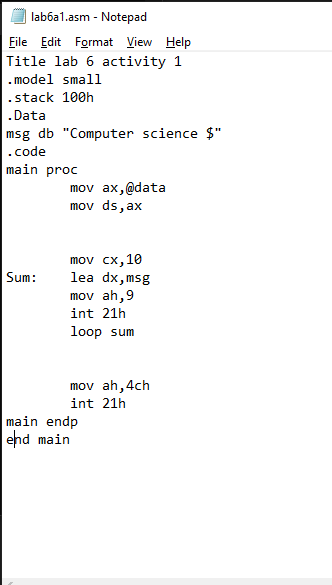


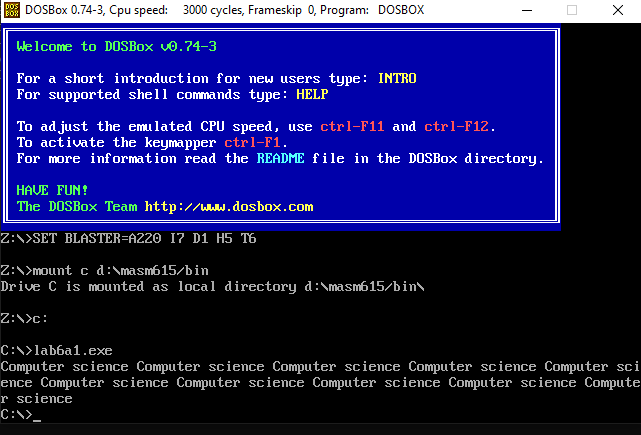
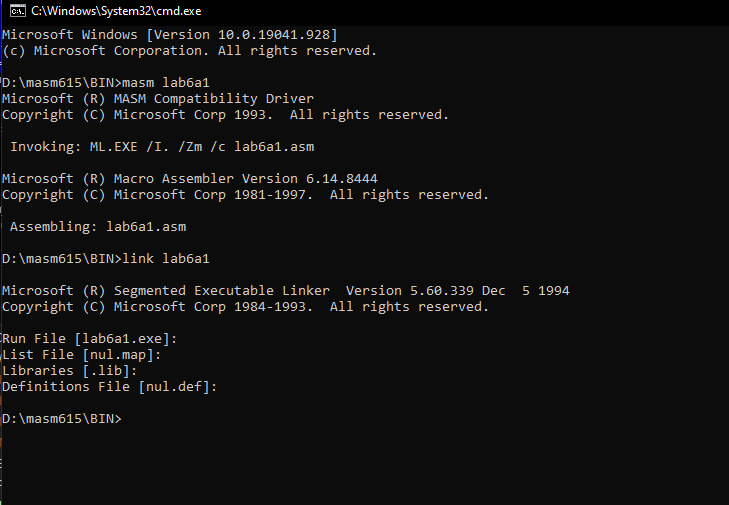
**ACTIVITY 2**





**LAB 6**

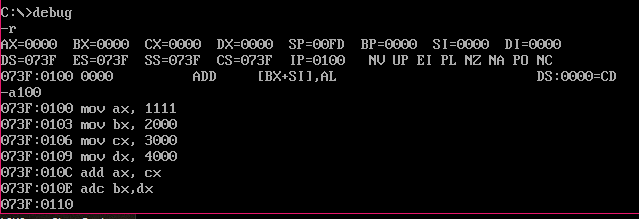
**ACTIVITY 01**

****

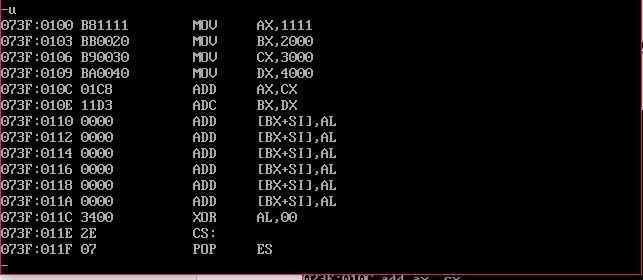
**ACTIVITY 02**

**Lab 7 Activity 1:**

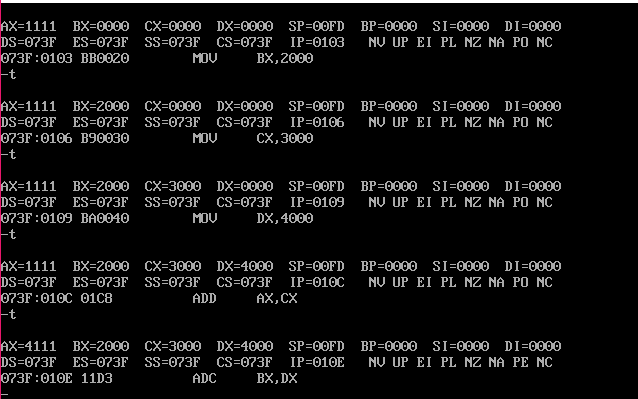
**Before Assembling:**



**Unassembled code:**



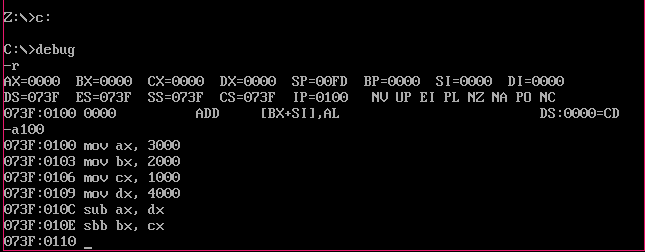
**After Assembling:**



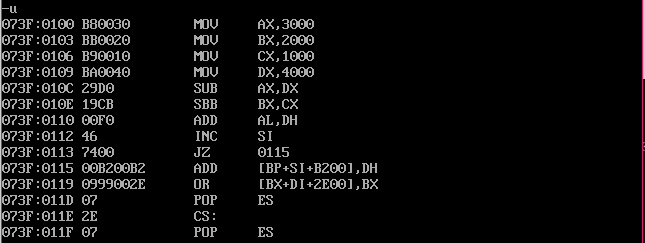
* ***“NO CHANGE IN FLAG STATUSES”***

**Lab 7 Activity 2:**

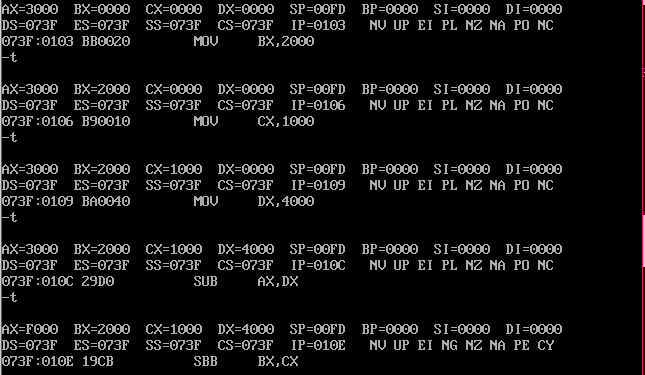
**Before Assembling:**



**Unassembled code:**

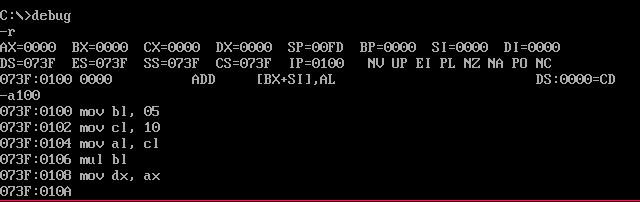


**After Assembling:**

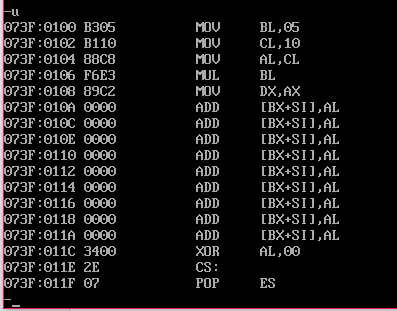


**Lab 7 Activity 3:**

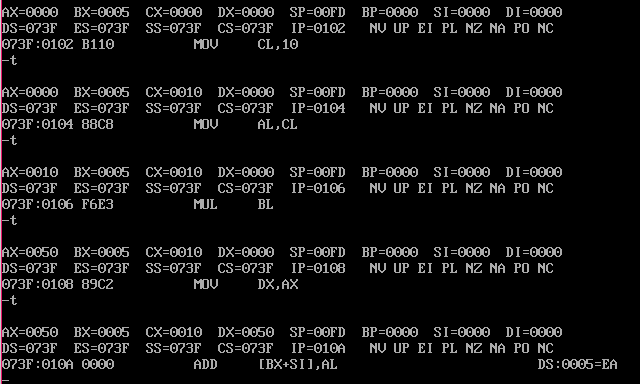
**Before Assembling:**



**Unassembled code:**



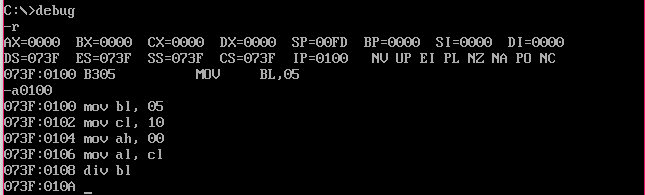
**After Assembling:**



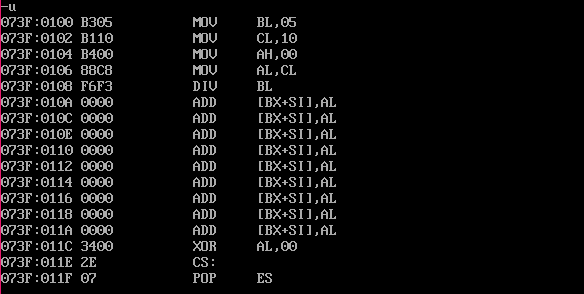
* ***“NO CHANGE IN FLAG STATUSES”***

**Lab 7 Activity 4:**

**Before Assembling:**



**Unassembled code:**



**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:

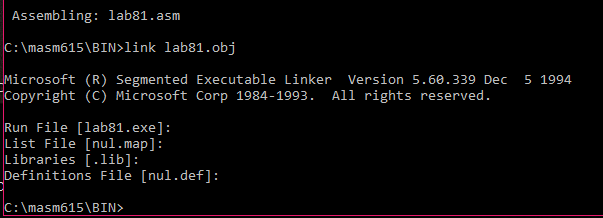
MOV HIGHEST, AL

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

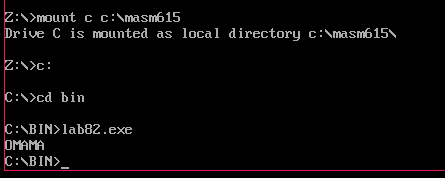
LOOP DISPLAY

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:

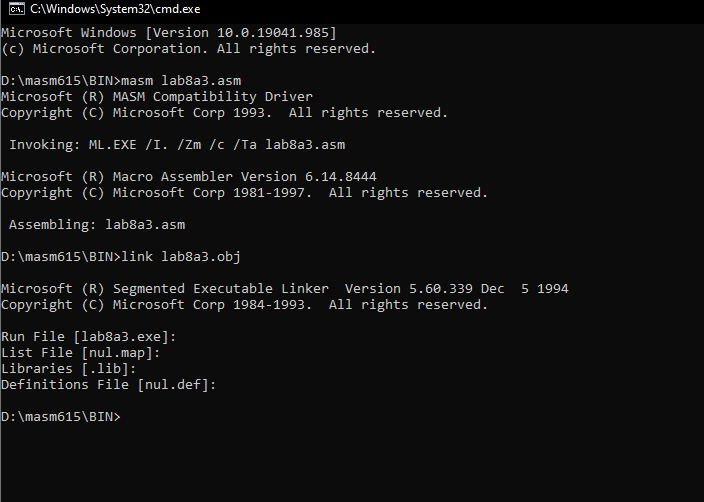
mov ones,bl

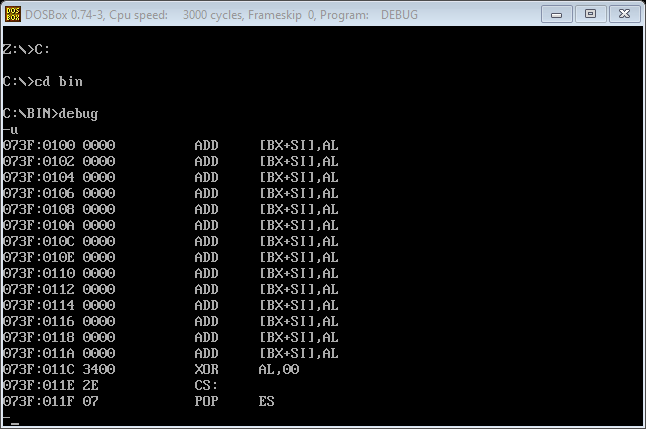
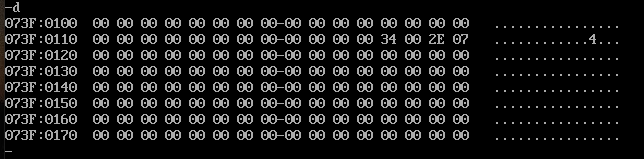
mov ah,4ch

int 21h

Main endp

end Main





**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

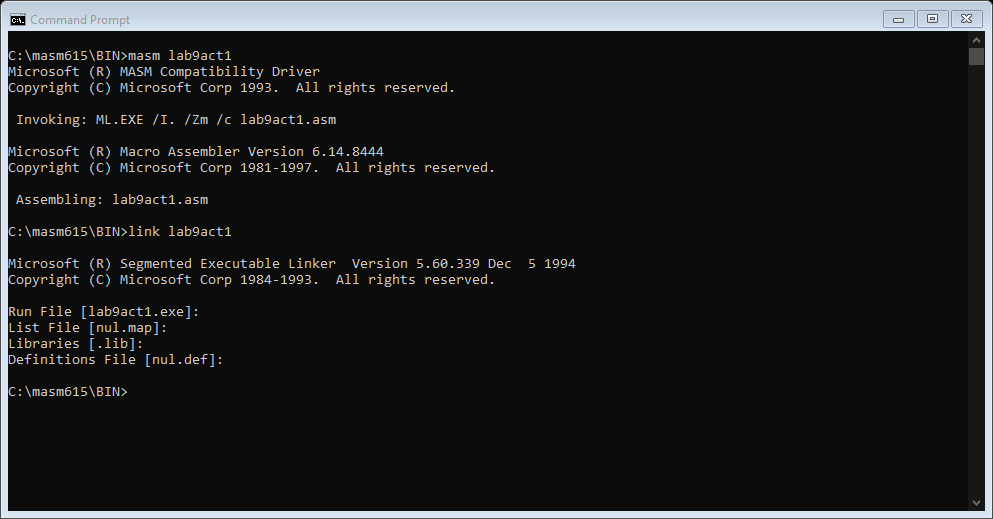
INT 21H

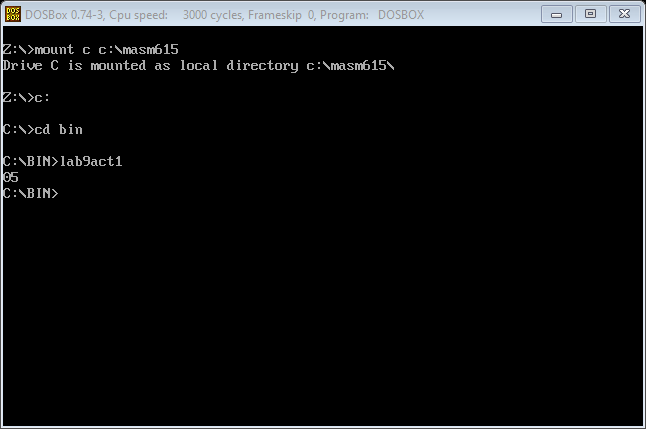
MOV AH,4CH

INT 21H

MAIN ENDP

END MAIN

****

****

**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

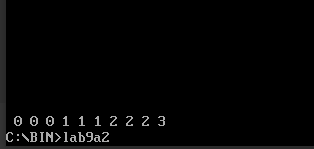
MOV AH,4CH

INT 21H

MAIN ENDP

PROCEDURE PROC

MOV AH,6

MOV AL,0

MOV CX,0

MOV DX,184FH

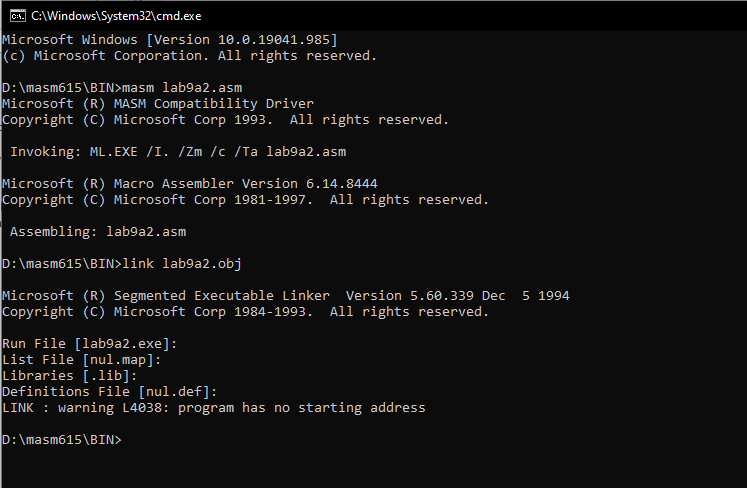
MOV BH,7

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

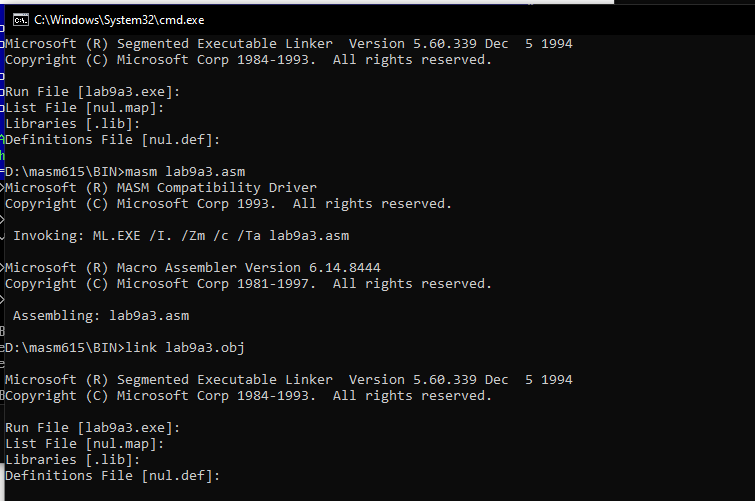
MOV AH,2

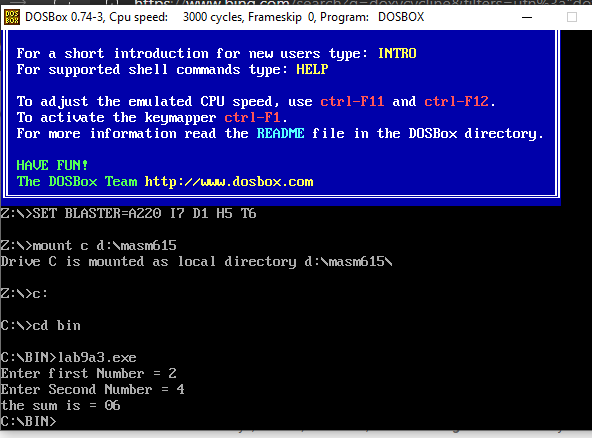
INT 21H

MOV AH,4CH

INT 21H

MAIN ENDP

END MAIN



**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

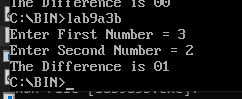
Int 21h

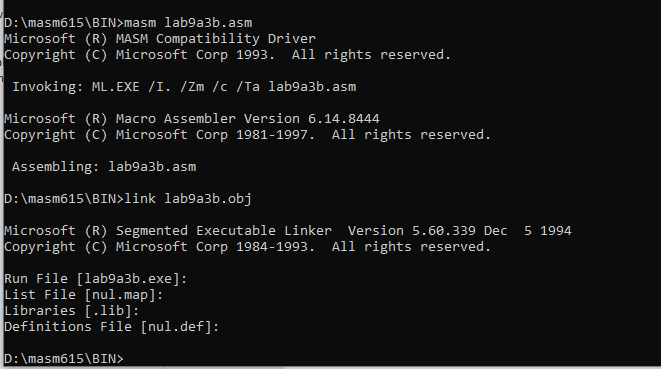
Mov AH,4CH

Int 21h

Main ENDP

END MAIN

****

****

**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

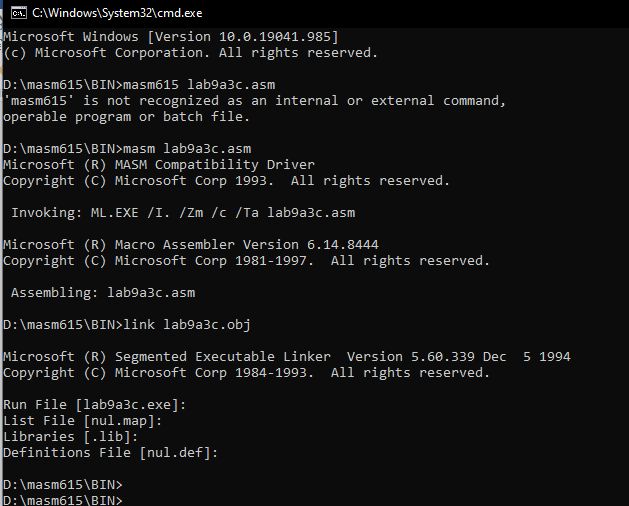
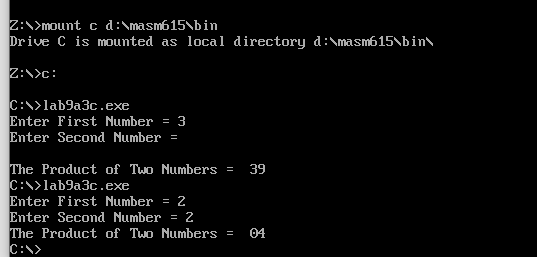
Int 21h

Mov AH,4CH

Int 21h

Main ENDP

END MAIN



**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

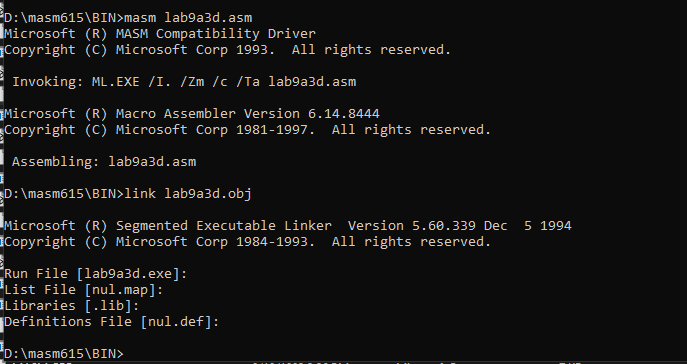
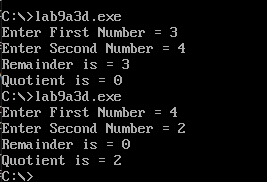
Int 21h

Mov AH,4CH

Int 21h

Main ENDP

END MAIN



**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

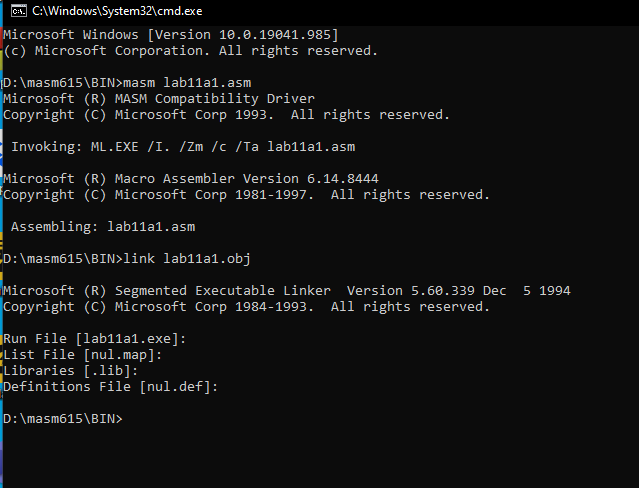
loop again

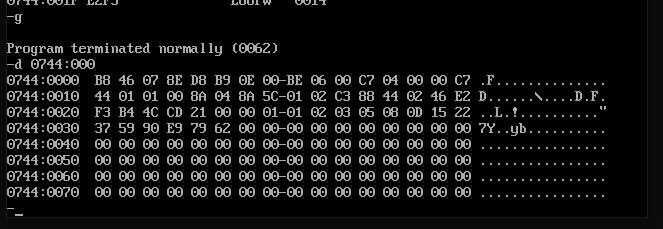
MOV AH,4CH

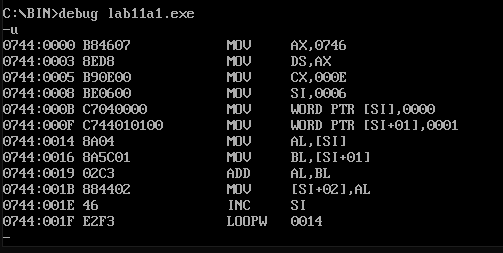
INT 21H

MAIN ENDP

END MAIN







**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**

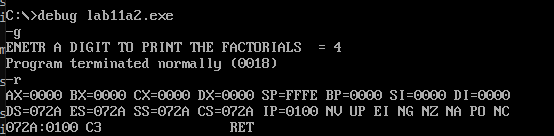
**AGAIN:**

**MUL BL**

**SUB BL,1**

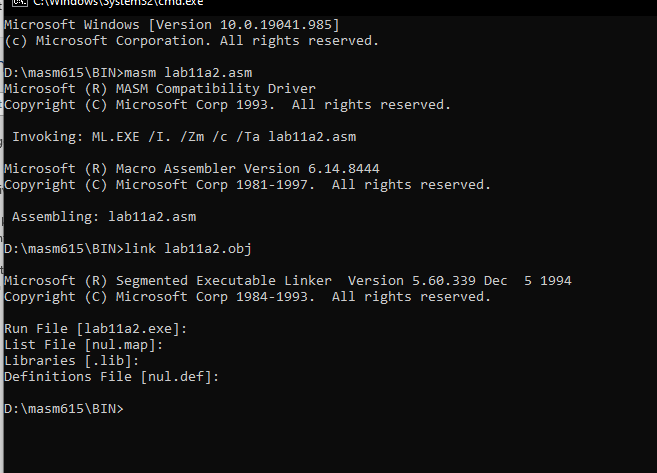
**LOOP AGAIN**

**MOV AH,4CH**

**INT 21H**

**MAIN ENDP**

**END MAIN**

****

**Lab12 activity A**

.MODEL SMALL

.stack 100h

.data

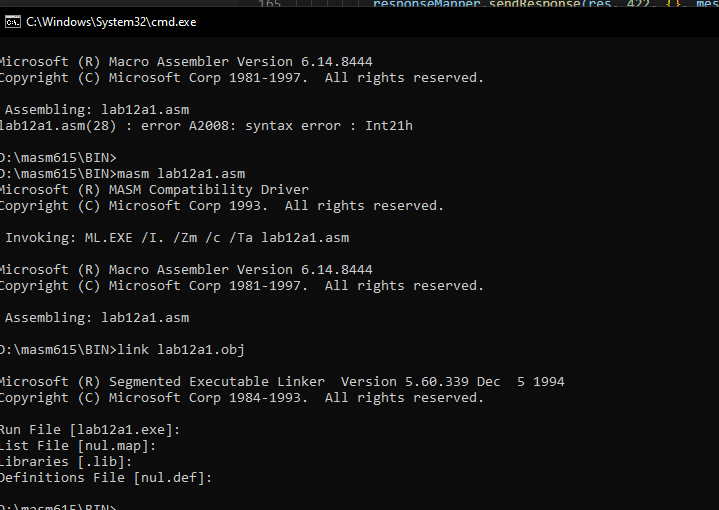
String1 db 'Munib$'

String2 db 'Munib$'

msg db 'Equal$'

msg2 db 'unEqual$'

.code

****Main proc

Mov ax,@data

Mov Ds, ax

Mov es, ax

Mov cx,6

LEA SI, String1

LEA DI, String2

Rep CMPSB

JE MESSAGE

Lea dx, msg2

Mov ah,09h

Int 21h

Jmp Exit

Message:

LEA DX, msg

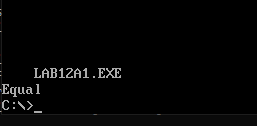
Mov ah,09h

Int 21h

Exit:

Mov ah,4ch

Int 21h

****

Main Endp

End MAIN