NAME – ARKA SINHA

BATCH - CSE A2

SUBJECT - MMC LAB REPORT

2020-2021

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
```

Screen Shot-

END START

```
DOSBox Status Window

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

CopCopuright (C) Hicrosoft Corp 1981-1985, 1987. All rights reserved.

CONDB ject filename (binary.OBJ):

CONDB ject filename (binary
```

	HMC Week-L
	Saath
	Date_/_/ Dimary search
	MODEL SHALL
-	DICELAY INDICATE AND
	DISPLAY MACRO MSG
	LEA DX, MSG
	INT 21H
	ENDH
	- DATA
	LIST DB OIM, DSH, DTH, 10H, 12H, 14H
-	NUMBER EQU (\$-LIST)
	KEY DB 12H
	MSGI DE ODM, DAH, "ELEMENT EOUND IN LIST \$"
	MSG2 DB ODH, DAH, "SEARCH FAILED! NO FOUND \$"
	.CODE
	START: HOU AX, @DATA
	hou Ds, Ax
	HOU CH, NUMBER-1; HIGH VALUE
	MOVCL, DOM ; LOW VALUE
	HUMIN: MOV SI, OFFSET LICT
	XOR AX, AX; clear the An nogieter
	Crip (L, Cr)
	JE NEXT
-	JNC FAILED
	NEXT: MOV AL, CL
-	ADD AL, CH
	SHR AL, OIH ; DIVIDE BY 2
	YOR AH, AH ; CLEAR AH
	MOV BP, AX
-	Page No.

	Date/_/ Gaathi
	hov AL, DS: [BP][ST]
	CHP AL, KEY ; COMPARE KEY AND ALE
	JE SUCCESS JEF EQUAL, DISPLAY SUCCESS MESSAGE
	JC INCLOW
	MOU CH, BL ; IF KEYSALID SHIFT HIGH
	DEC CH
	JMP AGAIN
	INGLOW: HOU CL, BL ; IFKEYCALID SHIFT ZOW
	INCCL
	ZMP ACIAIN
	SUCCESS: DISPLAY MSGI
	JAP FINAL
	FAILED: DISPLAY MSG2
	FINAL: MOV AH, 4CH
-	INT 21H
	EUD START
	Page No.

Source Code-

.MODEL

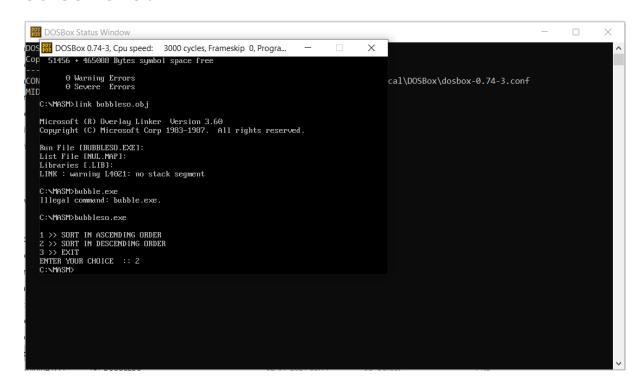
```
SMALL
        DISPLAY MACRO MSG
                LEA DX, MSG
                MOV AH, 09H
                INT 21H
         ENDM
         .DATA
        LIST DB 02H, 01H, 34H, 0F4H, 09H, 05H
        NUMBER EQU $-LIST
        MSG1 DB 0DH, 0AH, "1 >> SORT IN ASCENDING ORDER$"
        MSG2 DB 0DH, 0AH, "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-



(Saathi) JE FINAL ASCORT : HOU BL, OOM AGAIN: HOU SI, OFFSET LIST MOVIL, OOM ; JVALUE hov Bn, Cn SUB BH, BL N-1-1 NPASS: CHP CL, BH JNC NEXT MOU DE EST HOU BP, OIM CHP AL, DS: [BP][SI] JC - NOPE XCHA AL, [SI+1] DCHG(SD, AC - NOPE : INC CL INC SI JMP NAASS NEXT: INC BL CMP BL, CM JC AGAIN JMP FINAL DESCORT: MOUBL, DOM AGAINS, HOU SI, OFFSET LIST FOU CL, OUH ; I VALUE HOUBH, MICH 5 UB BH, BL NPASC1: CMP CL, BH JNC NEXT MOU AL, [SI] HOUBP, OIH CHP AL, DS: [BP][SI]

JNC - HOPE 1 XCHG AL, [SI+1] XCHC (SI), AL NOPE 1: INC CL 100 51 JAP NPASSI NEXT 1: INC BL (np BL, cn FINAL: MOV AM, 4CH INT 21H END START

Page No.

Source Code-

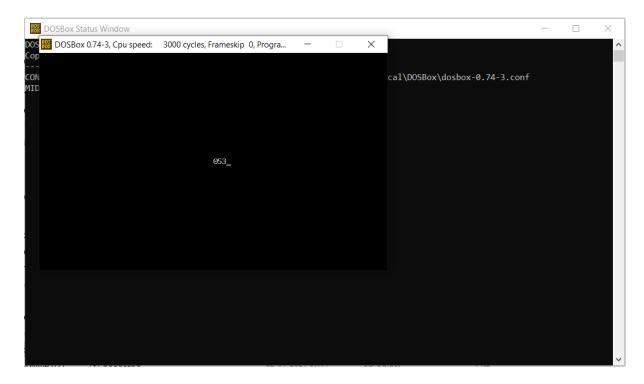
```
; PROGRAM
:: PROGRAM
TO READ AN
ALPHANUMERIC
CHARACTER
AND DISPLAY
ITS
               ; EQUIVALENT ASCII CODE AT THE CENTRE OF THE SCREEN
               .MODEL SMALL
               DISPLAY MACRO MSG
                       LEA DX, MSG
                       MOV AH, 09H
                       INT 21H
               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
                      ; SET CARRY FOR ERROR
ERR : STC
RET
FRET: CLC
RET
CHECK ENDP
; PROCEDURE ENDS HERE
FINAL : MOV AH, 4CH
       INT 21H
```

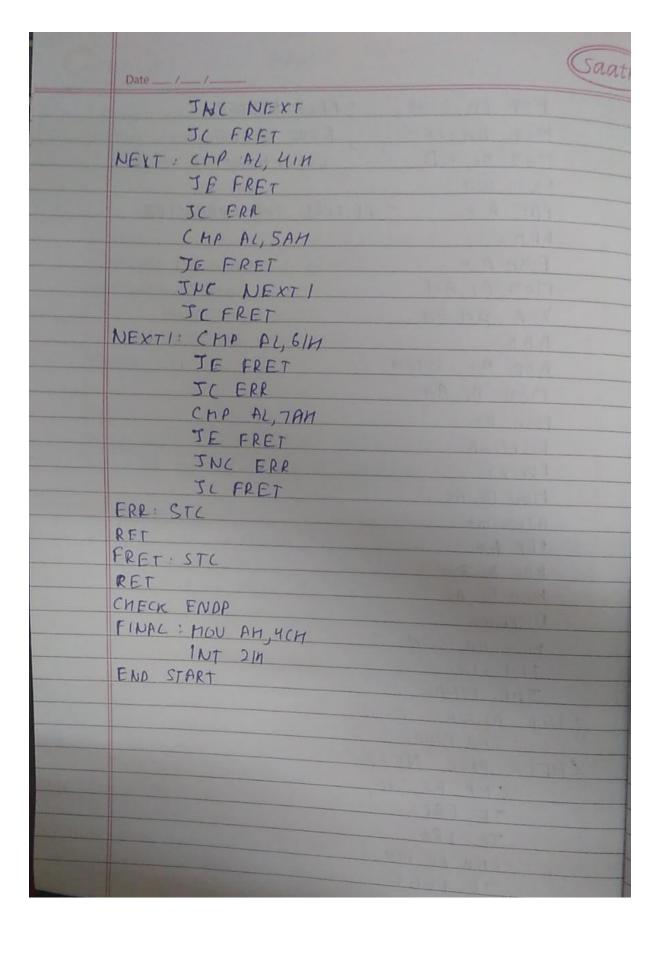
Screen shot-

END START



Date_/_/ Date_/_/ Date_/_/
. MODEL SMALL
DISPLA MACRO MSG
LEA DX, HSG
 MOV AH, O9H
INT 214
 ENDT
 ; MACRO TO DISPLAY A CHARACTER
DISPCHAR MACRO
ho v An, 62n
INT 214
ENDA
. PATA
HSGI DB OPH, OAM, "ENTER AN ALPHANUNERIC CHARACTER: \$"
hs62 DB COH, OAH, "NOT ALPHANUMERIC : 5"
 . CODE
START: MOV Ax, @PATA
MOV DS, AX
 DISPLAY MSGI
MOV AH, GIH
INT 21H
CALL CHECK ; CHECK FOR ALPHANUNERIC CHARACTER
30 ERROR
PUSN AX
; SET MODE AND CLEAR THE SCREEN
, ROW = 25 AND COLUMN = 80
MOV AM, OOH
hov pl, osn
INT IOH
HOU AN ONE TO' THE MID POINT OF THE SCREEN
HOU AN,O)H
Page No.

(Saathi) HOW BH, OOH ; PAGE NUMBER MOV DA,120 , ROW VALUE MOV D1, 40D COLUAN VALUE INT 10H POP A & ; RESTORE THE CHARACTER AAh PUSH AX MOU AL, AH DOR AM, AM AAN ADD Ax, 3030M MOV DZ, AM PUSH AX DISPCHAR POPAX HOU DL. AL DISPCHAR POP AX ADD AL,30H MOU DL, AL DISPEHAR MOV AM, OTH 1NT 214 JAP FINAL EFFOR: DISPLAY HSUIZ THP FINAL CHECK PROC NEAR CMP AL, JOH JE FRET JC ERR CMP AL, 39H JE FRET



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

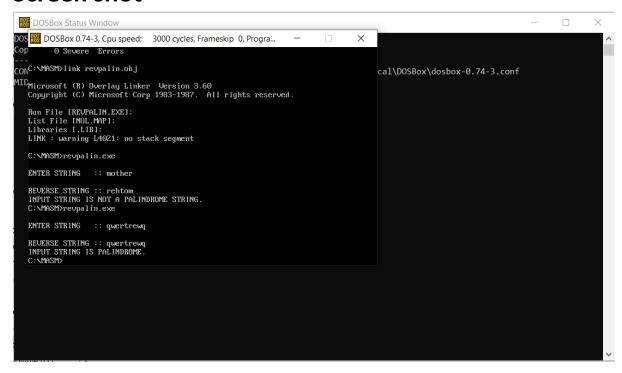
SUCCESS:DISPLAY MSG3

FINAL: MOV AH, 4CH

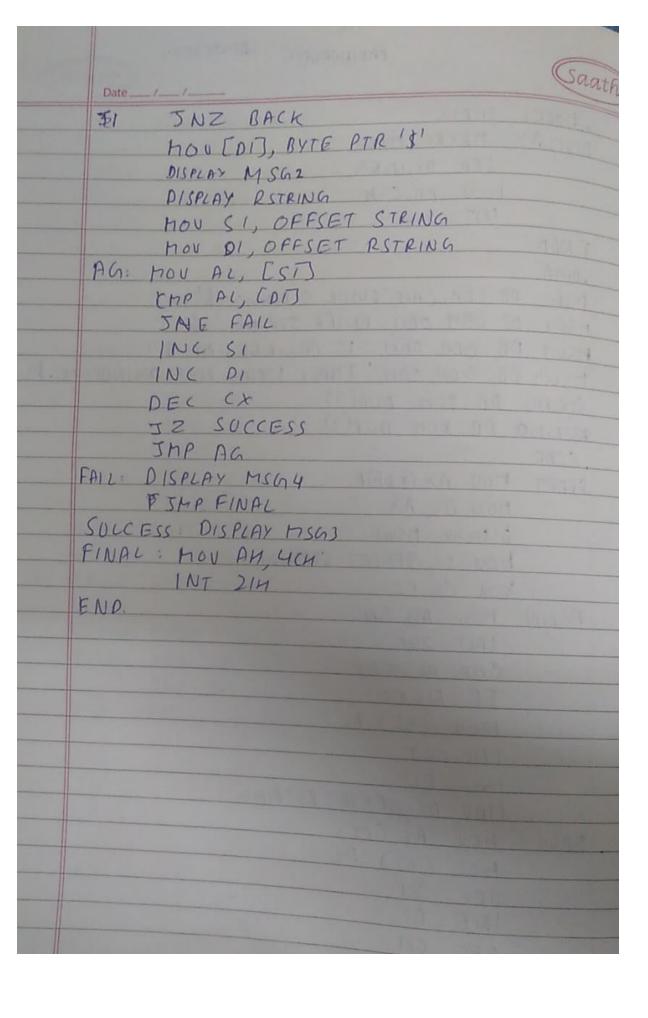
INT 21H

END

Screen shot-



nhc	-
PALINDRONE 18MIRCS024	
Date (Saathi)	
, MODEL SHALL	100
DISPLAY MACRO MSG	-0.
LEA DX, MSG	
HOV AM, 69 H	-
INT 21H	
ENON	-
MCC) DO 12	
MSGI DE ODH, DAH, "FUTER STRING :: \$"	_
MSGO DE ODM, OAH, "ENTER STRING !! \$"	-
MSG3 DB GDM, OAM, "IS PALINDROME \$" MSG4 DB ODM, OAM, "INPUT STRING NOT PALINDROME. S"	-
STRING DB 800 DUP(?)	
RSTRING DB 80h DUP(?)	
CODE	
START: HOU AX, @ DATA	-41
hou DS, Ax	
DISPLAY DSGI	
HOU SI, OFFSET STRING	
XOR CL, CL	-
AGAIN: HOU AH, OIH	-
INT 2IM	-
CHP AL, ODH	
JE NEXT	
MOU (SE), AL	
INC SI	
MOU DI, OFFSET RSTRING	
BACK: HOV AL, (SI)	
HOV COED, AL	
DEC SI	
INC DI	
DEC CM	
Page No.	
	1



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
                    ; COMPARE THE LENGTHS
       CMP CL, BL
                             ; IF LENGTHS ARE EQUAL, PROCESS NEXT STATMENT
       JNE FAIL
       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 074-3, Cpu speed: 3000 cycles, Frameskip 0, Progra... — X

Cop 51698 + 464936 Bytes symbol space free

CON 0 Marning Errors
CON
```

	STRING 18th 1905024 (Saathi)
	, MODEC SHALL
	DISPLAY HACRO HSG
_	LEA DX HSG
_	HOU AH, 09H
	1 NT 2/H
-	ENDH
_	DATA
	MSGI DB ODH, DAH, "ENTER FIRST STRING: \$"
	HISGI DE OPH OAM, "ENTER SECOND STRING: \$"
_	HSG3 DB DBM, OAM, "LENGTH OF FIRST STRING: \$"
	MSG4 DB ODH, OAH, "LENGTH OF SECOND STRING: \$"
	MSGS DB ODM, OAM, "STRINGS EQUAL \$"
	MSG DB ODA, OAM, "STRINGS NOT EQUAL \$"
	ITRINGI DB 80H DUP(?)
	ITRINGI DB 80H DUP(?)
-	. CODE
-	START: MOV AX, @PATA
	hov DS, Ax
	DISPLAY MSGI
	MON SI, OFFSET STRINGS
-	CALL READSTR
	HOV BL, CC
	DISPLAY MSGL
	MOU SI, OFFSET STRING?
	CALL READSTR
	PUSH BX
	PUSH CX
	MOV AL, BL
	CALL LEN_DIS
	POP CX
	POP BX
	Page No.

CMP (L, BL JNE FAIL HOU SI, OFFSET STRING! MOV DI, OFFSET STRINGS CLD CHK: HOU AL, [S] CHP AL, EDIS INE FAIL INC ST INC DI DEC CL JNZ CHK DISPLAY HSGS JHP FINAL LEN-DIS PROC NEAR XOR AH, AH ADD AL, OOH AAH ADD AX, 3030H HOU BH, AL hou PL, An hov An, OZH INT 214 HOU DL, BH MOV AM OZH 1NT 2111 RET LEN_DIS EDENDP READSTR PROC NEAR XOR CL, CL BACK: HOV AN, OIH INT 21H

CHP AL, ODM JE FINISH HOY CSD, AL INC SI INC CL JAP BPCK FINISH HON CSED, BYTE PTR 'S' RET READSTR ENDP FAIL: DISPLAX TISGE FINAL: HOY ANTCH INT 211 END START

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
```

Screen shot-

final: mov ah,4ch
 int 21h
 end

```
DOSBox Status Window

DOS DOSBox 0.74-3, Cpu speed: 3000 cycles, frameskip 0, Progra... — X

COPHICTOR OF Resembler Version 5.00

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

COM 0, ject filename Incr. 0.81.1:

MIL Source listing INUL.ISTI:

Cross-reference INUL.CRF:

51758 + 464786 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:NPASHDlink ncr.obj

Hicrosoft (R) Overlay Linker Version 3.60

Copyright (C) Hicrosoft Corp 1983-1987. All rights reserved.

Run File INUR. PREI:
List File INUL.HRP:
Libraries (LIBI:
Libraries (LIBI:
Libraries (LIBI:
Libraries (LIBI:
C:NPASHDncr.exe
0)

C:NPASHDncr.exe
0)

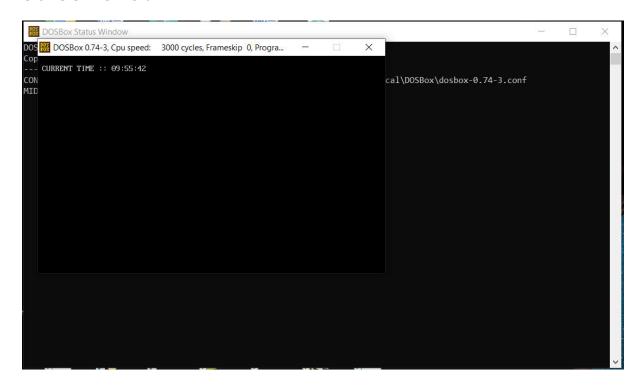
C:NPASHDncr.exe
```

HAC 1Bh 19CS024 DC8 Date___/__/__ model small · data ndw g s du 3 nes du D code mov an, a data me ds, ax mar ann may bx, a cali merpro call disp imp final morpho proc mean comp an, bu je sest emp by, 0 je seol cmp bx,1 je slen dere an comp in, an je inon push an push bn cal monkno pop bn popan der br push an push bn call me merpero

Date /	Caathi
but br	353102 37501
bub an	PARTY PARTY FOR
net ·	THE ASSESSMENT
sel: ime non	THE PARTY
net	70 7188
inco: inc non	E E E E
nean : add mer, an	47.80
ret	TINETIME DE COMP
merpro endp	THE DE LOCATION
	2001.
disp proc near	STATE OF STA
mar dr, men	all the stand
add In, 3030 h	MIN 100 11 15
mov dh 6h	part TAT
mor ah, ozh	MATACON PROP
1 nt 21h	
mor al ll	
mov ah, Ozh	THE REAL PROPERTY.
int 21h	100
net	1 3 21 904
disp ends	1.5 12.7 4.6
linal: mor at, 4ch	12 3/1
final: mar ah, 4ch	1 100 100 400
end	33 371
	17.17 USM
	737
	BERTHRON.
	MA FOLIMA
	17 17 1
	17 3 06 1
	Page No.
	Page No.

```
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
                  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, ØBH
       INT 21H
       CMP AL, 00H
       JE AG
FINAL : MOV AH, 4CH
       INT 21H
END START
```



	TIME IBNIACSO 24	Gaathi
-	Date//	
-	. HODEL SHALL	-
	DISPLAY MACRO MSG	
-	LEA DX, hSG	
	MOU AN, 094 1NT 211	
	ENDH	N. John Co.
7	. DATA	Control of the Contro
	TIMESTER DB OZOH DUP(?)	
	HSGI DB "CURRENT TIME : \$"	
	.CODE	
	START: MOU AX, @ PATA	4.3
	hou Os, Ax	
	MOV AL, O3H	
	1NT 10H	
	MOV SI, OFFSET TIMESTR	
	hov ph, 2ch	1
	INT 21H	A
	hov AL, (h	<u> </u>
	AAH	A
	APP Ax, 3030H	
	MOV CSIJ, AM	
	INC SI	The
	HOV (S.1), BYTE BTR (:)	
	INC SI	
-	MOU AL, CL	
	AAH	
	ADD AX, 3030h	
	hou Csi], AH	The same of the sa
	MOU CSI), AL	
	INC SI	
	MOV AL, DM	
	F10 0 11 C, D21	

THE PARTY OF THE STATE OF THE S (Gaathi) AAH ADD Ax, 3020H hou CSD, An HOV CSIJ, AL INCSI MOU (SI), BYTE PTR (5) 1 DISPLAX MSGI DISPLAY TIMESTR MOU AMOBA INT 21H CMP AL, OU JE AG FINAL MOV AN, 4CM INT 21H END START

Page No.

```
PROGRAM
::
PROGRAM
TO
SIMULATE
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
                   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
       MOV BX, 0FFH
AG:
AG1:
        NOP
        XOR AX, AX
        DEC BX
        JNZ AG1
        DEC CX
        JNZ AG
        POP BX
        POP AX
```

RET

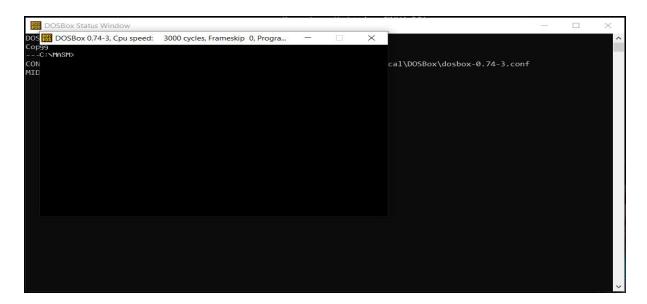
DELAY ENDP

FINAL: MOV AH, 4CH

INT 21H

POP CX

END START

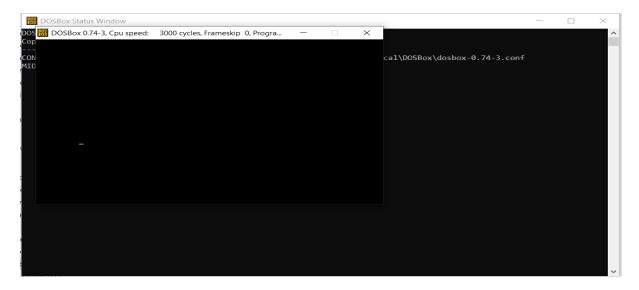


t	the		18HI9CSOZ4	
	PCO UNTE	K	Gaa	
Date / /				
. HODEL SHALL				
CODE	Azala			
START: HOU CL,				
hou Art				
MOU AL				
1 NT 10				
BACK: MOV BM,				
hov Du, o	OH			
hou Dz, o.				
hou DAN,	0211	Black State of		
1 NT 10H	,			
hou AL,		N. P. Control of the		
ADD AL, C	04			
AAH	0.0			
ADD AX				
hov chip				
hov AL,	An			
hou DL,	AL			
MOV AM	027		-	
N1 214				
how AL, C	n			
MOU OL, AL				
NOU AH, 02	Л			
CALL DELF	12			
INC CL	17		-	
DOR AX, AX			-	
Che EL, 10				
JNE BACK				
JE FINAL				
- L INNE				

(Saathi) DELAY PROX NEAR Push ex PUSH AX PUSH BX MOV CX, OFFH ACT : MOU BX, OFFH ACI: PNOP XOR AX, AX DEC BY JNZ AGI DEC CX JNZ AG POP BX POP AX POP CX RET DELAY ENDP FINAL : HON AM, 4CH 1 NT 214 ENP START

```
; 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
```



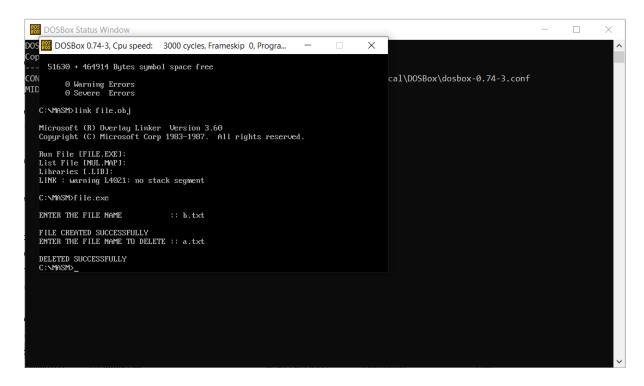
	Date_/_/	024 Saath
	. HODEL SMALL	
	DISPLAY MACRO MSM	4 4 1004
	LEA DX, HSG	all the h
	100 AH, 09H	OA TON
	INT 21H	42 3384
	ENDH	78 00 3 0
	DATA	ISLAM INV
	ROW DB OZH OUP (?)	A TOX
	(OLDEN DB OZM DUP(2)	A STATE OF THE STA
	MSGI DB ODH, OAH, "Enter The n	co-ordinate (Row): \$17
	MSG2 DB ODM, OAM, "Enter The yo	io-ordinate (lolumn): \$"
	CODE	49 939
	START: HOU AX, QDATA	9 618
	hou DS, Ax	13.9.9
	DISPLAY MSGI	639
	hov SI, OFFSET ROW	949554134
	CALL READ	a ser sunt
	DISRAP MSG2	
	MON SIJOFFSET COLUM	N
-	CALL READ	
	HON SI, OFFSET ROW	
	HOU AH, CST)	
	MOV AL, CST	
	SUB AX, 3030H	
	AAD	
	hou DE, AL	
	hov AH, OOH	
	hov AL, 03h	
	INT 10H	
	160 AH,024	
		Page No.

INTE IOH JHP FINAL READ PROC NEAR HOV CX, 02H BACK: HOU AM, OIM INT 21H nov LSIJ, AL INC SI DEC CX JNZ BACK RET REOPREAD ENDP FINAL: HOV AH, OIH INT 21H hov AH, 4CH INT 21H END START Page No.

```
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
```

```
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
```

BACK: MOV AH, 01H



	Date_/_/_ FILE IGHISCS 024 Saathi
	. HODEL SHALL
	DISPLAY HACRO HSG
	LEA DX, HSG
	MOV AH, O9H
	INT 214
	ENDH
-	DATA
-	MSGI DB ODH, OPH, "ENTER FILE NAME: \$"
_	HSG2 DB ODM, OAM, "FILE CREATED SUCCESSFULLY \$"
	MSG3 DB ODM, OAH, "CREATION FALLED \$"
	hand DB ODM DAH, " ENTER FILE NAME TO DELETE: \$"
	MSGS DB ODM, DAM, "DELETION SUICESCEUL \$"
	MSGLOB ODH, OAH, "DELETION FAILEDS"
	FNAME DB YOH DUP(!)
	FNAME 2 DB 40H DUP(?)
-	CODE
	START: HOU AX, @DATA
	hov Ds, Ax
	DISPLAY MSG1
	how SI, OFFSET FNAME
	BACK: NOV AH, OIH
	INT 21H
	enp Al, FODH
	JE WEXT
	hov ESI] AL
	1 NC SI
	Shp BACK
	NEXT: HOV [SI] BYTE PTR '\$'
	Len ux, FNAhe
	hov cx, DOM

(Gaathi)

Date___/__/__ HOU AH, 3CH INT 21H JC FAILED DISPLAY MSG2 JHP NEXTI FAILED DISPLAY MSG3 NEXT 1: DISPLAY MSG4 HON SI, OFFSET FNAMEL BACKI: how AMOIH INT 211 CHP AL, ODM JE NEXTL hou [SI], AL INC SI THP BACK! NEXTZ! HOU [SI], BYTE PTR 'S' LEA DX, FNAME2 HOV AH, 41H

INT WZIH JL DFAIL

DISPLAY MSGS JHP FINAL

DEALL: DISPLAY HSG6 FINAL: how AH, 4Ch NT 21H

END START

Page No.