

binsea - Notepad

File Edit Format View Help

```
.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 or cl>ch or cl=ch 0-5
        JE NEXT ; cf=1 cf=0 zf=1 cf=1
        JNC FAILED

NEXT:  MOV AL, CL ; mid=low+high
        ADD AL, CH
        SHR AL, 01H ; DIVIDE BY 2 0100 0010
        MOV BL, AL

Ln 1, Col 1 100% Windows (CRLF) UTF-8
Search            11:57 ENG 01-12-2020 
```

binsea - Notepad

File Edit Format View Help

MOV CL, 00H ; LOW VALUE...

AGAIN : MOV SI, OFFSET LIST

XOR AX, AX

CMP CL, CH ; cl<ch or cl>ch or cl=ch 0-5

JE NEXT ; cf=1 cf=0 zf=1 cf=1

JNC FAILED

NEXT: MOV AL, CL ; mid=low+high

ADD AL, CH

SHR AL, 01H ; DIVIDE BY 2 0100 0010

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



Search



Ln 1, Col 1

100%

Windows (CRLF)

UTF-8

11:57

01-12-2020

BINARY SEARCH.

• model small

; MACRO TO ~~DISPLAY~~ DISPLAY THE MESSAGE...

Display Macro msg

lea dx, MSG

mov ah, 09h

int 21h

end m

• data

list db 0h, 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

JNE 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, PS:[BP][SI]

CMP AL, KEY ; COMPARE KEY AND AL[I]

JE SUCCESS ; IF EQUAL, DISPLAY SUCCESS MESSAGE

JL INCLOW

MOV CH, BL ; If KEY > A [J] SHIFT HIGH

DEC CH

JMP AGAIN

INCLOW: MOV CL, BL ; if key < A [J] 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

END START.

```
0 Warning Errors  
3 Severe Errors
```

```
C:\MP_LAB\MASM>masm binsea;;  
Microsoft (R) Macro Assembler Version 5.00  
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.
```

```
51468 + 465076 Bytes symbol space free
```

```
0 Warning Errors  
0 Severe Errors
```

```
C:\MP_LAB\MASM>link binsea;;  
Microsoft (R) Overlay Linker Version 3.60  
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.
```

```
LINK : warning L4021: no stack segment
```

```
C:\MP_LAB\MASM>cv binsea
```

```
ELEMENT FOUND IN THE LIST ...  
C:\MP_LAB\MASM>_
```

*bubble_sort - Notepad

File Edit Format View Help

Bubble sort

.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

Ln 1, Col 1 100% Windows (CRLF) UTF-8

Search ○ ⏺ 🌐 📁 🛍️ L 📩 🔍 🚧 🖥️ 🎯 🌐 📁 🛍️ 🌐 📁 🛍️

11:36 ENG 02-12-2020

*bubble_sort - Notepad

File Edit Format View Help

JE DESSORT

CMP AL, 03H ; INPUT=3? EXIT

JE FINAL

DISPLAY MSG5

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

Ln 1, Col 1

100%

Windows (CRLF)

UTF-8



Search



11:36
ENG
02-12-2020

*bubble_sort - Notepad

File Edit Format View Help

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



Bubble sort.

model small

data

n db 5

a db 05, 07, 04, 03, 06

code

MOV ax, @data

MOV DS, ax

MOV CL, n

DEC CL

outloop : MOV CH, CL

MOV SI, 00H

inloop : MOV AL, a[SI]

INC SI

CMPL AL, a[SI]

JNC noexch

SCCHG AL, a[SI]

MOV a[SI - 1], AL

noexch : DEC CH

INC inloop

DEC CL

JNC outloop

Mov ah, 4ch

INT 21h

end.

File **Edit** **View** **Registers** **Stack** **Watch** **Options** **Call** **Help**

Locals
 0520:002B BH4C ADD AL, [BX] **Byte Ptr [BX],AL**
 0520:002D CD21 INT 21
 0520:002F 0035 ADD BX, AX
 0520:0031 050201 ADD AX, 0102
 0520:0034 03062000 ADD AX, Word Ptr 103201
 0520:0038 D94202 CALL 0202
 0520:003B C706E6012000 MOU Word Ptr 101E61,0020
 0520:0041 06 PUSH ES
 0520:0042 8B06E401 MOU ES,Word Ptr 101E41
 0520:0046 26BB162000 MOU DX,Word Ptr ES(100201)
 0520:004B D92702 CALL 0202
 0520:004E 8F7902 MOU BH,02
 0520:0050 B9FF7F MOU CX,7FFF
 0520:0053 802000 MOU AX,0020
 0520:0056 D97402 CALL 0202
 0520:0059 07 POP ES

Registers
 AX = 0000
 BX = 0000
 CX = 0000
 DX = 0000
 SF = 0000
 PF = 0000
 AF = 0001
 DF = 0000
 FS = 0520
 GS = 051A
 SS = 0529
 CS = 0520
 IP = 0002F
 FL = 0202

MU UP EI DS
 NC NA PE C

DS:00000000 00 00 00 00 00 00 20 00-EH 97 02 C7 06 E6 01 20
 DS:00100000 00 00 00 00 E4 01 26 00 16 20 00 EB 78 02 07 92

>

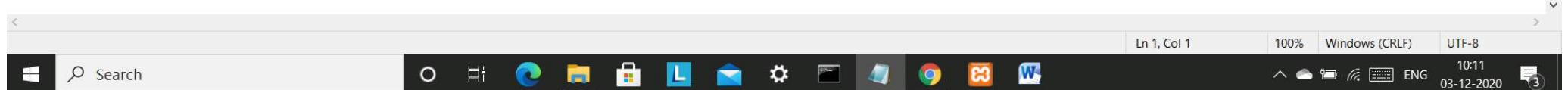
<F8=Trace> <F10=Step> <F5=Go> <F6=Windows> <F3=Display>

ASCII - Notepad

File Edit Format View Help

```
.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
    and bl,0fh
    cmp bl,0ah
    jc digit1
    add bl,07h
digit1: add bl,30h
    mov res+1,bl

    mov ah,00h
    mov al,03h    ; TEXT MODE
    int 10h
```



ASCII - Notepad

File Edit Format View Help

```
cmp bl,0ah
jc digit1
add bl,07h
digit1: add bl,30h
mov res+1,bl

mov ah,00h
mov al,03h ; TEXT MODE
int 10h

mov ah,02h ; SET THE CURSOR POS
mov bh,00h ; PAGE NUMBER
mov dh,0ch ; ROW (00 IS TOP
mov dl,28h ; COLUMN VAL
int 10h

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

disp proc near
mov ah,09h
int 21h
ret
disp endp
end
```



Program to find ASCII equivalent.

.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~~h~~

mov ah, 01h

int 21h

mov bl, al

mov cl, 4

shl al, cl

cmp al, 0ah

jc digit

Add al, 07h

digit : add al, 30h

mov res, al

and bl, 0fh

cmp bl, 0ah

jc digit1

add bl, 07h

digit1 : add bl, 30h

mov res1, bl

mov ah, 02h

mov bh, 00h

mov dh, 0ch

mov dl, 28h

int 10h

mov res2, '\$'

lea dx, res

call disp

mov ah, 4ch
int 21h

dish farc near
mov ah, 09h
int 21h

■ ~~ret~~ ret

dish end P
end

73

C:\MP_LAB\MASM>

checking if a string is a palindrome or not

.model small

display macro msg

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

end m

.data

msg1 db 0dh, 0ah, "Enter string: \$"

msg2 db 0dh, 0ah, "Reverse string: \$"

msg3 "Input string is a palindrome \$"

msg4 "Input string is not a palindrome \$"

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

cmpl al, 0dh

je next

mov [SI], AL

~~inc~~ 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 Rotating mov.di,
offset string

Back : mov ah, [SI]
mov [DI], ah
dec SI
inc DI

dec CH

jnz Back

mov [DI], byte PTR '\$'

Display msg 2

Display Rotating

mov SI, offset string

AG : mov Ah, [SI]

cmp AH, [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.

```
C:\>MSD\pal
```

```
Enter string: lam  
Reverse string: am  
Input string is not a palindrome.  
C:\>MSD\edit pal.asm
```

```
C:\>MSD\pal
```

```
Enter string: madam  
Reverse string: madam  
Input string is a palindrome.  
C:\>MSD\P
```

```
strcomp - Notepad
File Edit Format View Help
Minimize

.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

Ln 1, Col 1 100% Windows (CRLF) UTF-8
Search             10:19 ENG 03-12-2020
```

strcomp - Notepad

File Edit Format View Help

```
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 STATEMENT
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
```



strcomp - Notepad
File Edit Format View Help

```
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
```



Program to compare two strings and see if they are equal or not.

.model small

display macro msg

lea dx, msg

mov ah, 09h

int 21h

endm

.data

msg1 db 0DH, 0AH "Enter first string : \$"

msg2 db 0AH, 0AH "Enter second string ; \$"

msg3 db 00H, 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 string

display msg2

mov sp, offset String2

call readstr

push bx

push cx

display msg3

mov al, bl

call len-dis

display msg4

mov al, cl

call Im-dis

POP CX

POP BX

cmp cl, dl ; Compare the length
JNE fail ; If length are equal, process next statement
mov SI, offset string1
mov DI, offset string2

add

CHK : mov AL, [SI]

cmp AL, [DI]

JNE fail

inc SB

inc DI

dec CL

JNZ chk

display msg5

jmp final

len-dis proc near

XOR AH, AH

ADD AL, OOH

AAM

add AX, 3030H

MOV BH, AL

Mov AL, AH

Mov AH, 02H

int 21h

Mov DL, BH

Mov CH, 02H

int 21h

RET

len-dis ~~end~~ end P

Readstrs proc near

XOR CL, CL

Back: mov AH, DH

int 21h

cmp al, dh

je finish

mov [SI], al

inc SI

inc CL

jmp back

finish : mov [SI], byte ptr '\$'

ret

readstr endp

fail : display msg

final : mov ah, 4ch

int 21h

end start.

51618 + 464926 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\MP_LAB\MASM>link strcomp;;

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

LINK : warning L4021: no stack segment

C:\MP_LAB\MASM>cv strcomp

ENTER FIRST STRING : sohan

ENTER SECOND STRING : steve

LENGTH OF FIRST STRING: 05

LENGTH OF SECOND STRING: 05

---STRINGS ARE NOT EQUAL---

C:\MP_LAB\MASM>

ncr - Notepad

File Edit Format View Help

```
.model small
.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
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
```

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

Search          

10:22 03-12-2020 ENG 

ncr - Notepad

File Edit Format View Help

```
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
```



ncr - Notepad

File Edit Format View Help

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



nCr using recursion:

MODEL SMALL

~~DATA~~

n dw 4

r dw 2

ncr dw 0

~~CODE~~

mov ax, @data

mov ds, ax

mov ax, n

mov bx, r

call ncfr

call disp

jmp final

ncfr proc near

cmp ax, bx ; n=r

je res1

cmp bx, 0 ; n=0

je res0

cmp bx, 1 ; n=1

je res1

dec ax ; n=n-1

cmp bx, ax

je incr

push ax

push bx

call ncfr

pop bx

pop ax

dec bx

push ax

push bx

call ncfrs

pop bx

pop ax

ret

mes1 : inc nc

get

incn : inc nc

scn : add nc, ax ; $1+2+3 = 6$

get

ncfrs endh

disk fnec near

mov bx, nc

add bx, 3030h

mov dl, bh

mov ah, 02h

int 21h

mov dl, bl

mov ah, 02h

int 21h

get

disk endh

final : mov ah, 4ch

int 21h

end

```
51758 + 464786 Bytes symbol space free  
0 Warning Errors  
0 Severe Errors  
C:\MP_LAB\MASM>link ncr:  
Microsoft (R) Overlay Linker Version 3.60  
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.  
LINK : warning L4021: no stack segment  
C:\MP_LAB\MASM>ncr  
6  
C:\MP_LAB\MASM>
```

time - Notepad

File Edit Format View Help

Minimize

```
.Lab Programe_5
.MODEL SMALL
DISP MACRO MSG
LEA DX,MSG
MOV AH,09H
INT 21H
ENDM

.DATA
TIME DB 20H DUP(?)
MSG1 DB 0DH,0AH,"CURRENT TIME : $"

.CODE
MOV AX,@DATA
MOV DS,AX
;CLEAR SCREEN
MOV AH,00H
MOV AL,03H
INT 10H

AG: MOV BH,00H
    MOV DH,01H
    MOV DL,01H
    MOV AH,02H
    INT 10H
    LEA SI,TIME
    MOV AH,2CH
    INT 21H

MOV AL,CH
```

Ln 1, Col 1

100% Windows (CRLF) UTF-8

Search

○ ⊞ 🌐 📂 🛍️ L 📎 🚧 🖥 🌐 🌐 🌐

10:25
03-12-2020

time - Notepad

File Edit Format View Help

```
MOV AL,CH
AAM
ADD AX,3030H
MOV [SI],AH
INC SI
MOV [SI],AL
INC SI
MOV [SI],BYTE PTR '"'
INC SI

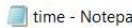
MOV AL,CL
AAM
ADD AX,3030H
MOV [SI],AH
INC SI
MOV [SI],AL
INC SI
MOV [SI],BYTE PTR '"'
INC SI

MOV AL,DH
AAM
ADD AX,3030H
MOV [SI],AH
INC SI
MOV [SI],AL
INC SI
MOV [SI],BYTE PTR '$'

DISP MSG1
```

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

Search             ENG 10:26 03-12-2020 



time - Notepad

File Edit Format View Help

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 '\$'

DISP MSG1

DISP TIME

MOV AH,0BH

INT 21H

CMP AL,00H

JE AG

MOV AH,4CH

INT 21H

END

<

Ln 1, Col 1

100%

Windows (CRLF)

UTF-8



Search



10:26
03-12-2020 ENG

Display the system time

.model small

.code

mov ah, 2ch

int 21h

mov al, ch

aam

mov bx, ax

call dish

mov dl, 20h

mov ah, 02h

int 21h

mov al, cl

aam

mov BX, AX

call dish

mov dl, 20h

mov ah, 02h

int 21h

mov al, dh

aam

mov bx, ax

call dish

mov ah, 40h

int 21h

dish proc near

mov dl, 6h

add dl, 30h

mov ah, 0ah

int 21h

mov dl, 66

add dl, 30h

MOV ah, 02h

INT 21h

RET

DISP END P

END .

0 Warnings 0 Errors
0 Severe Errors

C:\MSDOS>link timedsp

Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1984-1987. All rights reserved

LINK : warning L4821: no stack segment

C:\MSDOS>link timedsp

15 24:09

C:\MSDOS>edit timedsp.asm

C:\MSDOS>link timedsp

15 24:09

C:\MSDOS>edit timedsp.asm

C:\MSDOS>link timedsp

15 26:36

C:\MSDOS>edit timedsp.asm

C:\MSDOS>link timedsp

15 27:09

C:\MSDOS>

DCOUNT - Notepad
File Edit Format View Help

```
.MODEL SMALL
.CODE
MOV CL,00
MOV AH,00H
MOV AL,03H
INT 10H
BACK:MOV BH,00H
MOV DH,00H
MOV DL,00H
MOV DL,00H
MOV AH,02H
INT 10H
MOV AL,CL
ADD AL,00H
AAM      ;0000
ADD AX,3030H ;3030
MOV CH,AL
MOV DL,AH
MOV AH,02H
INT 21H
MOV DL,CH
MOV AH,02H
INT 21H
CALL DELAY
INC CL
XOR AX,AX
CMP CL,151D
JNE BACK
JE LAST
```



DCOUNT - Notepad
File Edit Format View Help

```
INT 21H
CALL DELAY
INC CL
XOR AX,AX
CMP CL,151D
JNE BACK
JE LAST
```

```
DELAY PROC NEAR
PUSH AX
PUSH BX
PUSH CX
MOV CX,00FFH
AG:MOV BX,0FFFH
AG1:NOP
DEC BX
JNZ AG1
DEC CX
JNZ AG
POP CX
POP BX
POP AX
RET
ENDP
LAST:MOV AH,4CH
INT 21H
END
```



Simulate Decimal up counter

classmate

Date _____
Page _____

[0-99]

• MODEL SMALL

• CODE

MOV CL, 00

MOV AH, 00H

MOV AL, 03H

INT 10H

BACK: MOV BH, 00H

MOV DH, 00H

MOV DL, 00H

MOV DL, 00H

MOV AH, 02H

INT 10H

MOV AL, CL

ADD AL, 00H

AAM

~~ADD AX, 3030H~~

MOV CH, AL

MOV DL, AH

MOV AH, 02H

INT 21H

MOV DL, CH

MOV AH, 02H

INT 21H

CALL DELAY

DNC CL

XOR AX, AX

CMP CL, 100D

JNE BACK

JB LAST

DELAY PROC NEAR

PUSH AX

PUSH BX
PUSH CX
MOV CX, 00FFH
AG : MOV BX, 0FFPH
AGL : NOP
DEC BX
JNZ AG1
DEC CX
JNZ AG
POP CX
POP BX
POP AX
RET
DELAY ENDP
LAST : MOV AH, 4CH
INT 21H
END

77
CUT LADIES -

CURSOR - Notepad
File Edit Format View Help

```
.MODEL SMALL
DISP MACRO MSG
LEA DX,MSG
MOV AH,09H
INT 21H
ENDM
.DATA
ROW DB 02 DUP(0)
COL DB 02 DUP(0)
MSG1 DB 0DH,0AH," ENTER X CO-ORDINATED : $"
MSG2 DB 0DH,0AH," ENTER Y CO-ORDINATED : $"
MSG3 DB 0DH,0AH," CURSOR DISPLAYED AT THE CORRECT CO-ORDINATES: $"
.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 SI
MOV AL,[SI]
SUB AX,3030H
AAD
MOV DH,AL ;ROW
MOV SI,OFFSET COL
MOV AH,[SI]
```



CURSOR - Notepad
File Edit Format View Help

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

```
MOV AH,00
MOV AL,03H
INT 10H
MOV AH,02H
INT 10H
DISP MSG3
JMP FINAL
```

```
READ PROC NEAR
MOV CX,02H
BACK:MOV AH,01H
INT 21H
MOV [SI],AL
INC SI
DEC CX
```



CURSOR - Notepad
File Edit Format View Help

SUB AX,3030H

AAD

MOV DL,AL

MOV AH,00

MOV AL,03H

INT 10H

MOV AH,02H

INT 10H

DISP MSG3

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



Program for cursor to point at given location.

.model small

disp macro msg

Lea px, msg

Mov ah, 09h

Int 21h

endm

.data

Row db 02 dup(0)

Col db 02 dup(0)

msg1 db 0dh, 0ah, "Enter X-Coordinate : \$"

msg2 db 0dh, 0ah, "Enter Y- coordinate : \$"

msg3 db 0dh, 0ah, "cursor is pointing exactly \$"

.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]

Cinc si

Mov al, [si]

Sub ax, 3030h

AAD

Mov dh, al ; now

Mov si, offset col

Mov ah, [si]

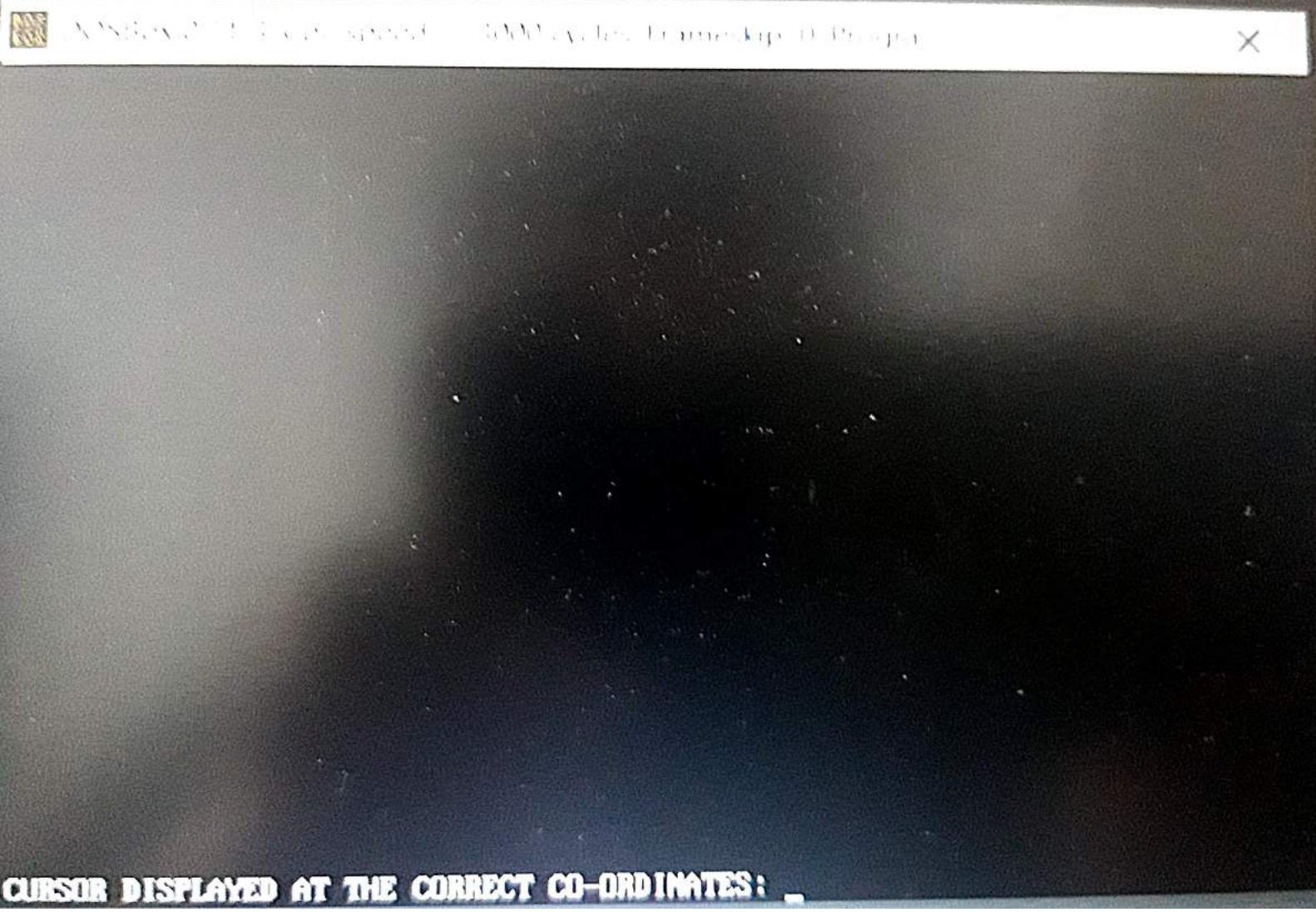
inc SI
mov al, [SI]
sub ax, 3030H
AAD
MOV DL, AL ; vol
mov AH, 00
mov al, 03H
int 10H
mov ah, 02H
int 10H → dup meg 3
JMP final

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

→ mov SI, offset col
mov ah, [SI]
inc SI
mov al, [SI]
sub ax, 3030H
AAD
MOV DL, AL.



CURSOR DISPLAYED AT THE CORRECT CO-ORDINATES:

CRTDEL - Notepad

File Edit Format View Help

```
.MODEL SMALL
DISP MACRO MSG
LEA DX,MSG
MOV AH,09H
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 SUCCESSFULLY$"
MSG6 DB 0DH,0AH,"DELETION FAILED$"
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,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
```



CRTDEL - Notepad
File Edit Format View Help

```
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],'$'
    LEA DX,FNAME2

MOV AH,41H
INT 21H
JC DFAIL
DISP MSG5
JMP LAST
DFAIL: DISP MSG6
LAST:MOV AH,4CH
INT 21H
END
```



Creation and deletion of file.

• MODEL SMALL

DISP MACRO MSG

LEA DX, MSG

MOV AH, 09H

INT 21H

ENDM

• DATA

MSG_1 DB 0DH, 0AH, "ENTER THE FILE NAME FOR CREATION: \$"

MSG_2 DB 0DH, 0AH, "FILE CREATED SUCCESSFULLY \$"

MSG_3 DB 0DH, 0AH, "FILE CREATION FAILED \$"

MSG_4 DB 0DH, 0AH, "ENTER THE FILE NAME FOR DELETION: \$"

MSG_5 DB 0DH, 0AH, "FILE DELETED SUCCESSFULLY \$"

MSG_6 DB 0DH, 0AH, "DELETION FAILED \$"

FNAME1 DB 10 DUP(0)

FNAME2 DB 10 DUP(0)

• CODE

MOV AX, @DATA

MOV DS, AX

DISP MSG_1

MOV SI, 00

BACK1: MOV AH, 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

JG OFAIL

DISP MSG 2

JMP DBL

DFAIL: DISP MSG 3

DBL: DISP MSG 4

MOV SI, 00

BACK2: MOV AH, 01H

INT 21H

CMP AL, 00H

JE NEXT2

MOV FNAMES2[SI], AL

INC SI

JMP BACK2

NEXT2: MOV FNAMES2[SI], '\$'

LEA DX, FNAMES2

MOV AH, 4DH

PNT 21H

JC DFAIL

DISP MSG 5

JMP LAST

DFAIL: DISP MSG 6

LAST: MOV AH, 4CH

PNT 21H

END.

- + ⌂ | A) Read aloud | ⌂ Draw ▾ ⌂ Highl

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

51622 + 464922 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\MP_LAB\MASM>link crtdel;;

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

LINK : warning L4021: no stack segment

C:\MP_LAB\MASM>crtdel

ENTER THE FILE NAME FOR CREATION:-sohan

FILE CREATED SUCCESSFULLY

ENTER THE FILE NAME FOR DELETION:-sohan

FILE DELETED SUCCESSFULLY

C:\MP_LAB\MASM>_