

MP Lab - 2

Multiplication operations

- model small

- data

a db 10h

b db 15h

c db ?

- code

```
mov ax, @data
```

```
mov ds, ax
```

```
mov al, b
```

~~```
mov bl, b
```~~

```
mul b
```

```
mov c, ax
```

```
mov ah, 4ch
```

```
int 21h
```

```
end.
```

in      { }

## division operation

.model small

. data

a dw 1100h

z dw 2100h

b dw 15 h

quo db ?

rem db ?

. code

mov ax, @data

mov ds, ax

mov ~~ax~~, a

mov dx, z

mov bx, b

div bx

mov quo, al

mov rem, ah

mov ah, 4ch

int 21h

end

MP LABBinary Search

## .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, "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 ; cl < ch

JE Next

JNC failed

Next : MOV AL, CL ; mid = low + high / 2  $0+5/2=2$

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 MSG 1

JMP FINAL

FAILED : DISPLAY MSG 2 ; JOB OVER. TERMINATE

FINAL : MOV AH, 4CH

INT 21H

## MP-LAB

### Bubble Sort.

.model small

Display Macro Msg

Lea dx, msg

mov ah, 09h

int 21h

endm

.DATA

N db 5

List db 02H, 01H, 34H, 0F4H, 09H, 05H

msg 1 db 0DH, 0AH, "1 >> Ascending order"

msg 2 DB 0DH, 0AH

"2 >> Descending order"

msg 3 DB 0DH, 0AH, "3 : Exit"

msg 4 DB 0DH, 0AH; "Enter your choice":\$"

.code

Mov AX, @Data

Mov DS, AX

Mov CL, N

DEC CL

Display msg 1

display msg 2

display msg 3

display msg 4

f. ~~loop~~

Mov AH, 30H

Cmp AL, 01H

INT 21H

SUB AL, 30H

CMP AL, 01H

JE AS

CMP AL, 02H

JE DES

CMP AL, 03H

JE FINAL

AS:

OUTLOOP1: MOV CH, CL

Mov SI, 00H

INLOOP1: MOV AL, LIST [SI]

INC SI

CMP AL, LIST [SI]

JC NO-EXCHG1

XCHG AL, LIST [SI]

Mov LIST [SI-1], AL

NO-EXCHG1: DEC CH

JNZ INLOOP1

DEC CL

JNZ OUTLOOP1

Mov AH, 4CH

INT 21H

E@

DES:

OUTLOOP2: MOV CH, CL

Mov SI, 00H

IN LOOP2: MOV AL, LIST [SI]

INC SI

CMP AL, LIST [SI]

JNC NO-EXCHG2

XCHG AL, LIST [SI]

MOV LIST [SI-1], AL

NO-EXCHG2: DEC CH

JNZ IN LOOP2

DEC CL

JNZ OUT LOOP2

MOV AH, 4CH

INT 21H

FINAL: MOV AH, 4CH

INT 21H

END

27/10/20

MP-LABAscii Value of String :-

• model small

• data

msg1 db 0dh, 0ah, "Enter alphanumeric character \$"

res db .02 dup(0)

• code

mov ax, @data

mov 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

and bl, 0fh

cmp bl, 0ah

jc digit I

add bl, 07h

digit 1: addl bl, 30h  
mov res + 1, bl

mov ah, 00h

mov al, 03h

int 10h

mov ah, 02h

mov bh, 00h

mov cl, 0ch

mov dl, 28h

int 10h

mov res + 2, '\$'

lea dx, res

call disp

mov ah, 0eh

int 21h

disp proc near

mov ah, 09h

int 21h

ret

disp endp

end

27/10/20 MP-LAB, Palindromic Seq.

.MODEL SMALL

DISPLAY MACRO: MSG

~~LEA DX, MSG~~

MOV AH, 09H

INT 21H

ENDM

.DATA

MSG 1 DB 0DH, 0AH, "Enter String:\$"

MSG 2 DB 0DH, 0AH, "Reverse String:\$"

MSG 3 DB 0DH, 0AH, "Input String is  
palindrome . \$"

MSG 4 DB 0DH, "Input String is not  
palindrome String . \$"

String DB . 80H ; dup (?)

String DB 80H , dup (?)

. Code

Start: mov ax, @data

mov ds, ax

display msg 1

mov si, offset string

xor cl, cl

again: mov ah, 01h  
int 21h  
cmp al, 0dh  
je next  
mov [si], al  
jmp again

next: mov [si], byte ptr '\$'  
dec si  
mov -ch, cl  
mov di, offset string

back      mov al, [si]  
              mov [di], al  
              dec si  
              inc di  
              dec ch  
              jnz back  
              mov [di], byte ptr '\$'  
              display meg 2  
              display string  
              mov -si, offset string  
              mov -di, offset string

ag: mov al, [si]  
cmp al, [di]  
jne fail  
inc si  
inc di  
dec cx  
jz success  
jmp ag

fail: display msg 4  
jmp final

Success: display msg 3

final: mov ah, 4ch  
int 21h

end

Date: / /

MP-LABString Comparison

03/11/20

. MODEL SMALL

Display Macro Msg

```

 LEA DX, MSG1
 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, "Strings are equal $"
MSG5 DB 0DH, 0AH, "Length of second
 string : $"
MSG6 DB 0DH, 0AH, "Strings are not
 equal $"

```

String 1 DB 80H DUP(?)

String 2 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 LENDIS  
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 MSG 5

TMP 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

RET

LEN-DIS ENDP

READSTR PROC NEAR

XOR CL, CL

MOV AH, 01H

INT 21H

CMP AL, 0DH

JE FINISH

MOV [SI], AL

INC SI

INC CL

INC BACK

FINISH : MOV [SI], BYTE PTR '\$'  
RET

'READ STR ENDP

FAIL : DISPLAY MSG 6

FINAL : MOV AH, 4CH

INT 21H

END START

## MP Lab

### System time :-

• model small

• code

mov ah, 2ch

int 21h

mov al, ch

aam

mov bx, ax

call disp

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

MP LAB

10/11/2020

CURSOR

.model small

display macro msg

lea dx, msg

mov ah, 09h

int 21h

endm

.data

row db 02 dup(0)

col db 02 dup(0)

msg 1 db .0dh, 0ah, "Enter the x-  
co-ordinate: \$"

msg 2 db .0dh, 0ah, "Enter the y-coordina-  
co-ordinate: \$"

.code

mov ax, @data

mov ds, ax

display msg 1

mov si, offset col

call read

mov si offset row

mov ah, [si]

inc si

mov al, [si]

sub ax, 3030h

add

mov si, offset col

mov ah, [si]

inc si

mov al, [si]

sub ax, 3030h

add

mov dh, al

mov si, offset col

mov ah, 00

mov al, 03h

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

## MP-LAB

### File programs:-

• model small

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

• data

msg 1 db 0dh, 0ah, "Enter the filename  
for creation: \$"

msg 2 db 0dh, 0ah, "File created successfully  
: \$"

msg 3 db 0dh, 0ah, "File Creation  
Uncesse full: \$"

msg 4 db 0dh, 0ah, "Enter the file name  
for deletion : \$"

msg 5 db 0dh, 0ah, "File Deleted Successfully  
: \$"

msg 6 db 0dh, 0ah, "File deletion  
failed: \$"

fname1 db 10 dup ('')

fname2 db 10 dup ('')

• code

mov ax, @data

mov ds, ax

display msg 1

mov si, 00

back1: mov ah, 01h

int 21h  
cmp al, 0dh  
je next 1  
mov f name 1 [si], al  
inc si  
jmp back 1

next 1: mov f name 1 [si]; '\$'  
lea dx f name 1  
mov cx, 00  
mov ah, 3 ch  
int 21h  
jc c fail  
display msg 2  
jmp del

c fail: display msg 3  
del: display msg 4  
mov si, 00

back 2: mov ah, 01h  
int 21h  
cmp al, 0dh  
je next 2  
mov f name 2 [si], al  
inc si  
jmp back 2

next 2: mov f name 2 [si]; '\$'  
lea dx, f name 2  
mov ah, 4ch  
int 21h

fc dj cd fail  
display msg 5  
jmp final

dfail'. display msg 6

final: mov ah, 4ch  
int 21h

end.