1. Write an ALP to read password & validate the user.

2. Write an ALP to display the status of keys described in 02H functions of BIOS INT 16H.

3. Write an ALP to simulate CLS (Clear Screen) command.

4. Write an ALP to simulate DEL (Delete file) and REN (Rename file) command.

5. Write an ALP to display the attribute and date/ time of any file.

1) install dosbox

2) exatract 8086\_assembler in some drive and give some name to floder  eg. d:\8086

3) write program using notepad save it

4) open dosbox

write

mount c d:\8086

5) c:\

6) masm programname.asm

7) link programname.obj

8) programname.exe

1) Program to check status of caps lock key

data segment

capson db 10,13,"capslock is on$"

capsoff db 10,13,"capslock is off$"

data ends

code segment

assume cs:code,ds:data

start: mov ax,data

mov ds,ax

mov ah,02h ;Read status of keys on keyboard

int 16h

mov cx,0002h

rcl al,cl ;Rotate key code to poll status keys bitwise

jnc off ;by rotating bits by 2 position if carry is not generated then it indicates capslock is off

mov ah,09h

lea dx,capson

int 21h

jmp exit

off: mov ah,09h

lea dx,capsoff

int 21h

exit: mov ah,4ch

int 21h

code ends

end start

2) Program to check status of insert key

data segment

inson db 10,13,"insert is on$"

insoff db 10,13," insert is off$"

data ends

code segment

assume cs:code,ds:data

start: mov ax,data

mov ds,ax

mov ah,02h ;Read status of keys on keyboard

int 16h

mov cx,0001h

rcl al,cl ;Rotate key code to poll status keys bitwise

jnc off ;by rotating bits by 1 position if carry is not generated then it indicates insert is off

mov ah,09h

lea dx, inson

int 21h

jmp exit

off: mov ah,09h

lea dx,insoff

int 21h

exit: mov ah,4ch

int 21h

code ends

end start

3) Program to check status of numlock key

data segment

numon db 10,13,"numlock is on$"

numoff db 10,13," numlock is off$"

data ends

code segment

assume cs:code,ds:data

start: mov ax,data

mov ds,ax

mov ah,02h ;Read status of keys on keyboard

int 16h

mov cx,0003h

rcl al,cl ;Rotate key code to poll status keys bitwise

jnc off ;by rotating bits by 3 position if carry is not generated then it indicates numlock is off

mov ah,09h

lea dx, numon

int 21h

jmp exit

off: mov ah,09h

lea dx,numoff

int 21h

exit: mov ah,4ch

int 21h

code ends

end start

4) Program to check status of scrolllock key

data segment

scrollon db 10,13," scrolllock is on$"

scrolloff db 10,13," scrolllock is off$"

data ends

code segment

assume cs:code,ds:data

start: mov ax,data

mov ds,ax

mov ah,02h ;Read status of keys on keyboard

int 16h

mov cx,0004h

rcl al,cl ;Rotate key code to poll status keys bitwise

jnc off ;by rotating bits by 4 position if carry is not generated then it indicates scrolllock is off

mov ah,09h

lea dx, scrollon

int 21h

jmp exit

off: mov ah,09h

lea dx, scrolloff

int 21h

exit: mov ah,4ch

int 21h

code ends

end start

5) Program to check status of alt key

data segment

alton db 10,13," alt is on$"

altoff db 10,13," alt is off$"

data ends

code segment

assume cs:code,ds:data

start: mov ax,data

mov ds,ax

mov ah,00h ;Read key from keyboard

int 16h

mov ah,02h ;Read status of keys on keyboard

int 16h

mov cx,0005h

rcl al,cl ;Rotate key code to poll status keys bitwise

jnc off ;by rotating bits by 5 position if carry is not generated then it indicates alt is off

mov ah,09h

lea dx, altlon

int 21h

jmp exit

off: mov ah,09h

lea dx, altoff

int 21h

exit: mov ah,4ch

int 21h

code ends

end start

6) Program to check status of ctrl key

data segment

ctrlon db 10,13," ctrl is on$"

ctrloff db 10,13," ctrl is off$"

data ends

code segment

assume cs:code,ds:data

start: mov ax,data

mov ds,ax

mov ah,00h ;Read key from keyboard

int 16h

mov ah,02h ;Read status of keys on keyboard

int 16h

mov cx,0006h

rcl al,cl ;Rotate key code to poll status keys bitwise

jnc off ;by rotating bits by 6 position if carry is not generated then it indicates ctrl is off

mov ah,09h

lea dx, ctrlon

int 21h

jmp exit

off: mov ah,09h

lea dx, ctrloff

int 21h

exit: mov ah,4ch

int 21h

code ends

end start

7) Program to check status of left shift key

data segment

lshifton db 10,13," left shift is on$"

lshiftoff db 10,13," left shift is off$"

data ends

code segment

assume cs:code,ds:data

start: mov ax,data

mov ds,ax

mov ah,00h ;Read key from keyboard

int 16h

mov ah,02h ;Read status of keys on keyboard

int 16h

mov cx,0007h

rcl al,cl ;Rotate key code to poll status keys bitwise

jnc off ;by rotating bits by 6 position if carry is not generated then it indicates left shift is off

mov ah,09h

lea dx, lshifton

int 21h

jmp exit

off: mov ah,09h

lea dx, lshiftoff

int 21h

exit: mov ah,4ch

int 21h

code ends

end start

7) Program to check status of right shift key

data segment

rshifton db 10,13," right shift is on$"

rshiftoff db 10,13," right shift is off$"

data ends

code segment

assume cs:code,ds:data

start: mov ax,data

mov ds,ax

mov ah,00h ;Read key from keyboard

int 16h

mov ah,02h ;Read status of keys on keyboard

int 16h

mov cx,0008h

rcl al,cl ;Rotate key code to poll status keys bitwise

jnc off ;by rotating bits by 6 position if carry is not generated then it indicates right shift is off

mov ah,09h

lea dx, rshifton

int 21h

jmp exit

off: mov ah,09h

lea dx, rshiftoff

int 21h

exit: mov ah,4ch

int 21h

code ends

end start

8) Program to set cursor position

code segment

assume cs:code

start: mov ah,02h ;set cursor position

mov bh,00h ;Set page number

mov dh,40h ;Set row number

mov dl,00h ;Set column number

int 10h

mov ah,4ch

int 21h

code ends

end start

9) **WRITE A PROGRAM TO CLEAR THE SCREEN AND MAKE BLUE FOREGROUND.**

data segment

msg db 10,13,"hello how r u?$"

data ends

code segment

assume cs:code,ds:data

start: mov ax,data

mov ds,ax

mov ah,06h ;used to scroll screen upwards

mov bh,71h ;background color cyan and foreground color blue

mov cx,0000h ;specify starting row no. and column no.

mov dx,184fh ;specify starting row no. and column no.

int 10h

mov ah,09h

lea dx,msg

int 21h

mov ah,4ch

int 21h

code ends

end start

10) Program to scroll up screen

code segment

assume cs:code

start: mov ax,data

mov ds,ax

mov ah,06h ;used to scroll screen upwards

mov ah,00h ;scroll whole screen

mov bh,71h ;background color cyan and foreground color blue

mov cx,0000h ;specify starting row no. and column no.

mov dx,184fh ;specify starting row no. and column no.

int 10h

mov ah,02h ;set cursor position at starting position

mov ah,00h ;page no. is loaded

mov al,00h ;specify row no.

mov dl,00h ;specify column no.

int 10h

mov ah,4ch

int 21h

code ends

end start

11) Program to create a file open file and write content on file.

data segment

startmsg db 10,13,"this program will create a file demo.txt on c drive$"

endmsg db 10,13,"file created $"

cerror db 10,13,"Error in file creation$"

oerror db 10,13,"Error in opening file$"

werror db 10,13,"Error in writing file$"

filename db "D:\demo.txt",0

handle dw ?

content db "hello, how r u?, welcome to masm"

data ends

code segment

assume cs:code,ds:data

start: mov ax,data

mov ds,ax

lea dx,startmsg

mov ah,09h

int 21h

lea dx,filename

mov cx,0 ;clear cx to make ordinary file

mov ah,3Ch ;create a file

int 21h

jc createerror ;jump if error in creating file

lea dx,filename

mov al,2h ;Access mode read and write

mov ah,3Dh ;Open file

int 21h

jc openerror ;jump if error in opening file

mov handle,ax ;Save the file handle

lea dx,content

mov bx,handle ;handle for file

mov cx,20h ;20 bytes to be written

mov ah,40h ;write to the file

int 21h

jc writeerror ;jump if error in writing file

cmp ax,cx ;check whether all data was written

jne writeerror ;jump if error in writing file

mov bx,handle

mov ah,3Eh ;close the file

int 21h

lea dx,endmsg

mov ah,09h

int 21h

jmp exit

createerror: lea dx,cerror

mov ah,09h

int 21h

jmp exit

writeerror: lea dx,werror

mov ah,09h

int 21h

jmp exit

openerror: lea dx,oerror

mov ah,09h

int 21h

jmp exit

exit: mov ah,4Ch

int 21h

code ends

end start

12) Program to copy one file from another.

data segment

src db "D:\demo.txt",0

dest db "E:\demo1.txt",0

buffer db 512 dup(0)

errmsg db 10,13,"Error$"

shandle dw ?

dhandle dw ?

count dw ?

data ends

code segment

assume cs:code,ds:data

start: mov ax,data

mov ds,ax

mov ah,3ch ;create destination file

lea dx,dest

mov cl,0

int 21h

mov bx,ax

mov ah,3eh ;close file

int 21h

mov ah,3dh ;open destination file

mov al,1 ;open for writing mode

lea dx,dest

int 21h

mov dhandle,ax

jc error

mov ah,3dh ;open source file

mov al,0 ;open for reading mode

lea dx,src

int 21h

mov shandle,ax

jc error

lea dx,buffer

readloop: mov ah,3fh ;read file

mov bx,shandle

mov cx,512

int 21h

jc error

mov count,ax ;count no. of charactes in source file

writeloop: mov ah,40h ;write file

mov bx,dhandle

mov cx,count

int 21h

cmp count,512

jl close

jmp readloop

close: mov ah,3eh ;close source file

mov bx,shandle

int 21h

mov ah,3eh ;close destination file

mov bx,dhandle

int 21h

jmp exit

error: mov ah,09h

lea dx,errmsg

int 21h

exit: mov ah,4ch

int 21h

code ends

end start

13) Program to delete file

data segment

fname db "E:\demo1.txt",0

errmsg db 10,13,"error occured$"

data ends

code segment

assume cs:code,ds:data

start: mov ax,data

mov ds,ax

mov ah,43h ;Obtain attribute of file (file handle)

mov al,00h ;open file in reading mode

lea dx,fname

int 21h

mov ah,41h ;delete file

lea dx,fname

int 21h

jnc exit

mov ah,09h

lea dx,errmsg

int 21h

exit: mov ah,4ch

int 21h

code ends

end start

14) Program to rename file

data segment

fname db "D:\demo.txt",0

data ends

extra segment

nname db "d:\new.txt",0

extra ends

code segment

assume cs:code,ds:data,es:extra

start: mov ax,data

mov ds,ax

mov ax,extra

mov es,ax

mov ah,56h ;file handle to rename file

lea dx,fname

lea di,nname

int 21h

mov ah,4ch

int 21h

code ends

end start

15) Multiple choice program to delete and rename file

data segment

dfile db "E:\demo1.txt",0

oname db "D:\new.txt",0

msg1 db 10,13,"Menu 1:Delete 2:rename 3:exit$"

msg2 db 10,13,"Enter ur choice$"

choice db ?

data ends

extra segment

nname db "D:\demo.txt",0

extra ends

code segment

assume cs:code,ds:data,es:extra

start: mov ax,data

mov ds,ax

mov ax,extra

mov es,ax

mov ah,09h

lea dx,msg1

int 21h

up: mov ah,09h

lea dx,msg2

int 21h

mov ah,01h ;read choice from keyboard

int 21h

cmp al,31h

jz del

cmp al,32h

jz rename

cmp al,33h

jz exit

del: mov ah,41h

lea dx,dfile

int 21h

jmp up

rename: mov ah,56h

lea dx,oname

lea di,nname

int 21h

jmp up

exit: mov ah,4ch

int 21h

code ends

end start