NAME – ARKA SINHA

BATCH – CSE A2

SUBJECT – MMC LAB REPORT

2020-2021

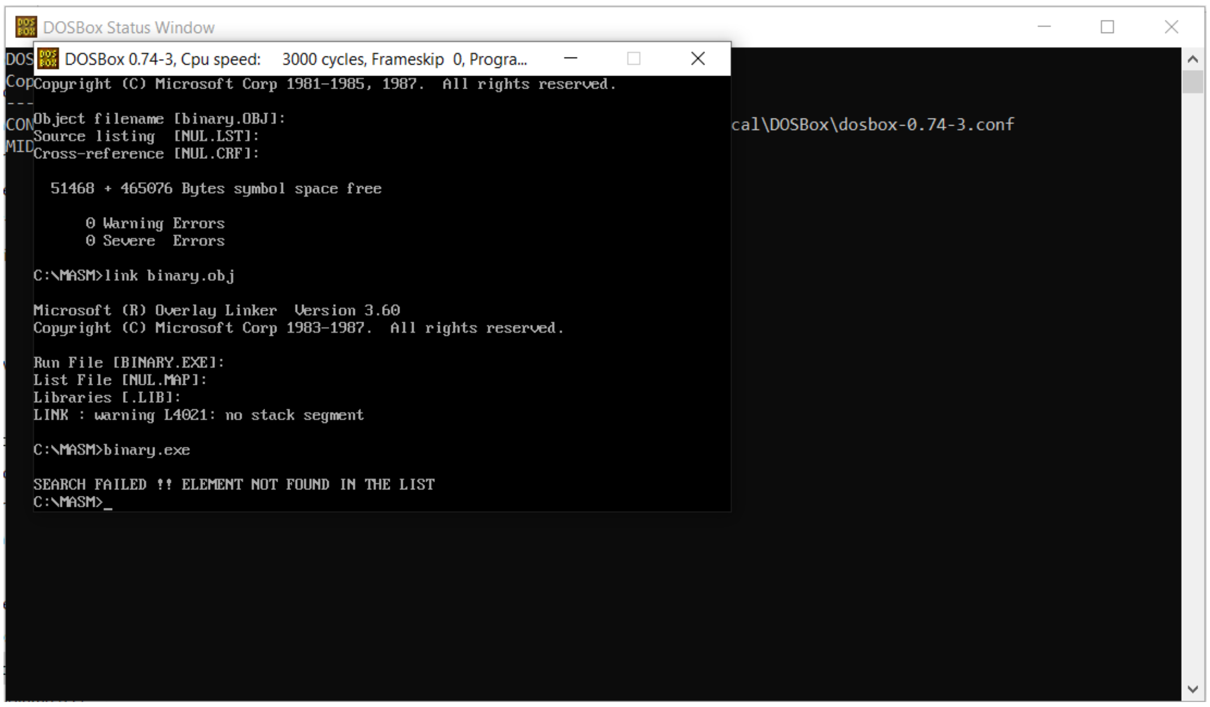
LAB PROGRAM 1

Source Code-

```
;
PROGRAM
::
ASSEMBLY
LANGUAGE
PROGRAM
TO
SEARCH A
KEY
ELEMENT
IN A
           ;            LIST OF 'n' NUMBER USING THE BINARY SEARCH ALGORITHM


           .MODEL SMALL


           ; MACRO TO DISPLAY THE MESSAGE....
           DISPLAY MACRO MSG
                   LEA DX, MSG
                   MOV AH, 09H
                   INT 21H
           ENDM


           .DATA
           LIST DB 01H, 05H, 07H, 10H, 12H, 14H
           NUMBER EQU ($-LIST)
           KEY DB 011H
           MSG1 DB 0DH, 0AH, "ELEMENT FOUND IN THE LIST...$"
           MSG2 DB 0DH, 0AH, "SEARCH FAILED !! ELEMENT NOT FOUND IN THE LIST $"


           .CODE
           START : MOV AX, @DATA
                   MOV DS, AX
                   MOV CH, NUMBER-1   ; HIGH VALUE...
                   MOV CL, 00H        ; LOW VALUE...
           AGAIN:  MOV SI, OFFSET LIST
                   XOR AX, AX
                   CMP CL, CH
```

```
        JE NEXT
        JNC FAILED
NEXT:   MOV AL, CL
        ADD AL, CH
        SHR AL, 01H             ; DIVIDE BY 2
        MOV BL, AL
        XOR AH, AH              ; CLEAR AH
        MOV BP, AX
        MOV AL, DS:[BP][SI]
        CMP AL, KEY             ; COMPARE KEY AND A[i]
        JE SUCCESS              ; IF EQUAL, DISPLAY SUCCESS MESSAGE
        JC INCLOW
        MOV CH, BL              ; IF KEY>A[i]  SHIFT HIGH
        DEC CH
        JMP AGAIN
INCLOW: MOV CL, BL              ; IF KEY<A[i]  SHIFT LOW
        INC CL
        JMP AGAIN
SUCCESS:DISPLAY MSG1
        JMP FINAL
FAILED: DISPLAY MSG2            ; JOB OVER. TERMINATE....
FINAL : MOV AH, 4CH
        INT 21H
END START
```

Screen Shot-


DOSBox Status Window
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [binary.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51468 + 465076 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\MASM>link binary.obj

Microsoft (R) Overlay Linker  Version 3.60
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.

Run File [BINARY.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\MASM>binary.exe

SEARCH FAILED !! ELEMENT NOT FOUND IN THE LIST
C:\MASM>_
```

```
cal\DOSBox\dosbox-0.74-3.conf
```

Writeup-

MMC Week-2

Binary search

```
.MODEL SMALL

DISPLAY  MACRO MSG
         LEA DX, MSG
         MOV AH, 09H
         INT 21H
ENDM

.DATA
        LIST DB 01H, 05H, 07H, 10H, 12H, 14H
        NUMBER EQU ($-LIST)
        KEY DB 12H
        MSG1 DB 0DH, 0AH, "ELEMENT FOUND IN LIST:..$"
        MSG2 DB 0DH, 0AH, "SEARCH FAILED!!NOFOUND $"

.CODE
START:  MOV AX, @DATA
        MOV DS, AX

        MOV CH, NUMBER-1 ; HIGH VALUE...
        MOV CL, 00H      ; LOW VALUE...
AGAIN:  MOV SI, OFFSET LIST
        XOR AX, AX ; clear the Ax register
        CMP CL,CH
        JE NEXT
        JNC FAILED
NEXT:   MOV AL, CL
        ADD AL, CH
        SHR AL, 01H    ; DIVIDE BY 2
        MOV BL,AL
        XOR AH, AH     ; CLEAR AH
        MOV BP, AX
```

```
    MOV AL, DS:[BP][SI]
    CMP AL, KEY          ; COMPARE KEY AND A[I]
    JE SUCCESS           ; IF EQUAL, DISPLAY SUCCESS MESSAGE
    JC INCLOW
    MOV CH, BL           ; IF KEY>A[I] SHIFT HIGH
    DEC CH
    JMP AGAIN
INCLOW: MOV CL, BL       ; IFKEY<A[I] SHIFTLOW
        INC CL
        JMP AGAIN
SUCCESS: DISPLAY MSG1
         JMP FINAL
FAILED: DISPLAY MSG2
FINAL: MOV AH, 4CH
       INT 21H
END START
```

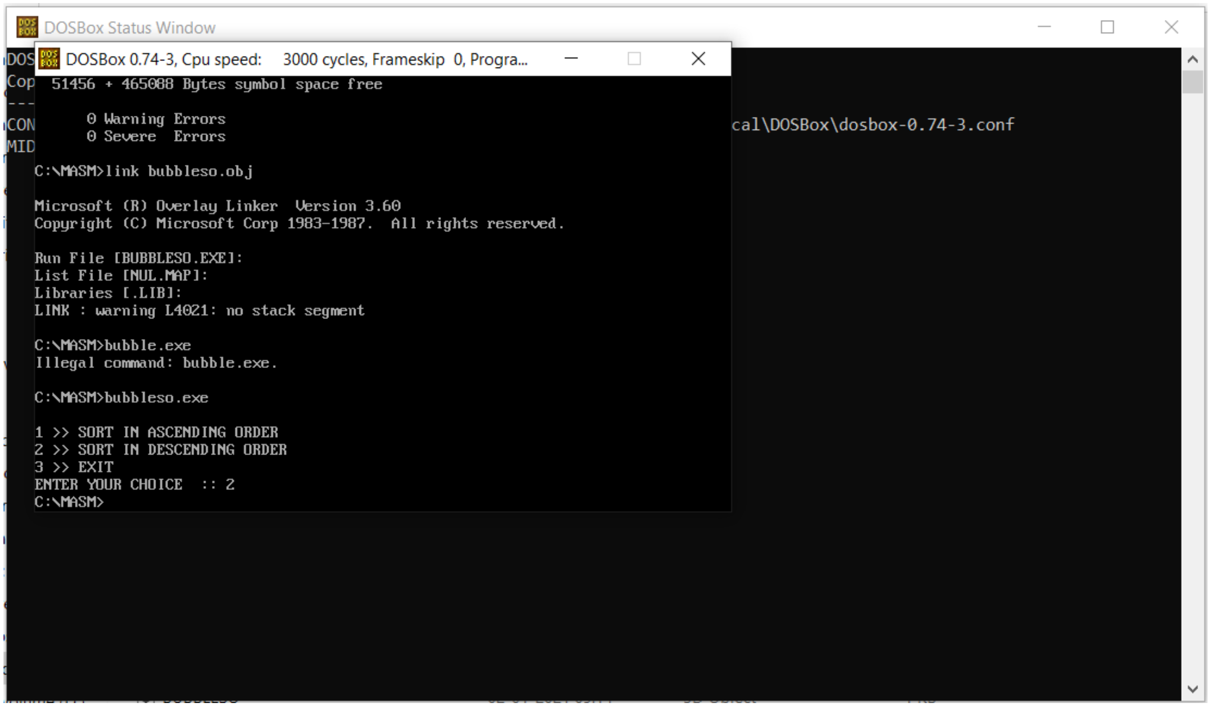
LAB PROGRAM 2

Source Code-

```
.MODEL
SMALL


         DISPLAY MACRO MSG
                 LEA DX, MSG
                 MOV AH, 09H
                 INT 21H
         ENDM


         .DATA
         LIST DB 02H, 01H, 34H, 0F4H, 09H, 05H
         NUMBER EQU $-LIST
         MSG1 DB 0DH, 0AH, "1 >> SORT IN ASCENDING ORDER$"
         MSG2 DB 0DH, 0AH, "2 >> SORT IN DESCENDING ORDER$"
         MSG3 DB 0DH, 0AH, "3 >> EXIT$"
         MSG4 DB 0DH, 0AH, "ENTER YOUR CHOICE  :: $"
         MSG5 DB 0DH, 0AH, "INVALID CHOICE ENTERED...$"


         .CODE
         START : MOV AX, @DATA
                 MOV DS, AX
                 LEA SI, LIST
                 MOV CH, NUMBER-1          ; CL STORES THE NUMBER OF ELEMENTS IN LIST
                 DISPLAY MSG1              ; DISPLAY THE MENU...
                 DISPLAY MSG2
                 DISPLAY MSG3
                 DISPLAY MSG4
                 MOV AH, 01H
                 INT 21H
                 SUB AL, 30H
                 CMP AL, 01H             ; INPUT=1? SORT IN ASCENDING ORDER
                 JE ASCSORT
                 CMP AL, 02H             ; INPUT=2? SORT IN DESCENDING ORDER
                 JE DESSORT
                 CMP AL, 03H             ; INPUT=3? EXIT
                 JE FINAL
                 DISPLAY MSG4
                 JMP FINAL
```

```
ASCSORT:MOV BL, 00H
AGAIN:  MOV SI, OFFSET LIST
        MOV CL, 00H             ; J VALUE
        MOV BH, CH
        SUB BH, BL              ; N-1-i
NPASS:  CMP CL, BH
        JNC NEXT
        MOV AL, [SI]
        MOV BP, 01H
        CMP AL, DS: [BP][SI]
        JC _NOPE
        XCHG AL, [SI+1]
        XCHG [SI], AL
_NOPE : INC CL
        INC SI
        JMP NPASS
NEXT:   INC BL
        CMP BL, CH
        JC AGAIN
        JMP FINAL


DESSORT:MOV BL, 00H
AGAIN1: MOV SI, OFFSET LIST
        MOV CL, 00H             ; J VALUE
        MOV BH, CH
        SUB BH, BL              ; N-1-i
NPASS1: CMP CL, BH
        JNC NEXT
        MOV AL, [SI]
        MOV BP, 01H
        CMP AL, DS: [BP][SI]
        JNC _NOPE1
        XCHG AL, [SI+1]
        XCHG [SI], AL
_NOPE1: INC CL
        INC SI
        JMP NPASS1
NEXT1:  INC BL
        CMP BL, CH
        JC AGAIN1
FINAL : MOV AH, 4CH
        INT 21H
END START
```

Screen shot-



DOSBox Status Window
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
 51456 + 465088 Bytes symbol space free

     0 Warning Errors
     0 Severe  Errors

C:\MASM>link bubbleso.obj

Microsoft (R) Overlay Linker  Version 3.60
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.

Run File [BUBBLESO.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\MASM>bubble.exe
Illegal command: bubble.exe.

C:\MASM>bubbleso.exe

1 >> SORT IN ASCENDING ORDER
2 >> SORT IN DESCENDING ORDER
3 >> EXIT
ENTER YOUR CHOICE  :: 2
C:\MASM>
```

```
cal\DOSBox\dosbox-0.74-3.conf
```

Writeup-

```
    JE FINAL
ASCORT : MOV BL, 00H
AGAIN : MOV SI, OFFSET LIST
        MOV CL, 00H      ; J VALUE
        MOV BH, CH
        SUB BH, BL       ; N-1-i
NPASS : CMP CL, BH
        JNC NEXT
        MOV AL, [SI]
        MOV BP, 01H
        CMP AL, DS: [BP][SI]
        JC _NOPE
        XCHG AL, [SI+1]
        XCHG [SI] , AL
_NOPE : INC CL
        INC SI
        JMP NPASS
NEXT : INC BL
        CMP BL,CH
        JC AGAIN
        JMP FINAL

DESCORT: MOV BL, 00H
AGAIN1: MOV SI, OFFSET LIST
        MOV CL, 00H      ; J VALUE
        MOV BH, CH
        SUB BH, BL
NPASS1: CMP CL, BH
        JNC NEXT
        MOV AL, [SI]
        MOV BP, 01H
        CMP AL, DS: [BP] [SI]
```

Date ___ / ___ / _____

```
    JNC _NOPE1
    XCHG AL, [SI+1]
    XCHG [SI], AL
_NOPE1: INC CL
        INC SI
        JMP NPASS1
NEXT1: INC BL
       CMP BL, CH
       JC AGAIN1
FINAL: MOV AH, 4CH
       INT 21H
END START
```

LAB PROGRAM 3

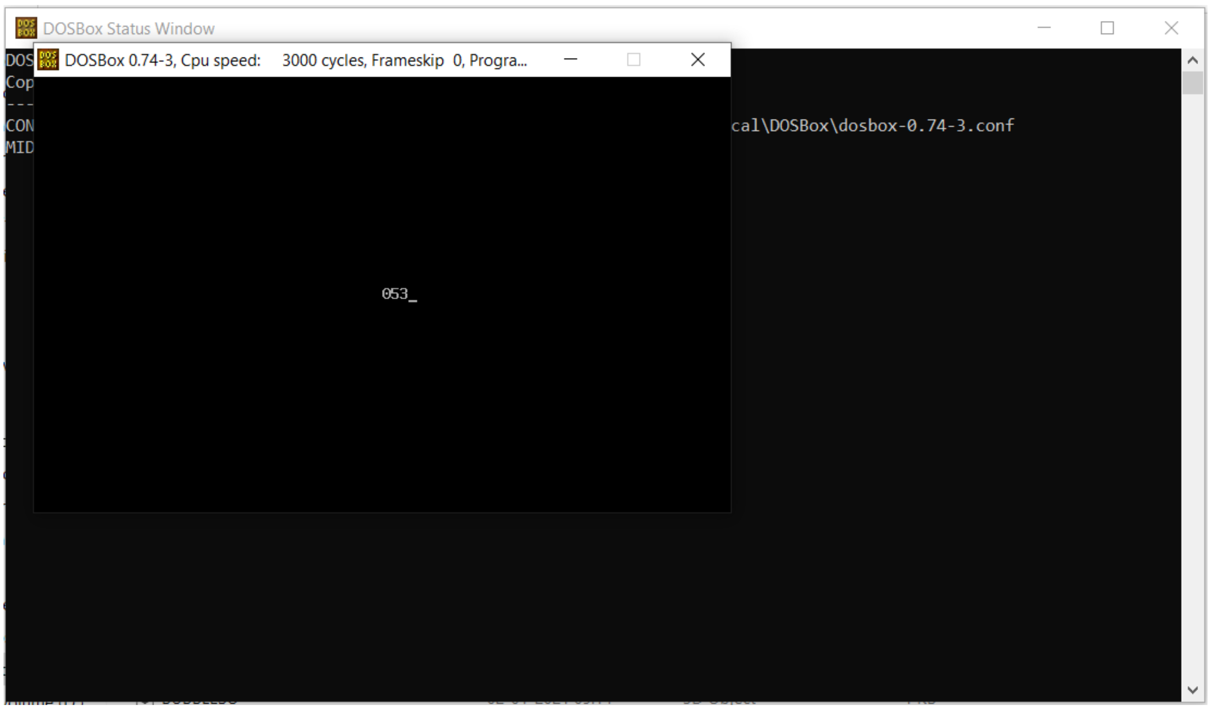
Source Code-

```
; PROGRAM
:: PROGRAM
TO READ AN
ALPHANUMERIC
CHARACTER
AND DISPLAY
ITS
               ; EQUIVALENT ASCII CODE AT THE CENTRE OF THE SCREEN


               .MODEL SMALL


               DISPLAY MACRO MSG
                       LEA DX, MSG
                       MOV AH, 09H
                       INT 21H
               ENDM


               ; MACRO TO DISPLAY A CHARACTER.
               DISPCHAR MACRO
                       MOV AH, 02H
                       INT 21H
               ENDM


               .DATA
               MSG1 DB 0DH, 0AH, "ENTER AN ALPHANUMERIC CHARACTER :: $"
               MSG2 DB 0DH, 0AH, "NOT AN ALPHANUMERIC CHARACTER...$"


               .CODE
               START : MOV AX, @DATA
                       MOV DS, AX
                       DISPLAY MSG1
                       MOV AH, 01H
                       INT 21H
                       CALL CHECK              ; CHECK FOR ALPHANUMERIC CHARACTER...
                       JC ERROR
                       PUSH AX
```

```
        ; SET MODE AND CLEAR THE SCREEN
        ; ROW =25 AND COLUMN = 80
        MOV AH, 00H
        MOV AL, 03H
        INT 10H
        ; MOVE THE CURSOR TO THE MID POINT OF SCREEN
        MOV AH, 02H
        MOV BH, 00H             ; PAGE NUMBER
        MOV DH, 12D             ; ROW VALUE
        MOV DL, 40D             ; COLUMN VALUE
        INT 10H
        POP AX                  ; RESTORE THE CHARACTER.
        AAM
        PUSH AX
        MOV AL, AH
        XOR AH, AH
        AAM
        ADD AX, 3030H
        MOV DL, AH
        PUSH AX
        DISPCHAR                ; DISPLAY THE ASCII VALUE
        POP AX
        MOV DL, AL
        DISPCHAR
        POP AX
        ADD AL, 30H
        MOV DL, AL
        DISPCHAR
        ; WAIT FOR USER TO PRESS ANY KEY
        MOV AH, 07H
        INT 21H
        ; FINISH ...JOB OVER
        JMP FINAL
ERROR : DISPLAY MSG2
        JMP FINAL


; THIS PROCEDURE CHECKS WHETHER THE INPUT IS ALPHANUMERIC OR NOT
CHECK PROC NEAR
        CMP AL, 30H
        JE FRET
        JC ERR
        CMP AL, 39H
        JE FRET
        JNC NEXT
        JC FRET
```

```
NEXT :  CMP AL, 41H
        JE FRET
        JC ERR
        CMP AL, 5AH
        JE FRET
        JNC NEXT1
        JC FRET
NEXT1 : CMP AL, 61H
        JE FRET
        JC ERR
        CMP AL, 7AH
        JE FRET
        JNC ERR
        JC FRET
ERR : STC                ; SET CARRY FOR ERROR
RET
FRET: CLC
RET
CHECK ENDP
; PROCEDURE ENDS HERE


FINAL : MOV AH, 4CH
        INT 21H
END START
```

Screen shot-



DOSBox Status Window

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
DOS
Cop
---
CON                                                    cal\DOSBox\dosbox-0.74-3.conf
MID
```

```
053_
```

Writeup-

MMC
ASCII
1BM19CS024

Date __/__/__

```
.MODEL SMALL
DISPLA  MACRO MSG
     LEA DX, MSG
     MOV AH, 09H
     INT 21H
ENDM
; MACRO TO DISPLAY A CHARACTER.
DISPCHAR  MACRO
        MOV AH, 02H
        INT 21H
ENDM

.DATA
MSG1 DB 0DH, 0AH, "ENTER AN ALPHANUMERIC CHARACTER: $"
MSG2 DB 0DH, 0AH, "NOT ALPHANUMERIC :... $"

.CODE
START: MOV AX,@DATA
        MOV DS, AX
        DISPLAY MSG1
        MOV AH, 01H
        INT 21H
        CALL CHECK  ;CHECK FOR ALPHANUMERIC CHARACTER
        JC ERROR
        PUSH AX
        ;SET MODE AND CLEAR THE SCREEN
        ;ROW = 25 AND COLUMN = 80
        MOV AH,00H
        MOV AL, 03H
        INT 10H
        ;MOVE CURSOR TO THE MID POINT OF THE SCREEN
        MOV AH,02H
```

Page No.

Date __/__/____

```
MOV BH,00H           ; PAGE NUMBER
MOV DH,12D           ; ROW VALUE
MOV DL,40D           ; COLUMN VALUE
INT 10H
POP AX            ; RESTORE THE CHARACTER
AAM
PUSH AX
MOV AL,AH
XOR AH,AH
AAM
ADD AX,3030H
MOV DL,AH
PUSH AX
DISPCHAR
POP AX
MOV DL,AL
DISPCHAR
POP AX
ADD AL,30H
MOV DL,AL
DISPCHAR
MOV AH,07H
INT 21H
JMP FINAL
ERROR: DISPLAY MSG2
       JMP FINAL
CHECK PROC NEAR
      CMP AL,30H
      JE FRET
      JC ERR
      CMP AL,39H
      JE FRET
```

Page No.

Date __/__/____

```
        JNC NEXT
        JC FRET
NEXT :  CMP AL, 41H
        JE FRET
        JC ERR
        CMP AL,5AH
        JE FRET
        JNC NEXT1
        JC FRET
NEXT1:  CMP AL,61H
        JE FRET
        JC ERR
        CMP AL,7AH
        JE FRET
        JNC ERR
        JC FRET
ERR: STC
RET
FRET: STC
RET
CHECK ENDP
FINAL : MOV AH,4CH
        INT 21H
END START
```

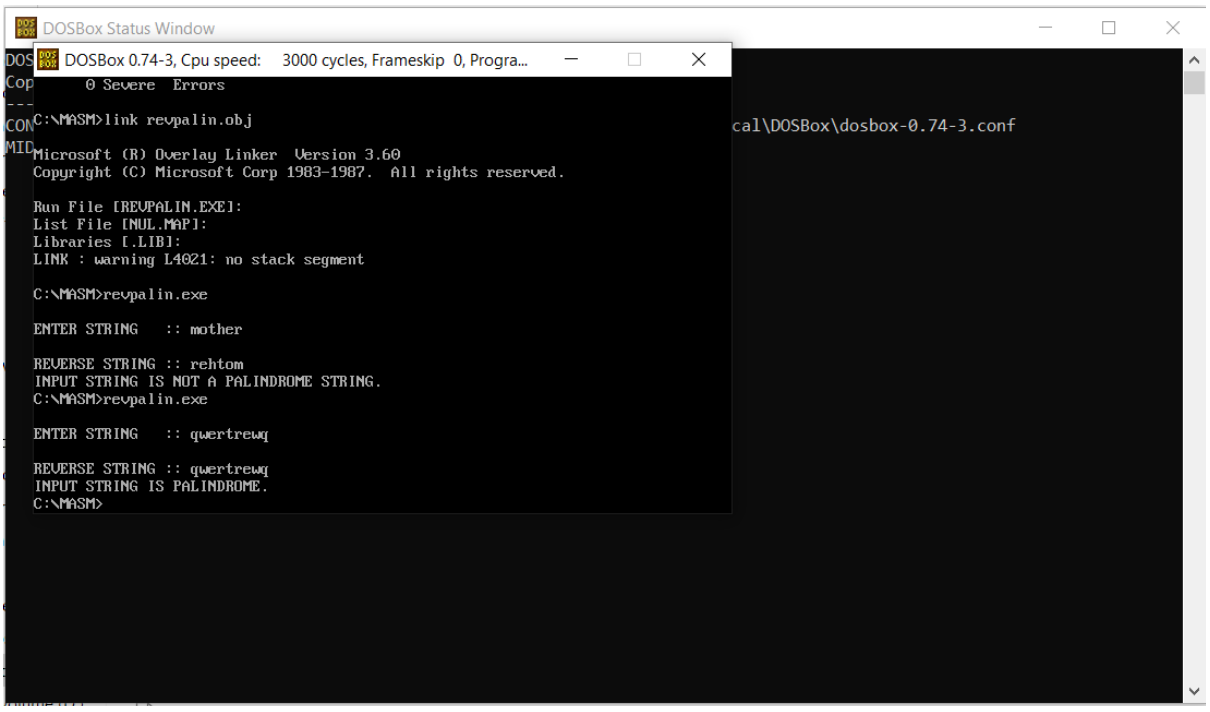
LAB PROGRAM 4

Source Code-

```
; PROGRAM
:: REVERSE
A GIVEN
STRING AND
CHECK
WHETHER IT
IS A
PALINDROME
             ; OR NOT


             .MODEL SMALL


             DISPLAY MACRO MSG
                     LEA DX, MSG
                     MOV AH, 09H
                     INT 21H
             ENDM


             .DATA
             MSG1 DB 0DH, 0AH, "ENTER STRING   :: $"
             MSG2 DB 0DH, 0AH, "REVERSE STRING :: $"
             MSG3 DB 0DH, 0AH, "INPUT STRING IS PALINDROME.$"
             MSG4 DB 0DH, 0AH, "INPUT STRING IS NOT A PALINDROME STRING.$"
             STRING DB 80H DUP(?)
             RSTRING DB 80H DUP(?)


             .CODE
             START : MOV AX, @DATA
                     MOV DS, AX
                     DISPLAY MSG1
                     ; TAKE THE STRING FROM KEYBOARD CHARACTER BY CHARACTER
                     MOV SI, OFFSET STRING
                     XOR CL, CL
             AGAIN:  MOV AH, 01H
                     INT 21H
                     CMP AL, 0DH
                     JE NEXT
```

```
        MOV [SI], AL
        INC SI
        INC CL
        JMP AGAIN
NEXT :  MOV [SI], BYTE PTR '$'
        ; STRING INPUT OVER....
        DEC SI
        MOV CH, CL
        ; REVERSE THE STRING AND STORE IN RSTRING
        MOV DI, OFFSET RSTRING
BACK:   MOV AL, [SI]
        MOV [DI], AL
        DEC SI
        INC DI
        DEC CH
        JNZ BACK
        MOV [DI], BYTE PTR '$'
        DISPLAY MSG2
        DISPLAY RSTRING
        MOV SI,OFFSET STRING
        MOV DI, OFFSET RSTRING
AG:     MOV AL, [SI]
        CMP AL, [DI]
        JNE FAIL
        INC SI
        INC DI
        DEC CX
        JZ SUCCESS
        JMP AG
FAIL:   DISPLAY MSG4
        JMP FINAL
SUCCESS:DISPLAY MSG3
FINAL:  MOV AH, 4CH
        INT 21H
END
```

Screen shot-



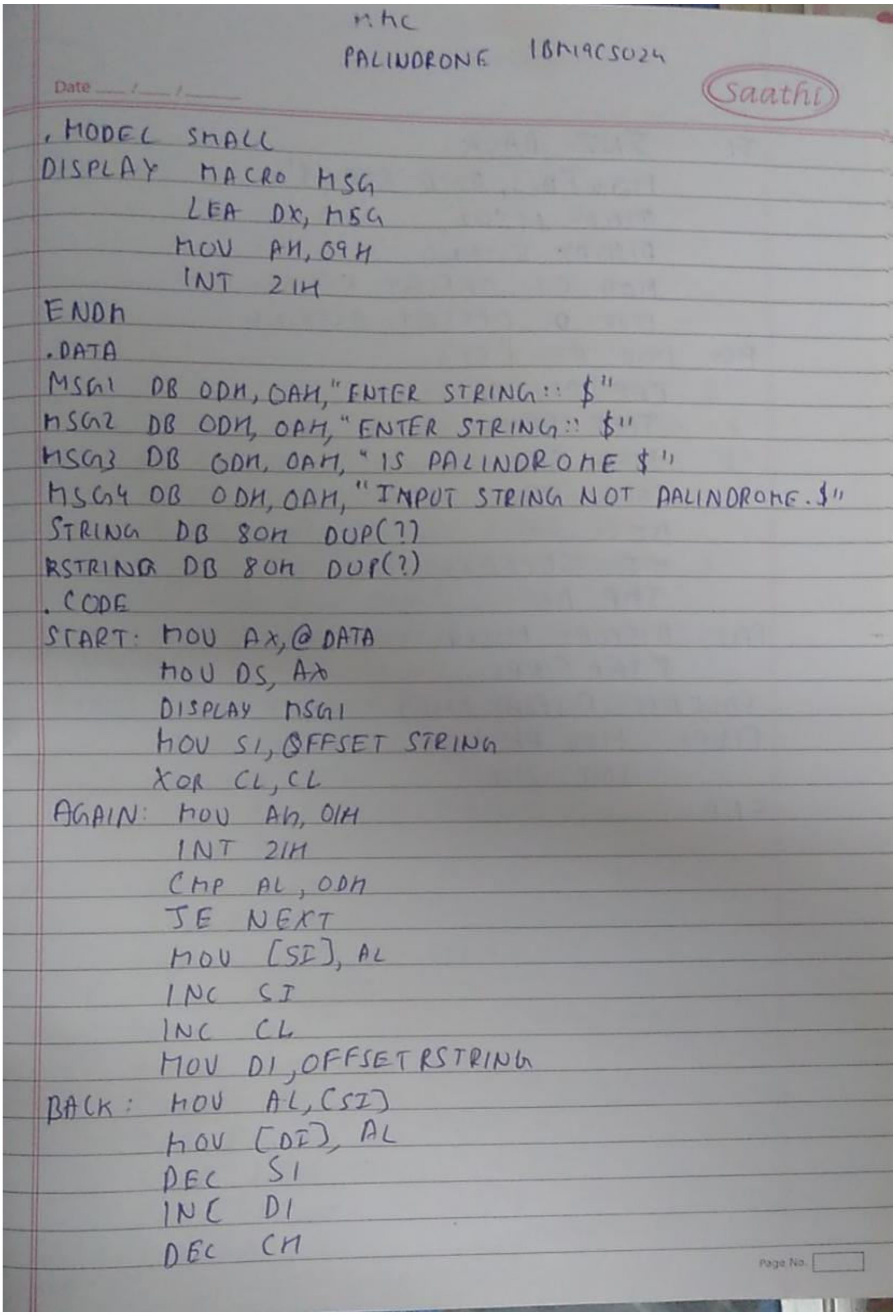
DOSBox Status Window

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
      0 Severe  Errors

C:\MASM>link revpalin.obj

Microsoft (R) Overlay Linker  Version 3.60
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.

Run File [REVPALIN.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\MASM>revpalin.exe

ENTER STRING   :: mother

REVERSE STRING :: rehtom
INPUT STRING IS NOT A PALINDROME STRING.
C:\MASM>revpalin.exe

ENTER STRING   :: qwertrewq

REVERSE STRING :: qwertrewq
INPUT STRING IS PALINDROME.
C:\MASM>
```

```
cal\DOSBox\dosbox-0.74-3.conf
```

Writeup-



M.MC
PALINDRONE 1BM19CS024

Date ___ / ___ / ______

Saathi

```
.MODEL SMALL
DISPLAY  MACRO MSG
         LEA DX, MSG
         MOV AH, 09H
         INT 21H
ENDM
.DATA
MSG1 DB ODH, OAH, "ENTER STRING:: $"
MSG2 DB ODH, OAH, "ENTER STRING:: $"
MSG3 DB ODH, OAH, "IS PALINDROME $"
MSG4 DB ODH, OAH, "INPUT STRING NOT PALINDROME.$"
STRING DB 80H DUP(?)
RSTRING DB 80H DUP(?)
.CODE
START: MOV AX,@DATA
       MOV DS, AX
       DISPLAY MSG1
       MOV SI, OFFSET STRING
       XOR CL, CL
AGAIN: MOV AH, 01H
       INT 21H
       CMP AL, ODH
       JE NEXT
       MOV [SI], AL
       INC SI
       INC CL
       MOV DI, OFFSETRSTRING
BACK:  MOV AL, [SI]
       MOV [DI], AL
       DEC SI
       INC DI
       DEC CH
```

Page No.

Date __/__/__

```
~~$1~~   JNZ BACK
         MOV [DI], BYTE PTR '$'
         DISPLAY MSG2
         DISPLAY RSTRING
         MOV SI, OFFSET STRING
         MOV DI, OFFSET RSTRING
AG: MOV AL, [SI]
        CMP AL, [DI]
        JNE FAIL
        INC SI
        INC DI
        DEC CX
        JZ SUCCESS
        JMP AG
FAIL: DISPLAY MSG4
        ~~F~~ JMP FINAL
SUCCESS: DISPLAY MSG3
FINAL : MOV AH, 4CH
           INT 21H
END.
```

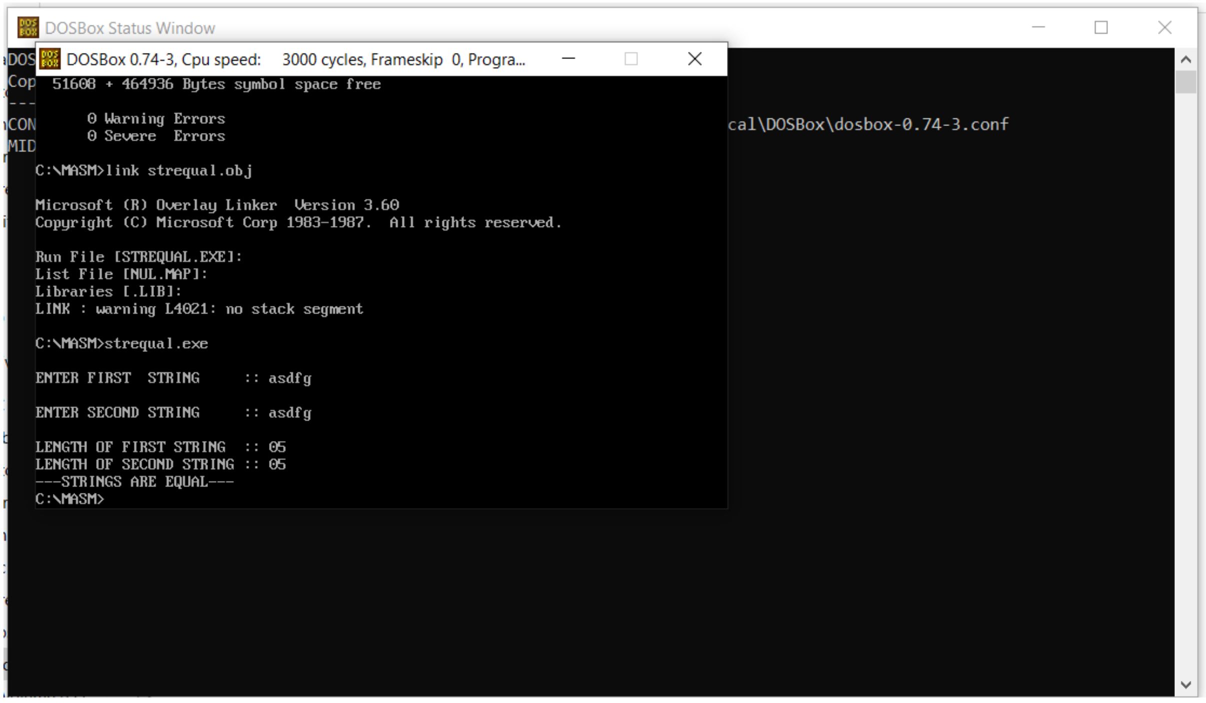
LAB PROGRAM 5

Source Code-

```
;
PROGRAM
::
PROGRAM
TO READ
TWO
STRING
AND TO
CHECK
WHETHER
THEY
ARE
          ; EQUAL OR NOT AND DISPLAY APPROPRIATE MESSAGES. ALSO DISPLAY THE
          ; LENGTH OF THE STRING


          ; LOGIC :: TAKE THE INPUT... CALCULATE THE LENGTH.. CHECK WHETHER THE
          ; LENGTHS ARE EQUAL OR NOT.. IF NOT, THEN STRINGS ARE NOT EQUAL
          ; IF YES, COMPARE CHARACTER BY CHARACTER....


          .MODEL SMALL


          DISPLAY MACRO MSG
                  LEA DX, MSG
                  MOV AH, 09H
                  INT 21H
          ENDM


          .DATA
          MSG1 DB 0DH, 0AH, "ENTER FIRST  STRING     :: $"
          MSG2 DB 0DH, 0AH, "ENTER SECOND STRING     :: $"
          MSG3 DB 0DH, 0AH, "LENGTH OF FIRST STRING  :: $"
          MSG4 DB 0DH, 0AH, "LENGTH OF SECOND STRING :: $"
          MSG5 DB 0DH, 0AH, "---STRINGS ARE EQUAL---$"
          MSG6 DB 0DH, 0AH, "---STRINGS ARE NOT EQUAL---$"
          STRING1 DB 80H DUP(?)
          STRING2 DB 80H DUP(?)
```

```
.CODE
START : MOV AX, @DATA
        MOV DS, AX
        DISPLAY MSG1
        MOV SI, OFFSET STRING1
        CALL READSTR
        MOV BL, CL              ; STORE THE LENGTH OF FIRST STRING
        DISPLAY MSG2
        MOV SI, OFFSET STRING2
        CALL READSTR
        PUSH BX
        PUSH CX
        DISPLAY MSG3
        MOV AL, BL
        CALL LEN_DIS
        DISPLAY MSG4
        MOV AL, CL
        CALL LEN_DIS
        POP CX
        POP BX
        CMP CL, BL              ; COMPARE THE LENGTHS
        JNE FAIL                ; IF LENGTHS ARE EQUAL, PROCESS NEXT STATMENT
        MOV SI, OFFSET STRING1
        MOV DI, OFFSET STRING2
        CLD
CHK:    MOV AL, [SI]            ; COMPARE BOTH THE STRING
        CMP AL, [DI]
        JNE FAIL
        INC SI
        INC DI
        DEC CL
        JNZ CHK
        DISPLAY MSG5
        JMP FINAL


LEN_DIS PROC NEAR
        XOR AH, AH
        ADD AL, 00H
        AAM
        ADD AX, 3030H
        MOV BH, AL
        MOV DL, AH
        MOV AH, 02H
        INT 21H
        MOV DL, BH
        MOV AH, 02H
        INT 21H
RET
LEN_DIS ENDP


READSTR PROC NEAR
        XOR CL, CL
BACK:   MOV AH, 01H
        INT 21H
        CMP AL, 0DH
        JE FINISH
        MOV [SI], AL
        INC SI
        INC CL
        JMP BACK
FINISH: MOV [SI], BYTE PTR '$'
        RET
READSTR ENDP


FAIL:   DISPLAY MSG6
FINAL:  MOV AH, 4CH
        INT 21H
END START
```

Screen shot-



DOSBox Status Window

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
 51608 + 464936 Bytes symbol space free

     0 Warning Errors
     0 Severe  Errors

C:\MASM>link strequal.obj

Microsoft (R) Overlay Linker  Version 3.60
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.

Run File [STREQUAL.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\MASM>strequal.exe

ENTER FIRST  STRING     :: asdfg

ENTER SECOND STRING     :: asdfg

LENGTH OF FIRST STRING  :: 05
LENGTH OF SECOND STRING :: 05
---STRINGS ARE EQUAL---
C:\MASM>
```

```
cal\DOSBox\dosbox-0.74-3.conf
```

Writeup-

HAC

STRING 1BM19CS024

Saathi

Date __/__/____

```
.MODEL SMALL
DISPLAY MACRO MSG
        LEA DX, MSG
        MOV AH, 09H
        INT 21H
ENDM
.DATA
MSG1 DB 0DH, 0AH, "ENTER FIRST STRING: $"
MSG2 DB 0DH, 0AH, "ENTER SECOND STRING: $"
MSG3 DB 0DH, 0AH, "LENGTH OF FIRST STRING: $"
MSG4 DB 0DH, 0AH, "LENGTH OF SECOND STRING: $"
MSG5 DB 0DH, 0AH, "STRINGS EQUAL $"
MSG6 DB 0DH, 0AH "STRINGS NOT EQUAL $"
STRING1 DB 80H DUP(?)
STRING2 DB 80H DUP(?)
.CODE
START: MOV AX, @DATA
        MOV DS, AX
        DISPLAY MSG1
        MOV SI, OFFSET STRING1
        CALL READSTR
        MOV BL, CL
        DISPLAY MSG2
        MOV SI, OFFSET STRING2
        CALL READSTR
        PUSH BX
        PUSH CX
        DISPLAY MSG3
        MOV AL, BL
        CALL LEN_DIS
        POP CX
        POP BX
```

Page No.

```
     CMP CL,BL
     JNE  FAIL
     MOV  SI, OFFSET STRING1
     MOV  DI, OFFSET STRING2
     CLD
CHK:  MOV  AL, [SI]
      CMP  AL, [DI]
      JNE  FAIL
      INC  SI
      INC  DI
      DEC  CL
      JNZ  CHK
      DISPLAY MSG5
      JMP  FINAL
LEN_DIS  PROC NEAR
      XOR  AH,AH
      ADD  AL, 00H
      AAM
      ADD  AX, 3030H
      MOV  BH, AL
      MOV  DL, AH
      MOV  AH, 02H
      INT  21H
      MOV  DL, BH
      MOV  AH, 02H
      INT  21H
 RET
 LEN_DIS ~~ED~~ENDP
 READSTR PROC NEAR
      XOR CL,CL
 BACK: MOV AH, 01H
      INT 21H
```

```
        CMP AL, ODH
        JE FINISH
        MOV [SI], AL
        INC SI
        INC CL
        JMP BACK
FINISH: MOV [SI], BYTE PTR '$'
        RET
READSTR ENDP
FAIL: DISPLAY MSG6
FINAL: MOV AH, 4CH
        INT 21H
END START
```

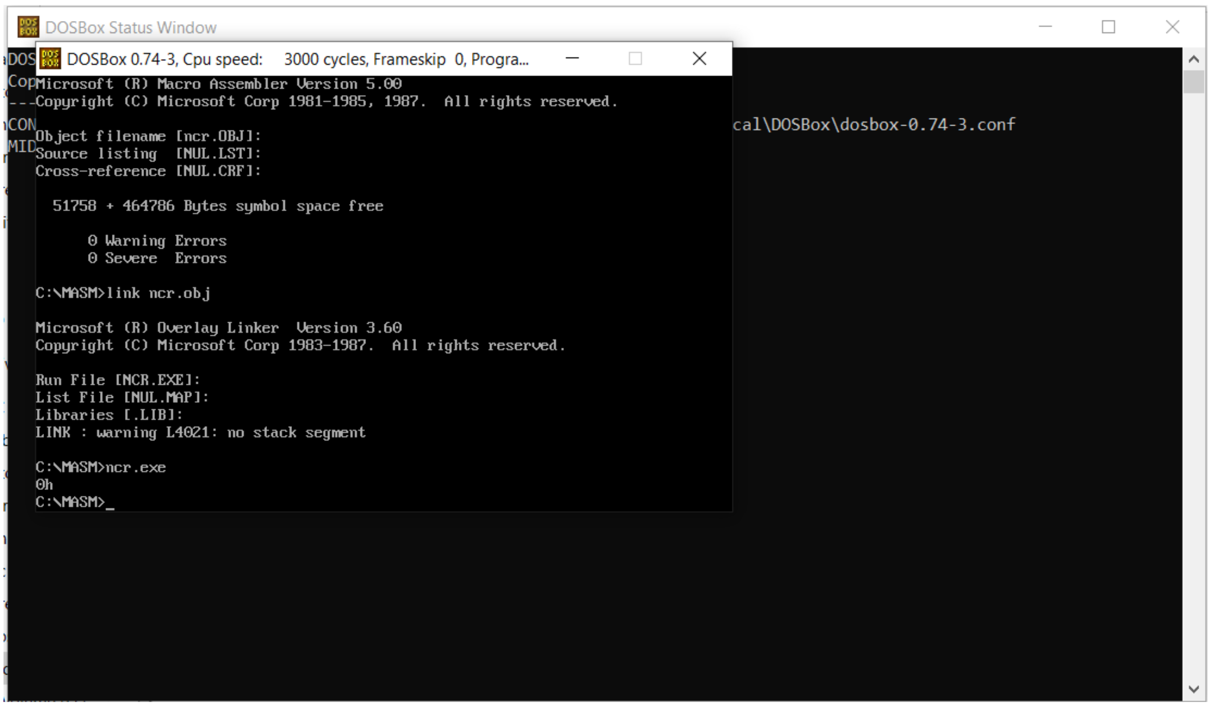
LAB PROGRAM 6

Source Code-

```
.model
small
         .data
         n dw 8
         r dw 3
         ncr dw 0


         .code
         mov ax,@data
         mov ds,ax


         mov ax,n
         mov bx,r
         call ncrpro
         call disp
         jmp final


         ncrpro  proc near
                 cmp ax,bx  ;r=n
                 je res1
                 cmp bx,0  ;r=0
                 je res1                                ; 3c2 +3c1
                 cmp bx,1   ;r=1
                 je resn
                 dec ax      ;r=n-1
                 cmp bx,ax
                 je incr
                 push ax
                 push bx
                 call ncrpro
         pop bx
         pop ax
         dec bx
         push ax
         push bx
         call ncrpro
         pop bx
         pop ax
```

```
ret


res1:inc ncr
ret


incr:inc ncr
resn:add ncr,ax  ;1+2 3+3=6
ret
ncrpro endp


disp proc near
  mov bx,ncr
  add bx,3030h
  mov  dl,bh
  mov ah,02h
  int 21h
  mov dl,bl
  mov ah,02h
  int 21h
  ret
  disp endp


final: mov ah,4ch
     int 21h
     end
```

Screen shot-



```
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [ncr.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  51758 + 464786 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\MASM>link ncr.obj

Microsoft (R) Overlay Linker  Version 3.60
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.

Run File [NCR.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\MASM>ncr.exe
0h
C:\MASM>_
```

Writeup-

MMC

NCR 1BM19CS024

Date ___/___/___

```
.model small
.data
n dw 8
r dw 3
ncr dw 0
.code
mov ax,@data
mov ds,ax
mov ax,n
mov bx,r
call ncrpro
call disp
jmp final
ncrpro proc near
        cmp ax,bx
        je res1
        cmp bx,0
        je res1
        cmp bx,1
        je resn
        dec ax
        cmp bx,ax
        je incr
        push ax
        push bx
        call ncrpro
pop bx
pop ax
dec bx
push ax
push bx
call ~~mc~~ ncrpro
```

```
pop bx
pop ax
ret
rec1: inc ncr
      ret
incr : inc ncr
recn : add ncr, ax
       ret
ncrpro endp

disp proc near
     mov bx, ncr
     add bx, 3030h
     mov dl, bh
     mov ah, 02h
     int 21h
     mov dl, bl
     mov ah, 02h
     int 21h
     ret
     disp endp
final: mov ah, 4ch
       int 21h
       end
```

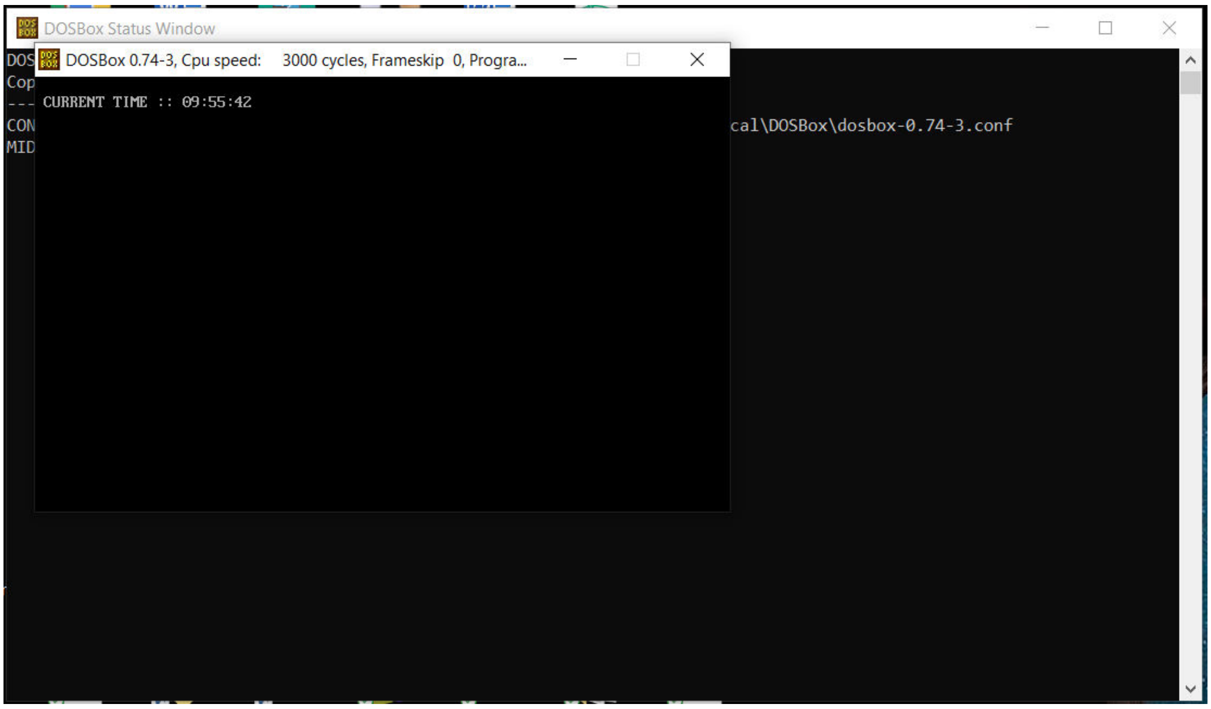
LAB PROGRAM 7

Source Code-

```
;
PROGRAM
:: READ
THE
CURRENT
TIME
FROM
THE
SYSTEM
AND
DISPLAY
IT IN
THE
          ; STANDARD FORMAT ON THE SCREEN


          .MODEL SMALL


          DISPLAY MACRO MSG
                  LEA DX, MSG
                  MOV AH, 09H
                  INT 21H
          ENDM


          .DATA
          TIMESTR DB 020H DUP(?)
          MSG1 DB "CURRENT TIME :: $"


          .CODE
          START : MOV AX, @DATA
                  MOV DS, AX
          ; CLEAR THE SCREEN
                  MOV AH, 00H
                  MOV AL, 03H
                  INT 10H
          ; SET A PARTICULAR LOCATION. FOR DYNAMIC CLOCK
          AG:     MOV BH, 00H
                  MOV DH, 01H
```

```
        MOV DL, 01H
        MOV AH, 02H
        INT 10H
        MOV SI, OFFSET TIMESTR
        MOV AH, 2CH             ; INTERRUPT FOR GETTING SYSTEM TIME
        INT 21H
        MOV AL, CH              ; CH=HOUR, CL=MINUTES, DH=SECOND
        AAM
        ADD AX, 3030H
        MOV [SI], AH
        INC SI
        MOV [SI], AL
        INC SI
        MOV [SI], BYTE PTR ':'
        INC SI
        MOV AL, CL
        AAM
        ADD AX, 3030H
        MOV [SI], AH
        INC SI
        MOV [SI], AL
        INC SI
        MOV [SI], BYTE PTR ':'
        INC SI
        MOV AL, DH
        AAM
        ADD AX, 3030H
        MOV [SI], AH
        INC SI
        MOV [SI], AL
        INC SI
        MOV [SI], BYTE PTR '$'
        DISPLAY MSG1
        DISPLAY TIMESTR                 ; DISPLAY THE TIME...
; CHECK FOR THE KEYBOARD STATUS....
; IF KEY IS PRESSED, TERMINATE THE PROGRAM..
        MOV AH, 0BH
        INT 21H
        CMP AL, 00H
        JE AG
FINAL : MOV AH, 4CH
        INT 21H
END START
```

Screen shot-


DOSBox Status Window
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
DOS
Cop
---  CURRENT TIME :: 09:55:42
CON                                                                    cal\DOSBox\dosbox-0.74-3.conf
MID
```

Writeup-

MMC
TIME 1BM19CS024

Date __/__/____

```
.MODEL SMALL
DISPLAY MACRO MSG
        LEA DX, MSG
        MOV AH, 09H
        INT 21H
ENDM
.DATA
TIMESTER DB 020H DUP(?)
MSG1 DB "CURRENT TIME : $"
.CODE
START: MOV AX,@DATA
       MOV DS,AX
       MOV AL, 03H
       INT 10H
       MOV SI, OFFSET TIMESTR
       MOV AH, 2CH
       INT 21H
       MOV AL, CH
       AAM
       ADD AX, 3030H
       MOV [SI], AH
       INC SI
       MOV [SI], BYTE PTR ':'
       INC SI
       MOV AL, CL
       AAM
       ADD AX, 3030H
       MOV [SI], AH
       INC SI
       MOV [SI], AL
       INC SI
       MOV AL, DH
```

```
      AAM
      ADD AX, 3030H
      MOV [SI], AH
      INC SI
      MOV [SI], AL
      INC SI
      MOV [SI], BYTE PTR '$'
       DISPLAY MSG1
       DISPLAY TIMESTR
       MOV AH, 0BH
       INT 21H
        CMP AL, 00H
        JE AG
FINAL  MOV AH, 4CH
        INT 21H
END START
```

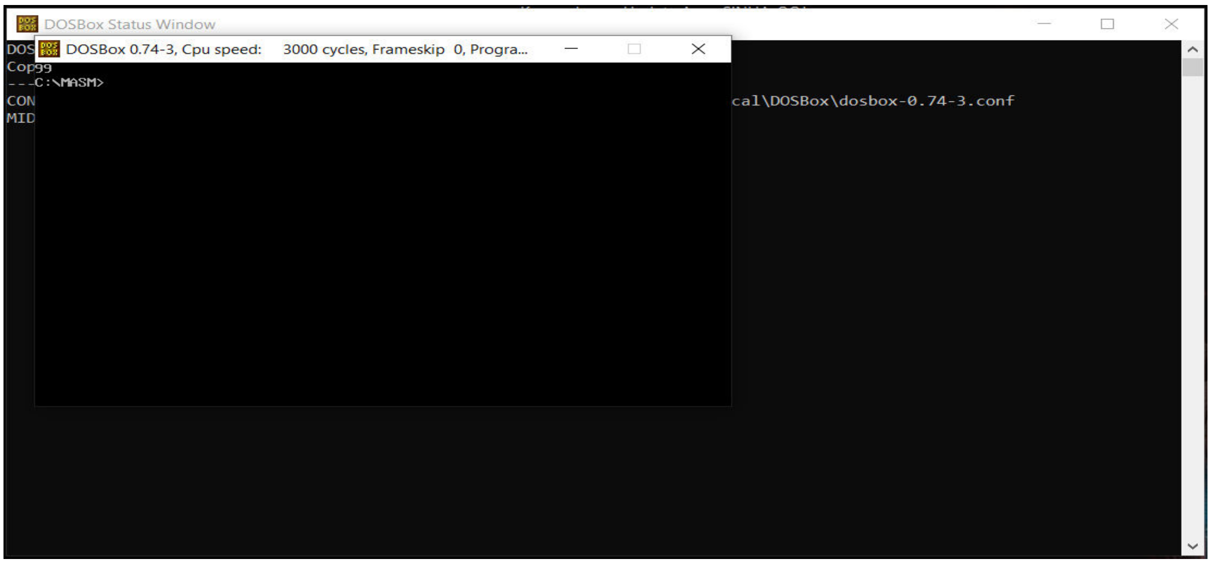
LAB PROGRAM 8

Source Code-

```
;
PROGRAM
::
PROGRAM
TO
SIMULATE
A
DECIMAL
UP
COUNTER
TO
DISPLAY
00-99


           .MODEL SMALL


           .CODE
           START : MOV CL, 00H
           ; CLEAR THE SCREEN FIRST
                   MOV AH, 00H
                   MOV AL, 03H
                   INT 10H


           BACK:   MOV BH, 00H
                   MOV DH, 00H            ; SET ROW
                   MOV DL, 00H            ; SET COLUMN
                   MOV AH, 02H
                   INT 10H
                   MOV AL, CL
                   ADD AL,00H
                   AAM
                   ADD AX, 3030H
                   MOV CH, AL
                   MOV AL, AH
                   MOV DL, AL
                   MOV AH, 02H
                   INT 21H
                   MOV AL, CH
```

```
        MOV DL, AL
        MOV AH, 02H
        INT 21H
        CALL DELAY
        INC CL
        XOR AX, AX
        CMP CL, 100D
        JNE BACK
        JE FINAL


DELAY PROC NEAR
        PUSH CX
        PUSH AX
        PUSH BX
        MOV CX, 0FFH
AG:     MOV BX, 0FFH
AG1:    NOP
        XOR AX, AX
        DEC BX
        JNZ AG1
        DEC CX
        JNZ AG
        POP BX
        POP AX
        POP CX
RET
DELAY ENDP
FINAL:  MOV AH, 4CH
        INT 21H
END START
```

Screen shot-



```
DOSBox Status Window
DOS
Cop99
---C:\MASM>
CON                                                     cal\DOSBox\dosbox-0.74-3.conf
MID
```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

Writeup-

MMC
UPCOUNTER 1BM19CS024

Saathi

Date ___/___/___

```
.MODEL SMALL
.CODE
START: MOV CL,00H
       MOV AH,00H
       MOV AL,03H
       INT 10H
BACK:  MOV BH,00H
       MOV DH,00H
       MOV DL,00H
       MOV AH,02H
       INT 10H
       MOV AL,CL
       ADD AL,00H
       AAM
       ADD AX,3030H
       MOV CH,AL
       MOV AL,AH
       MOV DL,AL
       MOV, AH,02H
       INT 21H
       MOV AL,CH
       MOV DL,AL
       MOV AH,02H
       INT 21H
       CALL DELAY
       INC CL
       XOR AX,AX
       CMP CL,100D
       JNE BACK
       JE FINAL
```

Date ___ /___ /_____

```
DELAY PROC NEAR
        PUSH CX
        PUSH AX
        PUSH BX
        MOV CX,OFFH
AG: MOU BX, OFFH
AG1: FNOP
        XOR AX, AX
        DEC BX
        JNZ AG1
        DEC CX
        JNZ AG
        POP BX
        POP AX
        POP CX
 RET
 DELAY ENDP
 FINAL : MOV AH, 4CH
         INT 21H
 END START
```

Page No.

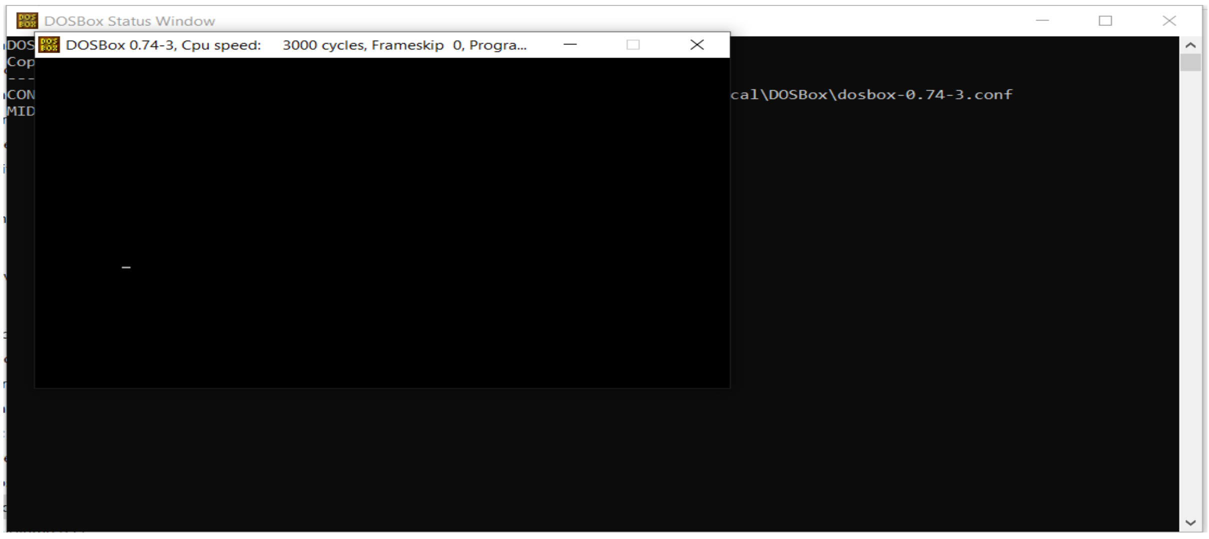
LAB PROGRAM 9

Source Code-

```
; PROGRAM
:: READ A
PAIR OF
INPUT CO-
ORDINATES
IN BCD
AND MOVE
THE
CURSOR
            ; TO THE SPECIFIED LOCATION ON THE SCREEN


            ; RESTERICTION :: PLEASE ENTER THE ROW AND COLUMN IN TWO DIGITS
            ; AS 00 OR 34


            .MODEL SMALL


            DISPLAY MACRO MSG
                    LEA DX, MSG
                    MOV AH, 09H
                    INT 21H
            ENDM


            .DATA
            ROW DB 02H DUP(?)
            COLUMN DB 02H DUP(?)
            MSG1 DB 0DH, 0AH, "ENTER THE X CO-ORDINATE (ROW)    :: $"
            MSG2 DB 0DH, 0AH, "ENTER THE Y CO-ORDINATE (COLUMN) :: $"


            .CODE
            START : MOV AX, @DATA
                    MOV DS, AX
                    DISPLAY MSG1
                    MOV SI, OFFSET ROW
                    CALL READ
                    DISPLAY MSG2
                    MOV SI, OFFSET COLUMN
```

```
        CALL READ
        MOV SI, OFFSET ROW
        MOV AH, [SI]
        INC SI
        MOV AL, [SI]
        SUB AX, 3030H
        AAD
        MOV DH, AL
        MOV SI, OFFSET COLUMN
        MOV AH, [SI]
        INC SI
        MOV AL, [SI]
        SUB AX, 3030H
        AAD
        MOV DL, AL
        MOV AH, 00H
        MOV AL, 03H
        INT 10H
        MOV AH, 02H
        INT 10H
        JMP FINAL
READ PROC NEAR
        MOV CX, 02H
BACK:   MOV AH, 01H
        INT 21H
        MOV [SI], AL
        INC SI
        DEC CX
        JNZ BACK
RET
READ ENDP
FINAL : MOV AH, 01H
        INT 21H
        MOV AH, 4CH
        INT 21H
END START
```

Screen shot-



DOSBox Status Window

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
DOS
Cop
---
CON                                                                cal\DOSBox\dosbox-0.74-3.conf
MID
```

Writeup-

MMC
BCD   1BM19CS024

Date __/__/____

```
.MODEL SMALL
DISPLAY MACRO MSG
        LEA DX,MSG
        MOV AH,09H
        INT 21H
ENDM
.DATA
ROW DB 02H DUP(?)
COLUMN DB 02H DUP(?)
MSG1 DB 0DH,0AH,"Enter The x co-ordinate(Row): $"
MSG2 DB 0DH,0AH,"Enter The y co-ordinate(column): $"

.CODE
START: MOV AX,@DATA
       MOV DS,AX
       DISPLAY MSG1
       MOV SI,OFFSET ROW
       CALL READ
       DISPLAY MSG2
       MOV SI,OFFSET COLUMN
       CALL READ
       MOV SI,OFFSET ROW
       MOV AH,[SI]
       INC SI
       MOV AL,[SI]
       SUB AX,3030H
       AAD
       MOV DL,AL
       MOV AH,00H
       MOV AL,03H
       INT 10H
       MOV AH,02H
```

```
        INT  10H
        JMP FINAL
READ  PROC  NEAR
        MOV  CX, 02H
BACK:  MOV  AH, 01H
         INT 21H
         MOV  [SI], AL
         INC SI
         DEC  CX
         JNZ  BACK
RET
~~REOP~~ READ  ENDP
FINAL: MOV AH, 01H
          INT 21H
          MOV  AH, 4CH
          INT 21H
END START
```

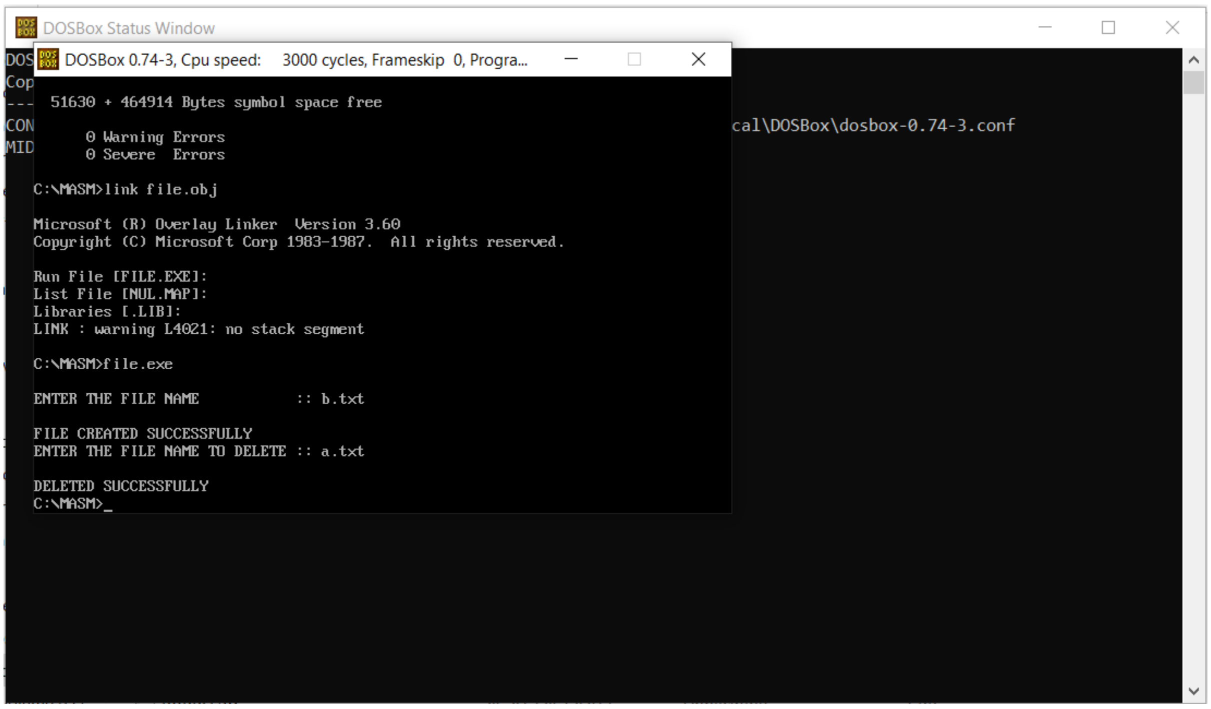
LAB PROGRAM 10

Source Code-

```
;
PROGRAM
::
PROGRAM
TO
CREATE
A FILE
(INPUT
FILE)
AND TO
DELETE
AN
          ; EXISTING FILE...


          .MODEL SMALL


          DISPLAY MACRO MSG
                  LEA DX, MSG
                  MOV AH, 09H
                  INT 21H
          ENDM


          .DATA
          MSG1 DB 0DH, 0AH, "ENTER THE FILE NAME           :: $"
          MSG2 DB 0DH, 0AH, "FILE CREATED SUCCESSFULLY  $"
          MSG3 DB 0DH, 0AH, "CREATION FAILED.$"
          MSG4 DB 0DH, 0AH, "ENTER THE FILE NAME TO DELETE :: $"
          MSG5 DB 0DH, 0AH, "DELETED SUCCESSFULLY $"
          MSG6 DB 0DH, 0AH, "DELETION FAILED $"
          FNAME DB 40H DUP(?)
          FNAME2 DB 40H DUP(?)


          .CODE
          START : MOV AX, @DATA
                  MOV DS, AX
                  DISPLAY MSG1
                  MOV SI, OFFSET FNAME
```

```
BACK:   MOV AH, 01H
        INT 21H
        CMP AL, 0DH
        JE NEXT
        MOV [SI], AL
        INC SI
        JMP BACK
NEXT  : MOV [SI], BYTE PTR '$'
        LEA DX, FNAME
        MOV CX, 00H
        MOV AH, 3CH             ; INTERRUPT FOR FILE CREATION
        INT 21H
        JC FAILED
        DISPLAY MSG2
        JMP NEXT1
FAILED: DISPLAY MSG3
NEXT1:  DISPLAY MSG4
        MOV SI, OFFSET FNAME2
BACK1:  MOV AH, 01H
        INT 21H
        CMP AL, 0DH
        JE NEXT2
        MOV [SI], AL
        INC SI
        JMP BACK1
NEXT2 : MOV [SI], BYTE PTR '$'
        LEA DX, FNAME2
        MOV AH, 41H             ; INTERRUPT FOR FILE DELETION
        INT 21H
        JC DFAIL
        DISPLAY MSG5
        JMP FINAL
DFAIL : DISPLAY MSG6
FINAL : MOV AH, 4CH
        INT 21H
END START
```

Screen shot-



DOSBox Status Window

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...

```
 51630 + 464914 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors

C:\MASM>link file.obj

Microsoft (R) Overlay Linker  Version 3.60
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.

Run File [FILE.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\MASM>file.exe

ENTER THE FILE NAME            :: b.txt

FILE CREATED SUCCESSFULLY
ENTER THE FILE NAME TO DELETE :: a.txt

DELETED SUCCESSFULLY
C:\MASM>_
```

```
DOS
Cop
---
CON                                                                    cal\DOSBox\dosbox-0.74-3.conf
MID
```

Writeup-

MMC
FILE 1BM19CS024

Date ___ / ___ / ____

```
.MODEL SMALL
DISPLAY MACRO MSG
        LEA DX, MSG
        MOV AH, 09H
        INT 21H
ENDM

.DATA
MSG1 DB 0DH, 0AH, "ENTER FILE NAME : $"
MSG2 DB 0DH ,0AH, " FILE CREATED SUCCESSFULLY $"
MSG3 DB 0DH, 0AH, "CREATION FAILED $"
MSG4 DB 0DH, 0AH, "ENTER FILE NAME TO DELETE: $"
MSG5 DB 0DH, 0AH, "DELETION SUCCESSFUL $"
MSG6 DB 0DH, 0AH, "DELETION FAILED $"
FNAME DB 40H DUP(?)
FNAME2 DB 40H DUP(?)

.CODE
START: MOV AX, @DATA
        MOV DS, AX
        DISPLAY MSG1
        MOV SI, OFFSET FNAME
BACK: MOV AH, 01H
        INT 21H
        CMP AL, 0DH
        JE NEXT
        MOV [SI], AL
        INC SI
        JMP BACK
NEXT: MOV [SI] BYTE PTR '$'
        LEA DX, FNAME
        MOV CX, 00H
```

Date ___ / ___ / _____

```
        MOV AH, 3CH
        INT 21H
        JC FAILED
        DISPLAY MSG2
        JMP NEXT1
FAILED: DISPLAY MSG3
NEXT1: DISPLAY MSG4
        MOV SI, OFFSET FNAME2
BACK1:  MOV AH, 01H
        INT 21H
        CMP AL, 0DH
        JE NEXT2
        MOV [SI], AL
        INC SI
        JMP BACK1
NEXT2: MOV [SI], BYTE PTR '$'
         LEA DX, FNAME2
         MOV AH, 41H
         INT 21H
         JC DFAIL
         DISPLAY MSG5
         JMP FINAL
DFAIL: DISPLAY MSG6
FINAL: MOV AH, 4CH
         INT 21H
END START
```