Password Checking

```
data       segment
password db 'mpmclab'
len equ ($-password)
rmsg1 db 10,13, 'enter your password:$'
cmsg2 db 10,13, 'good, correct password !!$'
icmsg3 db 10,13,'sorry,  incorrect password!$'
new db 10,13,'$'
inst db 10 dup(0)
data       ends
code       segment
           assume cs:code, ds:data
start:     mov ax, data
           mov ds, ax
           lea dx, rmsg1
           mov ah, 09h
           int 21h
           mov si, 00h
back1:     mov ah, 08h
           int 21h
           cmp al, 0dh
           je skip1
           mov [inst+si], al
           mov dl, '*'
           mov ah, 02h
           int 21h
           inc si
           jmp back1
skip1:     mov bx, 00h
           mov cx, len
check:     mov al, [inst+bx]
           mov dl, [password+bx]
           cmp al, dl
           jne fail
           inc bx
           loop check
           lea dx, cmsg2
           mov ah, 09h
           int 21h
           jmp finish
fail:      lea dx, icmsg3
           mov ah, 09h
           int 21h
finish:    mov ah, 4ch
           int 21h
code       ends
           end start
           end
```


RAM Size

```
cr equ 0dh
lf equ 0ah
data segment
table db '0123456789abcdef'
message db ' this program is used to find the
size of the ram memory in kbytes'
db ' memory size is returned in ax register'
db cr,lf, '$'
memorysize dw ?
displaysize db ' the ram memory size is:'
asciisize db 4 dup(?)
db cr,lf,'$'
data ends
code segment
assume cs:code, ds:data
start: mov ax, data
mov ds,ax
mov ah,09h
lea dx,message
int 21h
mov ah,088h
int 15h
mov memorysize, ax
lea si, asciisize
add si,3
lea bx, table
mov ax, memorysize
and ax,0000fh
xlat
mov [si],al
dec si
mov ax, memorysize
and ax,000f0h
mov cl,04h
shr ax,cl
xlat
mov [si],al
dec si
mov ax, memorysize
and ax,00f00h
mov cl,08h
shr ax,cl
xlat
```

```
mov [si],al
dec si
mov ax, memorysize
and ax,0f000h
mov cl,0ch
shr ax,cl
xlat
mov [si],al
mov ah,09h
lea dx, displaysize
int 21h
quit: mov al,0
mov ah,04ch
int 21h
code ends
end start
```

System Date

```
        .model small
        .data
        .code
start:  mov ax, @data
        mov ds, ax

date:   mov ah, 2ah
        int 21h
        mov al, dl
        aam
        mov bx, ax
        call disp
        mov dl,'/'
        mov ah, 02h
        int 21h

month:  mov ah, 2ah
        int 21h
        mov al, dh
        aam
        mov bx, ax
        call disp
        mov dl,'/'
        mov ah, 02h
        int 21h

year:   mov ah, 2ah
        int 21h
```

```
        mov al, 14h
        aam
        mov bx, ax
        call disp
        mov ah, 2ah
        int 21h
        add cx, 0f830h
        mov al, cl
        aam
        mov bx, ax
        call disp
        mov ah, 4ch
        int 21h

        disp proc
        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 start
```

System Time

```
        .MODEL SMALL
        .DATA
        .CODE
START:          MOV AX, @DATA
        MOV DS, AX
;HOUR PART
HOUR:   MOV AH, 2CH
        INT 21H
        MOV AL, CH
        AAM
        MOV BX, AX
        CALL DISP
        MOV DL, ':'
        MOV AH, 02H
        INT 21H
;MINUTES PART
MINUTES:   MOV AH, 2CH
        INT 21H
        MOV AL, CL
```

```
            AAM
            MOV BX, AX
            CALL DISP
            MOV DL, ':'
            MOV AH, 02H
            INT 21H
;SECONDS PART
SECONDS:    MOV AH, 2CH
            INT 21H
            MOV AL, DH
            AAM
            MOV BX, AX
            CALL DISP
            MOV AH, 4CH
            INT 21H

            DISP PROC
            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 START
```



```
C:\MASM611\BIN>time
23:10:30
```

Digital Clock

```
CODE  SEGMENT
        ASSUME      CS:CODE
START:      MOV     AX,00000H
        INT     10H
TOP:    MOV     AH,02H
        MOV     BH,00H
        MOV     DH,11
        MOV     DL,14
        INT     10H
        MOV     AH, 02CH
        INT     21H
        MOV     AL,CH
        AAM
        MOV     BX,AX
        MOV     DL,BH
        ADD     DL,30H
        MOV     AH,02H
        INT     21H
        MOV     DL,BL
        ADD     DL,30H
        MOV     AH,02H
        INT     21H
MIN:    MOV     DL,':'
        MOV     AH,02H
        INT     21H
        MOV     AL,CL
        AAM
        MOV     BX,AX
        MOV     DL,BH
        ADD     DL,30H
        MOV     AH,02H
        INT     21H
        MOV     DL,BL
        ADD     DL,30H
        MOV     AH,02H
        INT     21H
SEC:    MOV     DL, ':'
        MOV     AH,02H
        INT     21H
        MOV     AL,DH
        AAM
        MOV     BX,AX
        MOV     DL,BH
        ADD     DL,30H
        MOV     AH,02H
        INT     21H
        MOV     DL,BL
        ADD     DL,30H
        MOV     AH,02H
        INT     21H
        JMP     TOP
QUIT:   MOV     AX,04C00H
        INT     21H
CODE  ENDS
        END     START
```



```
23:12:40
```