**P229 监测点11.3**

(1)答

jb s0

ja s0

1. 答

jna s0

jnb s0

**P234实验11**

assume cs:codesg

datasg segment

db"Begjinner's All-purpose Symbolic Instruction Code.",0

datasg ends

codesg segment

begin:mov ax,datasg

mov ds,ax

mov si,0

call letterc

mov ax,4c00h

int 21h

letterc: mov cx,ds:[si]

jcxz s0

mov ax,cx

mov ah,0

cmp al,61h

jb s0

cmp al,7Ah

ja s0

and ax,11011111B

mov ds:[si],al

loop letterc

s0: ret

codesg ends

end begin

**P251实验12**

assume cs:code

code segment

start:mov ax,cs

mov ds,ax

mov si,offset do0

mov ax,0

mov es,ax

mov di,200h

mov cx,offset do0end-offset do0

cld

rep movsb

mov ax,0

mov es,ax

mov word ptr es:[0],200h

mov word ptr es:[2],0

mov ax,4c00h

int 21h

do0:jmp short do0start

db "divide error!"

do0start :mov ax,cs

mov ds,ax

mov si,202h

mov ax,0b800h

mov es,ax

mov di,12\*160+33\*2

mov cx,13

s:mov al,[si]

mov es:[di],al

mov byte ptr es:[di+1],2

inc si

add di,2

loop s

mov ax,4c00h

int 21h

do0end:nop

code ends

end start

**P257监测点13.1**

(1)128

**(2)**

assume cs:code  
data segment  
    db 'conversation',0  
data ends  
code segment  
start:  
        mov ax,cs  
        mov ds,ax  
        mov si,offset nr  
        mov ax,0  
        mov es,ax  
        mov di,200h  
        mov cx,offset nrend-offset nr  
        cld  
        rep movsb  
   
        mov word ptr es:[7ch\*4],200h  
        mov word ptr es:[7ch\*4+2],0   
           
        mov ax,data  
        mov ds,ax  
        mov si,0  
        mov ax,0b800h  
        mov es,ax  
        mov di,12\*160  
s:      cmp byte ptr [si],0  
        je ok  
        mov al,[si]  
        mov es:[di],al  
        inc si  
        add di,2  
        mov bx,offset s-offset ok  
        int 7ch  
ok:     mov ax,4c00h  
        int 21h  
   
nr:     push bp

        mov bp,sp  
        add [bp+2],bx  
nrret:  pop bp  
        iret             
  
nrend:  nop

code ends  
end start

**P259监测点13.2**

(1)错误BIOS是不可写的不能往里边写程序

(2)错误，19好终端是引导操作系统的必须在操作系统还没有执行前提供

**P262实验13**

(1)答：

1. asm

assume cs:code  
  
data segment  
db "Welcome to masm!", 0  
data ends  
  
code segment  
start: mov dh, 10  
mov dl, 10  
mov cl, 2  
mov ax, data  
mov ds, ax  
mov si, 0  
int 7ch  
  
mov ax, 4c00h  
int 21h   
  
code ends  
end start

1. asm

assume cs :code  
  
code segment  
start:  
mov ax,cs  
mov ds,ax  
mov si,offset do0  
  
mov ax,0  
mov es,ax  
mov di,200h  
  
mov cx,offset do0end- offset do0  
  
cld  
rep movsb  
  
mov ax,0  
mov es,ax  
mov word ptr es:[7ch\*4],200h  
mov word ptr es:[7ch\*4+2],0  
  
mov ax,4c00h  
int 21h  
  
do0:  
  
push ax  
push bx  
push cx  
push dx  
push si  
  
mov ah,0  
mov al,dh  
mov dh,0Ah  
mul dh  
  
add ax,0b800h  
mov es,ax  
  
mov al,dl  
mov dl,2  
mul dl  
mov bx,ax  
  
s:  
mov al,ds:[si]  
cmp al,0  
je ok  
mov ah,cl  
  
mov es:[bx],ax  
  
inc si  
add bx,2  
jmp s  
  
ok:  
  
pop si  
pop dx  
pop cx  
pop bx  
pop ax  
  
mov ax,4c00h  
int 21h  
  
do0end:nop  
code ends   
end start

**13.2**

1.asm

assume cs:code  
code segment  
start: mov ax, 0b800H  
       mov es, ax  
       mov di, 160\*12  
         
       mov bx, s - se  
       mov cx, 80  
   s:  mov byte ptr es:[di], '!'  
       add di, 2  
       int 7cH  
   se: nop  
       mov ax, 4c00H  
       int 21H  
code ends  
end start

1. Asm

assume cs :code  
  
code segment  
start:  
mov ax,cs  
mov ds,ax  
mov si,offset do0  
  
mov ax,0  
mov es,ax  
mov di,200h  
  
mov cx,offset do0end- offset do0  
  
cld  
rep movsb  
  
mov ax,0  
mov es,ax  
mov word ptr es:[7ch\*4],200h  
mov word ptr es:[7ch\*4+2],0  
  
mov ax,4c00h  
int 21h  
  
do0:  
push si  
dec cx  
jcxz ok  
  
mov si,sp  
add ss:[si+2],bx  
  
ok:  
pop si  
iret  
  
do0end:nop  
code ends   
end start

**13.3**

assume cs:code  
  
code segment  
s1: db 'Good, better, best,', '$'  
s2: db 'Never let it rest,', '$'  
s3: db 'Till good is better,', '$'  
s4: db 'And better,best.', '$'  
s: dw offset s1, offset s2, offset s3, offset s4  
row: db 2, 4, 6, 8  
  
start: mov ax, cs  
mov ds, ax  
mov bx, offset s  
mov si, offset row  
mov cx, 4  
ok: mov bh, 0  
mov dh, ds:[si]  
mov dl, 0  
mov ah, 2  
int 10h  
  
mov dx, ds:[bx]  
mov ah, 9  
int 21h  
inc si  
add bx, 2  
loop ok  
  
mov ax, 4c00h  
int 21h   
  
code ends  
end start