汇编代码模板

1.基本模板

```
dataseg segment
    data dw 3,5,7,14,2,6
dataseg ends

codeseg segment
    assume cs:codeseg, ds:dataseg, es: dataseg
main proc far
start:
    push ds
    sub ax,ax
    push ax
    mov ax,dataseg
    mov ds,ax
    mov es,ax
;main begin




;---------------------------------------------------------------
;回车换行
crlf proc near
    mov dl,0dh
    mov ah,02h
    int 21h
    mov dl,0ah
    mov ah,02h
    int 21h
ret
crlf endp
;--------------------------------------------------------------
exit:
    ret
main endp
codeseg ends
    end start
```

2.把bx中的2进制数以16进制的形式显示在屏幕上

```
;----------------------------------------------
trans proc near
    mov ch,4

rotate:
    mov cl,4
    rol bx,cl
    mov al,bl
    and al,0fh
    add al,30h
    cmp al,3ah
    jl print
    add al,7h;'A-Z'

print:
    mov dl,al
    mov ah,02h
    int 21h
    dec ch
    jnz rotate
ret
trans endp
;--------------------------------------------
```

3.把bx中的数2进制转10进制输出

```
;-----------------------------------
binidec proc near
    mov cx,10000d
    call dec_div
    mov cx,1000d
    call dec_div
    mov cx,100d
    call dec_div
    mov cx,10d
    call dec_div
    mov cx,1d
    call dec_div
    ret
binidec endp
;-------------------------------------------
dec_div proc near
    mov ax,bx
    mov dx,0
    div cx
    mov bx,dx
    mov dl,al
    add dl,30h
    mov ah,2
    int 21h
    ret
dec_div endp
;------------------------------------------
```

4.从键盘接收10进制数并转成2进制存入bx

```
;--------------------------------------------------
tentotwo proc near
    mov bx,0
newchar:
    mov ah,1
    int 21h
    sub al,30h
    jl exit;小于0退出
    cmp al,9
    jg exit;大于9退出
    cbw
    xchg ax,bx
    mov cx,10
    mul cx
    xchg ax,bx
    add bx,ax

    jmp newchar

exit:
    ret
tentotwo endp
;-----------------------------------------
```

5.从键盘接收16进制数转成2进制并存入bx

```
;-------------------------------------
HtoB proc near
    mov bx,0
newchar:
    mov ah,1
    int 21h
    sub al,30h
    jl exit;小于0退出
    cmp al,10
    jl add_to;处于0-9，可直接计算
    sub al,27h;'a-f'
    cmp al,0ah
    jl exit;小于a退出
    cmp al,10h
    jge exit;大于f退出
add_to:
    mov cl,4
    shl bx,cl
    mov ah,0
    add bx,ax
    jmp newchar
exit:
    ret
HtoB endp
;----------------------------
```

6.将正数n插入一个已正序的正整字数组

```
dataseg segment
    x dw ?
    array_head dw 3,5,15,23,37,49,52,65,78,99
    array_end dw 105
    n dw 32
dataseg ends
;

;--------------------------------------------------
    mov ax,n
    mov x,-1
    mov si,0
```

```
compare:
    cmp array_end[si],ax
    jle insert
    mov bx,array_end[si]
    mov array_end[si+2],bx
    sub si,2
    jmp compare
insert:
    mov array_end[si+2],ax
;----------------------------------------------------
```

7.冒泡排序，从大到小

```
dataseg segment
    a dw 100,30,78,99,15,12,66,45,189,256
dataseg ends

;---------------------------------------------------------
    mov cx,10;数组中有10个数
    dec cx;外循环9次，完成每次外循环后把最小的数沉底，放到最后
loop1:
    mov di,cx;外循环次数暂存到di里
    mov bx,0
loop2:
    mov ax,a[bx]
    cmp ax,a[bx+2]
    jge continue;如果前面的数大，不交换
    xchg ax,a[bx+2]
    mov a[bx],ax
continue:
    add bx,2
    loop loop2
    mov cx,di;内循环次数等于cx，内循环用完cx后，恢复外循环的cx
    loop loop1
;-------------------------------------------------------------
```

8.每隔1秒响铃一次（中断）

```
    .model small
    .stack
    .data
count dw 1
msg db 'The bell is ringing!',13,10,'$'
;
    .code
main proc far
start:
    mov ax,@data
    mov ds,ax

;main begin
    ;取原中断向量
    mov al,1ch
    mov ah,35h
    int 21h
    ;保存原中断向量
    push es
    push bx
    push ds
    ;设置新的中断向量

    mov dx,offset ring
    mov ax,seg ring
    mov ds,ax
    mov al,1ch
    mov ah,25h
    int 21h

    pop ds
    ;打开定时器
    in al,21h
    and al,11111110b
    out 21h,al
    ;设置IF=1
    sti
    ;等待
    mov di,20000
delay:
    mov si,30000
delay1:
    dec si
    jnz delay1
    dec di
    jnz delay

    ;恢复原中断向量
    pop dx
    pop ds
    mov al,1ch
    mov ah,25h
    int 21h

    mov ax,4c00h
    int 21h
main endp
;-----------------------------------------
ring proc near
    push ds
    push ax
    push cx
    push dx

    mov ax,@data
    mov ds,ax
    sti;开中断

    dec count
    jnz exit
    ;输出信息
    mov dx,offset msg
    mov ah,09h
    int 21h

    mov dx,100
    in al,61h;获取端口开关
    and al,0fch
sound:
    xor al,02
    out 61h,al

    mov cx,1400h
wait1:
    loop wait1
    dec dx
    jne sound
    mov count,18;每隔1s

exit:
    cli;关中断
    pop dx
    pop cx
    pop ax
    pop ds
    iret
ring endp
;-----------------------------------------------------
    end start
```

9.输出目标字符串中所有待查字符串位置

```
dataseg segment
    mess1 db 'Enter keyword:','$'
    mess2 db 'Enter Sentence:','$'
    mess3 db 'Match at location:','$'
    mess4 db 'H of the sentence.','$'
    mess5 db 'No match.','$'
    offsetnum dw 0
    num dw 0
;
keyin label byte
    maxk db 10
    actk db ?
    key db 10 dup(?)
;
senin label byte
    maxs db 50
    acts db ?
    sen db 50 dup(?)
;

dataseg ends

codeseg segment
    assume cs:codeseg, ds:dataseg, es: dataseg
main proc far
start:
    push ds
    sub ax,ax
    push ax
    mov ax,dataseg
    mov ds,ax
    mov es,ax
;main begin
    lea dx,mess1;输出Enter keyword:
    mov ah,09h
    int 21h
    lea dx,keyin;读入关键字
    mov ah,0ah
    int 21h
    call crlf
;
a00:
    lea dx,mess2;输出Enter Sentence:
    mov ah,09h
    int 21h
    lea dx,senin;读入句子
    mov ah,0ah
    int 21h
    call crlf
;

;若输入句子是ctrl+C则退出
    mov di,0
    mov al,sen[di]
    cmp al,3;一定要和低字节，否则加上高字节就不是3了
    jz exit
;

    mov cl,acts
    mov ch,0
    sub cl,actk
    inc cx;cx里是外总比较次数
find:
    lea si,key
    lea di,sen
    add di,offsetnum
    push cx
    mov cl,actk;内比较次数
    mov ch,0
    cld
    repe cmpsb
    jnz next;暂时不匹配
match:
    lea bx,sen
    sub di,bx
    mov al,actk
    mov ah,0
    sub di,ax
    mov bx,di
    call trans
    lea dx,mess4
    mov ah,09h
    int 21h
    inc num
    call crlf
next:
    pop cx
    inc offsetnum
    loop find
    cmp num,0
    jnz exit
nomatch:
    lea dx,mess5
    mov ah,09
    int 21h
    jmp exit

;----------------------------------------------
trans proc near
    mov ch,4

rotate:
    mov cl,4
    rol bx,cl
    mov al,bl
    and al,0fh
    add al,30h
    cmp al,3ah
    jl print
    add al,7h;'A-Z'

print:
    mov dl,al
    mov ah,02h
    int 21h
    dec ch
    jnz rotate
ret
trans endp
;--------------------------------------------
;---------------------------------------------------------------
;回车换行
crlf proc near
    mov dl,0dh
    mov ah,02h
    int 21h
    mov dl,0ah
    mov ah,02h
    int 21h
ret
crlf endp
;--------------------------------------------------------------
exit:
    ret
main endp
codeseg ends
    end start
```

10.比较两个字符串是否相等

```
dataseg segment
    string1 db 'aaa','$'
    string2 db 'aaa',"$"
    flag dw ?
dataseg ends
```

```
codeseg segment
    assume cs:codeseg, ds:dataseg, es: dataseg
main proc far
start:
    push ds
    sub ax,ax
    push ax
    mov ax,dataseg
    mov ds,ax
    mov es,ax
;main begin
    lea si,string1
    lea di,string2
    mov cx,3
    repe cmpsb;用于比较两个字符串是否相等
    je xiangdeng
    mov flag,0
    jmp show
xiangdeng:
    mov flag,1
show:
    mov bx,flag
    call trans
    jmp exit
;---------------------------------------
trans proc near
    mov ch,4

rotate:
    mov cl,4
    rol bx,cl
    mov al,bl
    and al,0fh
    add al,30h
    cmp al,3ah
    jl print
    add al,7h;'A-Z'

print:
    mov dl,al
    mov ah,02h
    int 21h
    dec ch
    jnz rotate
ret
trans endp
;--------------------------------------
exit:
    ret
main endp
codeseg ends
    end start
```

11.查找目标字符串中有多少个待查字符串

```
dataseg segment
    mess1 db 'Enter keyword:','$'
    mess2 db 'Enter Sentence:','$'
    mess3 db 'Match at location:','$'
    mess4 db 'H of the sentence.','$'
    mess5 db 'No match.','$'
    num dw 0
    offsetnum dw 0
;
keyin label byte
    maxk db 10
    actk db ?
    key db 10 dup(?)
;
senin label byte
    maxs db 50
    acts db ?
    sen db 50 dup(?)
;

dataseg ends

codeseg segment
    assume cs:codeseg, ds:dataseg, es: dataseg
main proc far
start:
    push ds
    sub ax,ax
    push ax
    mov ax,dataseg
    mov ds,ax
    mov es,ax
;main begin
    lea dx,mess1;输出Enter keyword:
    mov ah,09h
    int 21h
    lea dx,keyin;读入关键字
    mov ah,0ah
    int 21h
    call crlf
;
a00:
    lea dx,mess2;输出Enter Sentence:
    mov ah,09h
    int 21h
    lea dx,senin;读入句子
    mov ah,0ah
    int 21h
    call crlf
;

;若输入句子是ctrl+C则退出
    mov di,0
    mov al,sen[di]
    cmp al,3;一定要和低字节，否则加上高字节就不是3了
    jz exit
;

    mov cl,acts
    mov ch,0
    sub cl,actk
    inc cx;cx里是外总比较次数
find:
    lea si,key
    lea di,sen
    add di,offsetnum
    push cx
    mov cl,actk;内比较次数
    mov ch,0
    cld
    repe cmpsb
    jnz next;暂时不匹配
    ;匹配
    inc num
next:
    pop cx
    inc offsetnum
    loop find

    cmp num,0
    jnz a0
nomatch:
    lea dx,mess5
    mov ah,09
    int 21h
    jmp exit
a0:
    mov bx,num
    call trans

    jmp exit

;----------------------------------------------
trans proc near
    mov ch,4

rotate:
    mov cl,4
    rol bx,cl
    mov al,bl
    and al,0fh
    add al,30h
    cmp al,3ah
    jl print
    add al,7h;'A-Z'

print:
    mov dl,al
    mov ah,02h
    int 21h
    dec ch
    jnz rotate
ret
```

```
trans endp
;--------------------------------------------
;---------------------------------------------------------------
;回车换行
crlf proc near
    mov dl,0dh
    mov ah,02h
    int 21h
    mov dl,0ah
    mov ah,02h
    int 21h
ret
crlf endp
;--------------------------------------------------------------
exit:
    ret
main endp
codeseg ends
    end start
```

12.成绩输入并按从高到低排序

```
;接收不限次数的成绩输入，成绩格式为两位10进制数。不输入即回车时表示输入结束，此时将已输入的所有成绩按降序序列输出，且换行后
;以16进制形式输出已输入的成绩的个数
dataseg segment
    score dw 50 dup(?)
    num dw 0
    mess1 db 'input score:','$'
    flag dw 1
dataseg ends

codeseg segment
    assume cs:codeseg, ds:dataseg, es: dataseg
main proc far
start:
    push ds
    sub ax,ax
    push ax
    mov ax,dataseg
    mov ds,ax
    mov es,ax
;main begin
a0:
    lea dx,mess1
    mov ah,09h
    int 21h
    call input_score
    call crlf
    cmp flag,0
    jz next
    mov di,num
    add di,di
    mov score[di],bx
    inc num
    jmp a0
;输入完毕
```

```
next:
    mov di,num
    add di,di
    mov score[di],bx
    inc num;加上最后一个
    call maxtomin;降序排列
    mov cx,num
    mov si,0
show:
    mov bx,score[si]
    push cx
    call binidec
    call crlf
    pop cx
    add si,2
    loop show
;输出num
    call crlf
    mov bx,num
    call trans
;    call crlf
;    mov si,0
;    mov bx,score[si]
;    call binidec
    jmp exit
;----------------------------------------------------
input_score proc near
    mov bx,0
newchar:
    mov ah,1
    int 21h
    cmp al,0dh
    jz endinput
    sub al,30h
    jl exit1;小于0退出
    cmp al,9
    jg exit1;大于9退出
    cbw

    xchg ax,bx
    mov cx,10
    mul cx
    xchg ax,bx
    add bx,ax

    jmp newchar
endinput:
    mov flag,0
exit1:
    ret
input_score endp
;-----------------------------------------------
crlf proc near
    mov dl,0dh
    mov ah,02h
    int 21h
    mov dl,0ah
    mov ah,02h
    int 21h
ret
crlf endp
;----------------------------------------------
;---------------------------------------------------------
maxtomin proc near
    mov cx,num;数组中有10个数
    dec cx;外循环9次，完成每次外循环后把最小的数沉底，放到最后
loop1:
    mov di,cx;外循环次数暂存到di里
    mov bx,0
loop2:
    mov ax,score[bx]
    cmp ax,score[bx+2]
    jge continue;如果前面的数大，不交换
    xchg ax,score[bx+2]
    mov score[bx],ax
continue:
    add bx,2
    loop loop2
    mov cx,di;内循环次数等于cx，内循环用完cx后，恢复外循环的cx
    loop loop1
ret
maxtomin endp
;-------------------------------------------------------------
;-----------------------------------
binidec proc near
;    mov cx,10000d
;    call dec_div
;    mov cx,1000d
;    call dec_div
    mov cx,100d
    call dec_div
    mov cx,10d
    call dec_div
    mov cx,1d
    call dec_div
    ret
binidec endp
;-------------------------------------------
dec_div proc near
    mov ax,bx
    mov dx,0
    div cx
    mov bx,dx
    mov dl,al
    add dl,30h
    mov ah,2
    int 21h
    ret
dec_div endp
;------------------------------------------
trans proc near
    mov ch,4

rotate:
    mov cl,4
    rol bx,cl
    mov al,bl
    and al,0fh
    add al,30h
    cmp al,3ah
    jl print
    add al,7h

print:
    mov dl,al
    mov ah,02h
    int 21h
    dec ch
    jnz rotate
ret
trans endp
;-----------------------------------
exit:
    ret
main endp
codeseg ends
    end start
```

13.逻辑尺

```
dataseg segment
    x dw 1,2,3,4,5,6,7,8,9,10
    y dw 2,3,4,5,6,7,8,9,10,11
    z dw 10 dup(?)
    logic_rule dw 00dch
dataseg ends
;
codeseg segment
    assume cs:codeseg, ds:dataseg, es:dataseg
main proc far
start:
    push ds
    sub ax,ax
    push ax
    mov ax,dataseg
    mov ds,ax
    mov es,ax
;main begin
    mov bx,0
    mov cx,10
    mov dx,logic_rule
next:
    mov ax,x[bx]
    shr dx,1
    jc substract
    add ax,y[bx]
    jmp result
substract:
    sub ax,y[bx]
result:
    mov z[bx],ax
    add bx,2
    loop next
```

```
show:
    lea si,z
    mov bx,[si+2]
    call tt
    jmp exit
    ;------------------------
;把bx的二进制形式打印出来
tt proc near
    mov ch,16
rr:
    mov cl,1
    rol bx,cl
    mov al,bl
    and al,01h
    add al,30h
    mov dl,al
    mov ah,02h
    int 21h
    dec ch
    jnz rr
ret
tt endp
;----------------------------
exit:
    ret
main endp
codeseg ends
    end start
```

14.search3

```
dataseg segment
    mess1 db 'Enter keyword:','$'
    mess2 db 'Enter Sentence:','$'
    mess3 db 'Match at location:','$'
    mess4 db 'H of the sentence.','$'
    mess5 db 'No match.','$'
    offsetnum dw 0
;
keyin label byte
    maxk db 10
    actk db ?
    key db 10 dup(?)
;
senin label byte
    maxs db 50
    acts db ?
    sen db 50 dup(?)
;

dataseg ends

codeseg segment
    assume cs:codeseg, ds:dataseg, es: dataseg
main proc far
```

```
start:
    push ds
    sub ax,ax
    push ax
    mov ax,dataseg
    mov ds,ax
    mov es,ax
;main begin
    lea dx,mess1;输出Enter keyword:
    mov ah,09h
    int 21h
    lea dx,keyin;读入关键字
    mov ah,0ah
    int 21h
    call crlf
;
a00:
    lea dx,mess2;输出Enter Sentence:
    mov ah,09h
    int 21h
    lea dx,senin;读入句子
    mov ah,0ah
    int 21h
    call crlf
;

;若输入句子是ctrl+C则退出
    mov di,0
    mov al,sen[di]
    cmp al,3;一定要和低字节，否则加上高字节就不是3了
    jz exit
;

    mov cl,acts
    mov ch,0
    sub cl,actk
    inc cx;cx里是外总比较次数
find:
    lea si,key
    lea di,sen
    add di,offsetnum
    push cx
    mov cl,actk;内比较次数
    mov ch,0
    cld
    repe cmpsb
    jz match;匹配
    pop cx
    inc offsetnum
    loop find
nomatch:
    lea dx,mess5
    mov ah,09
    int 21h
    jmp exit

match:
    pop cx;把循环里欠下的pop补上
    lea bx,sen
    sub di,bx
    mov al,actk
    mov ah,0
    sub di,ax
    mov bx,di
    call trans
    lea dx,mess4
    mov ah,09h
    int 21h
    jmp exit

;----------------------------------------------
trans proc near
    mov ch,4

rotate:
    mov cl,4
    rol bx,cl
    mov al,bl
    and al,0fh
    add al,30h
    cmp al,3ah
    jl print
    add al,7h;'A-Z'

print:
    mov dl,al
    mov ah,02h
    int 21h
    dec ch
    jnz rotate
ret
trans endp
;--------------------------------------------
;---------------------------------------------------------------
;回车换行
crlf proc near
    mov dl,0dh
    mov ah,02h
    int 21h
    mov dl,0ah
    mov ah,02h
    int 21h
ret
crlf endp
;--------------------------------------------------------------
exit:
    ret
main endp
codeseg ends
    end start
```

15.查找匹配字符并删除

```
dataseg segment
    mess1 db 'Enter keyword:','$'
    mess2 db 'Enter Sentence:','$'
    keychar db ?
;
senin label byte
    maxs db 50
    acts db ?
    sen db 50 dup(?)
;
dataseg ends

codeseg segment
    assume cs:codeseg, ds:dataseg, es: dataseg
main proc far
start:
    push ds
    sub ax,ax
    push ax
    mov ax,dataseg
    mov ds,ax
    mov es,ax
;main begin
    lea dx,mess1;输出Enter keyword:
    mov ah,09h
    int 21h
    ;读入关键字
    mov ah,01h
    int 21h
    mov keychar,al
    call crlf
;
a00:
    lea dx,mess2;输出Enter Sentence:
    mov ah,09h
    int 21h
    lea dx,senin;读入句子
    mov ah,0ah
    int 21h
    call crlf
;
    mov cl,acts
    mov ch,0
    mov si,0
goon:
    mov al,keychar
    cmp sen[si],al
    jnz next
    call delete;如果匹配，删除这个字符
    dec acts
next:
    inc si
    loop goon

show:
    mov cl,acts
    mov ch,0
    mov si,0
a1:
    mov dl,sen[si]
    mov ah,02
    int 21h
    inc si
    loop a1

    jmp exit

;---------------------------------------------------------------
;回车换行
crlf proc near
    mov dl,0dh
    mov ah,02h
    int 21h
    mov dl,0ah
    mov ah,02h
    int 21h
ret
crlf endp
;--------------------------------------------------------------
;----------------------------------------------
trans proc near
    mov ch,4

rotate:
    mov cl,4
    rol bx,cl
    mov al,bl
    and al,0fh
    add al,30h
    cmp al,3ah
    jl print
    add al,7h;'A-Z'

print:
    mov dl,al
    mov ah,02h
    int 21h
    dec ch
    jnz rotate
ret
trans endp
;--------------------------------------------
delete proc near

    push bx
    lea bx,sen
    push si
    mov di,si
    inc si
    add di,bx
    add si,bx
    cld
    push cx
    rep movsb
    pop cx
    pop si
    pop bx
    dec si;从当前字符开始比
    inc cx;从当前字符开始比

    ret
delete endp
;-----------------------------------------------
exit:
    ret
main endp
codeseg ends
    end start
```

16.找到最小偶数

```
dataseg segment
    data dw 3,5,7,14,2,6
dataseg ends

codeseg segment
    assume cs:codeseg, ds:dataseg, es: dataseg
main proc far
start:
    push ds
    sub ax,ax
    push ax
    mov ax,dataseg
    mov ds,ax
    mov es,ax

    mov ax,0
    mov bx,0
    mov cx,6
;找到数组中第一个偶数
findeven:
    mov ax,data[bx]
    add bx,2
    test ax,01h
    jnz findeven

    mov bx,0
;找到最小偶数放在AX中
compare:
    mov dx,data[bx]
    add bx,2
    test dx,01h
    jnz loop1
    cmp ax,dx
    jle loop1
    mov ax,dx

loop1:
    loop compare
;显示最小偶数，缺点是必须是0-9中的数才能正确显示
```

```
show:
    add ax,30h
    mov dl,al
    mov ah,2
    int 21h

ret
main endp
codeseg ends
    end start
```

17.人名排序

```
;课本231页
        .model  small
        .stack  40h
        .data
namepar label byte
    maxnlen db 21
    namelen db ?
    namefld db 21 dup(?)

    crlf db 13,10,'$'
    endaddr dw ?
    messg1 db 'Name?','$'
    messg2 db 'Sorted names:',13,10,'$'
    namectr db 0;输入的名字个数
    nametab db 30 dup(20 dup(' '))
    namesav db 20 dup(?),13,10,'$'
    swapped db 0
;
        .code
begin proc far
    mov ax,@data
    mov ds,ax
    mov es,ax

    cld
    lea di,nametab
a20loop:
    call b10read
    cmp namelen,0
    jz a30;结束输入，去排序
    cmp namectr,30
    je a30;30个表项满了，去排序
    call d10stor;存储名字到表中
    jmp a20loop
a30:
    cmp namectr,1
    jbe a40;不用排序
    call g10sort;排序
    call k10disp;打印
a40:
    mov ax,4c00h
    int 21h
begin endp
;--------------------------------------------
```

```
;接收名字输入到缓冲区
b10read proc near
    mov ah,09
    lea dx,messg1
    int 21h
    mov ah,0ah
    lea dx,namepar
    int 21h
    mov ah,09
    lea dx,crlf
    int 21h
;不足的补空格
    mov bh,0
    mov bl,namelen
    mov cx,21
    sub cx,bx
b20:
    mov namefld[bx],20h
    inc bx
    loop b20
    ret
b10read endp
;------------------------------------------------
;存储名字到表中
d10stor proc near
    inc namectr
    cld
    lea si,namefld
    mov cx,10
    rep movsw
    ret
d10stor endp
;---------------------------------------------------
;名字排序
g10sort proc near
    sub di,40
    mov endaddr,di;设置停止标志
g20:
    mov swapped,0
    lea si,nametab
g30:
    mov cx,20;比较长度
    mov di,si
    add di,20;下一个名字
    mov ax,di
    mov bx,si
    repe cmpsb
    jbe g40;不交换
    call h10xchg
g40:
    mov si,ax
    cmp si,endaddr;到表尾结束了吗?
    jbe g30;没结束
    cmp swapped,0;有交换吗?
    jnz g20;有交换
    ret
g10sort endp
;-------------------------------------------
```

```
;交换表中的元素
h10xchg proc near
    mov cx,10
    lea di,namesav
    mov si,bx
    rep movsw;把前一个元素放到缓存
;
    mov cx,10
    mov di,bx
    rep movsw;把后一个数放到前一个数中
;
    mov cx,10
    lea si,namesav
    rep movsw;把缓存中的数放到后一个数中
    mov swapped,1;设置交换标志位
    ret
h10xchg endp
;-------------------------------------------
;显示输入的人名
k10disp proc near
    mov ah,09h
    lea dx,messg2
    int 21h
    lea si,nametab
k20:
    lea di,namesav
    mov cx,10
    rep movsw
    mov ah,9
    lea dx,namesav
    int 21h
    dec namectr
    jnz k20
    ret
k10disp endp
;--------------------------------------------
end begin
```

18.查询表格

```

datasg      segment para    'data'
messl       db      'stock number?',13,10,'$'
;
stoknin     label   byte
  max       db      3
  act       db      ?
  stokn     db      3 dup(?)
;
stoktab     db      '05','  Excavators'
            db      '08','  Lifters   '
            db      '09','  Presses   '
            db      '12','  Valves    '
            db      '23','  Processors'
            db      '27','  Pumps     '
;
descrn      db      14 dup(20h),13,10,'$'
```

```
mess        db      'Not in table! ','$'

datasg ends
;

codesg      segment para 'code'
            assume  cs:codesg,ds:datasg,es:datasg
;


main        proc    far
            push    ds
            sub     ax,ax
            push    ax

            mov     ax,datasg
            mov     ds,ax
            mov     es,ax
;MAIN PART OF PROGRAM GOES HERE

start:
            lea     dx,messl
            mov     ah,09
            int     21h
            lea     dx,stoknin
            mov     ah,0ah
            int     21h
            cmp     act,0
            je      exit
            mov     al,stokn
            mov     ah,stokn+1
            mov     cx,06
            lea     si,stoktab

a20:
            cmp     ax,word ptr[si];比较两个字节，一个字
            je      a30
            add     si,14
            loop    a20
            lea     dx,mess
            mov     ah,09
            int     21h
            jmp     exit
a30:
            mov     cx,14
            lea     di,descrn
            rep     movsb
;
            lea     dx,descrn
            mov     ah,09
            int     21h
            jmp     start
exit:
            ret
main        endp
;
codesg      ends
```

```
;
            end     main
```