

Yashaswini Shah
18M19CS216

LAB1: Binary Search

```
.MODEL SMALL
DISPLAY MACRO MSG
    LEA DX, MSG
    MOV AH, 0DH
    INT 21H
ENDM

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

        MOV CH, NUMBER-1
        MOV CL, 00H

AGAIN : MOV SI, OFFSET LIST
        XOR AX, AX
        CMP CL, CH
        JE NEXT
        INC FAILED

NEXT : MOV SI, CL
        ADD AL, CH
        SHR AL, 01H
        MOV BL, AL
        XOR AH, AH
        MOV BP, AX
        MOV AL, DS:[BP][SI]
        CMP AL, KEY
        JE SUCCESS
        JC INGLOW
        MOV CH, BL
        DEC CH
        JMP AGAIN

CODE ENDM
```

14/10/2020
classmate
Date _____
Page _____

Yashaswini Shah
18M19CS216

```
NEXT : MOV SI, CL
        ADD AL, CH
        SHR AL, 01H
        MOV BL, AL
        XOR AH, AH
        MOV BP, AX
        MOV AL, DS:[BP][SI]
        CMP AL, KEY
        JE SUCCESS
        JC INGLOW
        MOV CH, BL
        DEC CH
        JMP AGAIN
```

```
INGLOW : MOV CL, BL
        INC CL
        JMP AGAIN
```

```
SUCCESS : DISPLAY MSG1
        JMP FINAL
```

```
FAILED : DISPLAY MSG2
FINAL : MOV AH, 4CH
        INT 21H
```

END START

classmate
Date _____
Page _____

LAB2: Bubble sort.

```

.MODEL SMALL
DISPLAY MACRO MSG1
    LEA DX, MSG1
    MOV AH, 09H
    INT 21H
ENDM
.DATA
LIST DB 02H, 01H, 34H, 0F4H, 09H, 08H
NUMBER EQU $-LIST
MSG1 DB 0DH, 0AH, "1 => SHORT IN ASCENDING
                ORDER$"
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
        DISPLAY MSG1
        DISPLAY MSG2
        DISPLAY MSG3
        DISPLAY MSG4
        MOV AH, 01H
        INT 21H

```

SUB AL, 30H

CMP AL, 01H

JE ASCSORT

CMP AL, 02H

JE DESSORT

CMP AL, 08H

JE FINAL

DISPLAY MSG 5

JMP FINAL

ASCSORT : MOV BL, 00H

AGAIN : MOV SI, OFFSET LIST

MOV CL, 00H

MOV BH, CH

SUB BH, BL

NPASS : CMP CL, BH

INC NEXT

MOV AL, [SI]

MOV BP, 01H

CMP AL, DS:[BP][SI]

JC -NOPE

XCHG AL, [SI + 1]

XCHG [SI], AL

NOPE : INC CN

INC SI

JMP INPASS

NEXT : INC BL
 CMP BL, CH
 JC AGAIN
 JMP FINAL

DESSORT : MOV BN, OOH

AGAINL : MOV SI, OFFSET LIST
 MOV CL, OOH
 MOV BH, CH
 SUB BH, BL

NPASSL : CMP CL, BH
 JNC NEXT
 MOV AL, [SI]
 MOV BP, OTH
 CMP AL, D8 : [BP][SI]
 JNC NOPG1
 XCHG1 AL, [SI + L]
 XCHG1 [SI], AL

NOPEL : INC CL
 INC SI
 JMP NPASSL

NEXTL : INC BL
 CMP BL, CH
 JC AGAINL

FINAL : MOV AH, 4CH
 INT 21H

END START.

Lab 3: DISPLAYING EQUIVALENT ASCII CODE

```

.model small
.data
msg1 db 0dh, 0ah, "enter alphanumeric
character $"
res db 02 dup(0)
.code
mov ax, @data
mov ds, ax
lea dx, msg1
call disp
mov ah, 01h
int 21h
mov bl, al
mov cl, 4
shr al, cl
cmp al, 0ah
jc digit
ADD AL, 07H

digit :
add al, 30h
mov res, al
add bl, 0fh
cmp bl, 0ah
jc digit1
add bl, 07h

```

```

digit1: add bl, 30h
        mov res+1, bl

```

```

        mov ah, 00h
        mov al, 03h
        int 10h

```

```

        mov ah, 02h
        mov bh, 00h
        mov dl, 28h
        int 10h

```

```

        mov res+2, '$'
        lea dx, res
        call disp
        mov ah, 4ch
        int 21h

```

```

disp proc near
        mov ah, 0ah
        int 21h
        ret
disp endp
end

```

LAB4: Palindrome Program

.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
PALLINDROME STRING.\$"

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, 0DH

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 STRING & 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

LAB 10: Read two strings, store them in locations STR1 and STR2. Check whether they are equal or not and display the length too.

JMP AX

FAIL : DISPLAY MSG1

JMP FINAL

SUCCESS : DISPLAY MSG3

FINAL : MOV AH, 4CH

INT 21H

END.

.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 SECOND STRING:\$"

MSG5 DB 0DH, 0AH, "-- STRINGS ARE EQUAL-\$"

MSG4 DB 0DH, 0AH, "LENGTH OF FIRST STRING:\$"

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

DISPLAY MSG2

MOV SI, OFFSET STRING2

CALL READSTR

PUSH BX

classmate
Date _____
Page _____

```
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
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
```

classmate
Date _____
Page _____

```
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
```

Lab 6: Develop an assembly language program to
compute nCr using recursive
procedure. Assume that 'n' and
'r' are non-negative integers.

classmate

Date _____
Page _____

.MODEL

.DATA

N DW 4
R DW 2
NCR DW 0

.CODE

```
MOV AX, @DATA
MOV DS, AX
MOV AX, N
MOV BX, R
CALL NCRPRO
JMP FINAL

NCRPRO PROC NEAR
CMP AX, BX
JE RES1
CMP BX, 0
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 NCRPRO
POP BX
POP AX
RET

RES1: INC NCR
RET

NCR : INC NEAR
RESN: ADD NCR, AX
RET

NCRPRO ENDP

DISP PROC NEAR
MOV BX, NCR
ADD BX, 30H
MOV DL, BH
MOV AH, 02H
INT 21H
MOV DL, BL
MOV AH, 02H
INT 21H
RET
DISP ENDP

classmate
Date _____
Page _____

Lab 7: Read the current line from the system
and display it in standard format
on the screen.

FINAL: mov AH, 0CH
INT 21H
END

MODEL SMALL

CODE

mov AH, 02H

INT 21H

mov AL, CH

AAM

mov BX, AX

CALL DISP

MOV DL, ":",

MOV AH, 02H

INT 21H

MOV AL, CL

AAM

MOV BX, AX

CALL DISP

MOV DL, ":",

MOV AH, 02H

INT 21H

MOV AL, DH

AAM

MOV BX, AX

CALL DISP

mov AH, 4CH

INT 21H

DISP PROC NEAR

MOV DL, BH

ADD DL, 30H

MOV AH, 02H

INT 21H

MOV DL, BL

ADD DL, 30H

MOV AH, 02H

INT 21H

RET

DISP ENDP

END

Lab

Q: Write a program to simulate a
Decimal Up-counter to display
00-99.

.MODEL ~~SMALL~~ SMALL

.CODE

MOV CL, 00

MOV AH, 00H

MOV AL, 08H

INT 10H

BACK : MOV BH, 00H

MOV DH, 00H

MOV DL, 00H

MOV AH, 02H

TNT 10H

MOV AL, CL

ADD AL, 00H

AAM

ADD AX, 0030H

MOV CH, AL

MOV DL, AH

MOV AH, 02H

INT 21H

MOV DL, CH

MOV AH, 02H

INT 21H

CALL DELAY

INC CL

BOOR AX, AX

CMP CL, 100D

JNE BACK

JG LAST

DELAY PROC NEAR

PUSH AX

PUSH BX

PUSH CX

MOV CX, 00FFH

AG : MOV BX, 0FFFH

AG1 : NOP

DEC BX

JN2 AG1

DEC CX

JNZ AG

POP CX

POP BX

POP AX

RET

DELAY&NOP

LAST : MOV AH, 4CH

INT 21H

END

CLASSMATE

Date _____

Page _____

Lab 9

: Read a pair of input co-ordinates in BCD
and move the cursor to the specified
location on the screen.

CLASSMATE

Date _____

Page _____

MODEL SMALL

DISP MACRO msg

LEA DX, msg

MOV AH, 09H

INT 21H

ENOM

.DATA

ROW DB 02 DUP(0)

COL DB 02 DUP(0)

MSG1 DB 0DH, 0AH, "Enter X-co-ordinate:\$"

MSG2 DB 0DH, 0AH, "Enter Y-co-ordinate:\$"

MSG3 db 0dh, 0ah, "cursor displayed at
one correct coordinates \$"

.code.

MOV AX, @DATA

MOV DS, AX

DISP MSG1

MOV SI, OFFSET ROW

CALL READ

DISP MSG2

MOV SI, OFFSET COL

CALL READ

MOV SI, OFFSET ROW

MOV AH, [SI]

INC ST

MOV AL, [SI]

SUB AX, 3030H

AAD

MOV DH, AL ; row
MOV SI, OFFSET .COL
MOV AH, [SI]
INC SI
MOV AL, [SI]
SUB AX, 3030H
AAD
MOV DL, AL

MOV AH, 00
MOV AL, 0BH
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

JN2 BACK

RET

READ ENDP

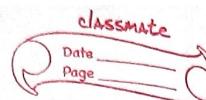
CLASSMATE

Date _____

Page _____

FINAL : MOV AH, 01H
INT 21H
MOV AH, 4CH
INT 21H
END.

Lab 10: Write a program to create an input file and to delete an existing file.



```
.MODEL SMALL
DISP MACRO MSG
  LEA DX,MSG
  MOV AH,0AH
  INT 21H
ENDM
```

```
.DATA
MSG1 DB 0DH,0AH,"ENTER THE FILE NAME FOR
CREATION : $"
MSG2 DB 0DH,0AH,"FILE CREATED SUCCESSFULLY"
MSG3 DB 0DH,0AH,"CREATION FAILED$"
MSG4 DB 0DH,0AH,"ENTER THE FILE NAME
FOR DELETION :-$"
MSG5 DB 0DH,0AH,"FILE DELETED SUCCESS-
FULLY$"
MSG6 DB 0DH,0AH,"DELETION FAILED$"
FNAME1 DB 10 DUP(0)
FNAME2 DB 10 DUP(0)
```

```
.CODE
MOV AX,@DATA
MOV DS,AX
DISP MSG1
MOV SI,00
BACK1: MOV AH,0DH
INT 21H
```

classmate
Date _____
Page _____

```
CMP AL,0DH
JE NEXT1
MOV FNAME1[SI],AL
INC SI
JMP BACK1
NEXT1: MOV FNAME1[SI],255
LEA DX,FNAME1
MOV CX,00
MOV AH,8DH
INT 21H
JC CFAIL
DISP MSG2
JMP DEL
CFAIL: DISP MSG3
DEL: DISP MSG4
MOV SI,00
BACK2: MOV AH,0DH
INT 21H
CMP AL,0DH
JE NEXT2
MOV FNAME2[SI],AL
INC SI
JMP BACK2
NEXT2: MOV FNAME2[SI],255
LEA DX,FNAME2
MOV AH,4DH
INT 21H
JC DFAIL
DISP MSG5
```

JMP LAST

DFAIL : DISP MSG6

LAST : MOV AH, 4CH

INT 21H

END