

LAB PROGRAM 1 Binary Search

classmate
Date _____
Page _____

Lab Program

• 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 012H

MSG1 DB 0DH, 0AH, ^{cc} ELEMENT FOUND IN THE
LIST ... \$⁹⁹

MSG2 DB 0DH, 0AH, ^{cc} Search failed !! Element
not found in the list \$⁹⁹

• 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 AX, AX ; Clear AX

```

MOV BP, AX
MOV AL, DS: [BP][SI]
CMP AL, KEY ; Compare key & A[SI]
JE SUCCESS ; If equal, display success msg
JC INCLW ; If KEY > A[SI] shift high
MOV CH, BL ; If KEY < A[SI] shift low
DEC CH
JMP AGAIN

INCLW : MOV CL, BL
INC CL
JMP AGAIN

SUCCESS : DISPLAY MSG1
JMP FINAL

FAILED : DISPLAY MSG2 ; Job over, terminate
FINAL : MOV AH, 4CH
INT 21H

END START

```

Bubble Sort.

classmate

Date

Page

.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 ASC \$"

MSG2 DB 0DH, 0AH, "2>> SORT IN DES \$"

MSG3 DB 0DH, 0AH, "3>> EXIT \$"

MSG4 DB 0DH, 0AH, "4>> ENTER A CHOICE \$"

MSG5 DB 0DH, 0AH, "5>> INVALID \$"

.CODE

START: MOV AX, @DATA

MOV DS, AX

LEA SI, LIST

MOV CX, NUMBER-1

DISPLAY MSG1

DISPLAY MSG2

DISPLAY MSG3

DISPLAY MSG4

MOV AH, 01H

INT 21H

SUB AL, 30H

CMP AL, 01H

JE ASCORT

CMP AL, 02H

JE DESORT

CMP AL, 03H

JE FINAL

DISPLAY MSG5

JMP FINAL

ASCORT: MOV BL, 0DH

AGAIN : mov SI, OFFSET LIST

mov CL, DDH

mov BX, CH

SUB BX, BL

NPASS : CMP CL, BX

JNC NEXT

mov AL, [SI]

mov BP, DIH

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, DDH

AGAIN1 : mov SI, OFFSET LIST

mov CL, DDH

mov BX, CH

SUB BX, BL

NPASS1 : CMP CL, BX

JNC NEXT

mov AL, [SI]

mov BP, DIH

CMP AL, DS: [BP][SI]

JNC _NOPE1

XCHG AL, [SI+1]

XCHG [SI], AL

_NOPE1 :

INC SI

JMP NPASS1

NEXT1: INC BL

CMP BL, CL

JC AGAIN1

FINAL: MOV AX, CL

INT 21H

END START

// ASCII

SURYA Gold

Date

Page

.MODEL small

.data

msg1 db 0ah, 0ah, "Enter alphanumeric char:"

res db 02 dup(0)

.code

mov ax, @data

mov ds, ax

lea dx, msg1

call disp

mov ah, 01h

int 21h

mov al, 0ah

mov cl, 4

shr al, cl

cmp al, 0ah

jc digit

add al, 07h

digit: add al, 30h

mov rcx, al

and bl, 0Fh

cmp bl, 0ah

jc digit1

add bl, 07h

digit1: add bl, 30h

mov rcx+1, bl

mov ah, 00h

mov al, 03h ; TEXT mode

int 10h

mov ah, 02h ; Set the cursor pos

mov bh, 00h ; Page no

mov dh, 0ch ; Row coord is TOP

mov dl, 28h ; Column val

int 10h

mov rax, 2, 0x0

lea dx, res

call disp

mov ah, 4ch

int 21h

disp proc near

mov ah, 09h

int 21h

ret

disp endp

end

// PALINDROME

SURYA Gold

Date

Page

.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, "if string is palindrome \$"
MSG4 DB 0DH, 0AH, "if 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 keyboard ch by ch

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 & store in RString

Mov DI, OFFSET RSTRING
 BACK: MOV AL, [SI] ; AMSE
 MOV [DI], AL
 DEC SI
 INC DI
 DEC CX
 JNZ BACK
 MOV [DI], BYTE PTR '6'
 DISPLAY MSG2
 DISPLAY RSTRING
 MOV SI, OFFSET STRING
 MOV DI, OFFSET RSTRING
 AG: MOV AL, [SI] ; AMSE madam
 MOV AL, [DI] ; ESSMA madam
 JNE FAIL
 INC SI
 INC DI
 DEC CX
 JZ ^CSUCCESS
 JMP AG
 FAIL: DISPLAY MSG4
 JMP FINAL
 SUCCESS: DISPLAY MSG3
 FINAL: MOV AX, 4CH
 INT 21H
 END

String Compare :

• mode small

```
display macro msg
    lea dx, msg
    mov ah, 09h
    int 21h
```

ENDM

• data

```
msg1 db 0dh, 0ah "Enter 1st String $"
msg2 db 0dh, 0ah "Enter 2nd String $"
msg3 db 0dh, 0ah "Length of 1 String $"
msg4 db 0dh, 0ah "Length of 2 String $"
msg5 db 0dh, 0ah "Are Equal ..."
msg6 db 0dh, 0ah "Are unequal $"
```

```
String1 db 80h dup(?)
String2 db 80h dup(?)
```

• code

```
START: mov ax, @data
        mov ds, ax
        display msg 1
        mov SI, offset String1
        call readstr
        mov dx, 1
        display msg 2
        mov SI, offset String2
        call readstr
        push bx
        push cx
        display msg 3
```

```

mov AL, CL
call len-dis
pop CX
pop BX
cmp CL, BL
JNE FAIL
mov SI, OFFSET STRING1
mov DI, OFFSET STRING2

```

```

    cmp
chk: mov AL, [SI]
      mov AL, [DI]
      JNE FAIL
      INC SI
      INC DI
      dec CL
      JNZ CHK
display msg 5
JMP FINAL

```

```

len-dis proc near
    XOR AX, AX
    ADD AL, 00H
    AAM
    ADD AX, 3030H
    mov BH, AL
    mov AH, 02H
    INT 21H
    Ret

```

```

len-dis endp
readstr proc near
    XOR CL, CL
    Back: mov AL, 01H
           INT 21H

```



```
Loop: AL, 0DH
      JE  FINISH
      mov [SI], AL
      INC SI
      INC CX
      JMP BACK
FINISH: mov [SI], Byte ptr $
      RET
Readstr endp
fail:  display msgG
final : mov AX, 4Ch
       int 21h

END START.
```

NCR

model small

.data

n dw 4

r dw 2

ncr dw 0

.code

mov ax, @data

mov ds, ax

mov ax, n

mov dx, 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 incs

push ax

push bx

call ncrpro

pop bx

pop ax

dec bx

push ax

push bx

call ncrpro

pop bx

pop ax

ret

real: inc ncr

ret

incr: inc ncr

resn: add ncr, ax

ret

ncrpro endh

disp proc near

mov bx, ncr

add bx, 3030h

mov dx, bx

mov cx, 02h

int 21h

mov dx, dx

mov ah, 02h

int 21h

ret

disp endp

Final: mov ah, 4ch

int 21h

TIME

```
.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

        MOV AH, 00H
        MOV AL, 03H
        INT 10H

AG: MOV BH, 0AH
    MOV BH, 01H
    MOV DL, 01H
    MOV AH, 02H
    INT 10H

    MOV SI, offset TIMESTR
    MOV AH, 2CH
    INT 21H

    MOV AL, CH
    AAM
    ADD AX, 3080H
    MOV [SI], AH
    INC SI
```

MOV [SI], AL

INC SI

MOV [SI], BYTE PTR '\$'

INC SI

MOV AL, DH

ADD

ADD AX, 3030H

MOV [SI], AH

INC SI

MOV [SI], AL

INC SI

MOV [SI], BYTE PTR '\$'

DISPLAY MSG1

DISPLAY TIMESTR

MOV AH, 0DH

INT 21H

CMP AL, 0DH

JE A6

FINAL: MOV AH, 4CH

INT 21H

END.

Decimal up Counter:

• model small

• code

mov AX, 2CH

int 21H

mov AL, CH

AAM

mov BX, AX

CALL DISP

mov DL, 5

mov AH, 02H

int 21H

mov AL, CH

AAM

mov AL, 0H

AAM

mov BX, AX

CALL DISP

mov AX, 4CH

int 21H

disp proc near

add al, 20H

mov dl, dl

add dl, 30h

mov ah, 02h

int 21h

ret disp endp end

X-Y - Coordinate Cursor

• model small

DISP MACRO MSG

LEA DX, MSG

MOV AH, 09H

INT 21H

END M

• Data

ROW DB 02 DUP (0)

COL DB 02 DUP (0)

MSG 1 DB 0DH, 0AH, "Enter X - coordinate" \$

MSG 2 DB 0AH, 0DH, "Enter Y - coordinate" \$

MSG 3 DB 0DH, 0AH, "Cursor displayed at \$" \$

• Code

MOV AX, @Data

MOV DS, AX

DISP MSG1

MOV SI, OFFSET ROW

CALL READ

DISP MSG 2

MOV SI, OFFSET COL

CALL READ

MOV SI, OFFSET ROW

MOV AH, [SI]

INC SI

MOV AL, [SI]

SUB AX, 3030H

AAD

```

MOV DX, 0
MOV SI, OFFSET A
MOV AX, [SI]
PUSH SI
MOV AL, [SI]
SUB AX, 3030H
AND
MOV DX, AL
MOV AH, 0
MOV AL, SI
INT 10H
MOV AX, 0
INT 10H
JMP FINAL
LEAD PROC NEAR
MOV CX, 0
BACK: MOV AX, 0
INT 20H
MOV [SI], AL
INC SI
DEC CX
JNZ BACK
RET
READ END

```

```

FINAL : MOV AX, 0
INT 20H
MOV AX, 40H
INT 20H
END.

```

FILE CREATE

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

.Data

MSG1 DB 0DH, 0AH, "Enter file name for creation:?"
MSG2 DB 0DH, 0AH, "File Created Successfully?"
MSG3 DB 0DH, 0AH, "Creation failed?"
MSG4 DB 0DH, 0AH, "Enter file name for deletion" "File deleted Successfully?"
MSG5 DB 09, 0DH, 0AH, "↓"
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, 000
BACK1: MOV AX, 01H
 INT 21H
 CMP AL, 0DH
 JE NEXT1
 MOV FNAME1[SI], AL
 INC SI
 JMP BACK1
NEXT1: MOV FNAME1[SI], "?"
 LEA DX, FNAME1
 MOV CX, 00


```
MOV AH, 3CH
INT 21H
JC CFail
DISP MSG2
JMP DEL
CFail: DISP MSG3
```

```
DEL: DISP MSG4
MOV SI, 00
Back2: MOV AH, 01H
INT 21H
CMP AL, 0DH
JE NEXT2
MOV FNAME2[SI], AL
INC SI
JMP BACK2
```

```
NEXT2: MOV FNAME2[SI], 0
LEA DX, FNAME2

MOV AH, 41H
INT 21H
CMP AL, 0DH
JE NEXT2
MOV FNAME2[SI], AL
INC SI
JMP BACK2
```

```
NEXT2: MOV FNAME2[SI], 0
LEA DX, FNAME2
MOV AH, 41H
INT 21H
```

JG DFALL

DISP MEGS

JMP LAST

DFALL : DISP MEG L

LAST : MOV AH, 4CH

INT 21H

END

