考点✔ 作者：2015 计机 wx

基础概念（不考简答）

- 中断向量（用到的寄存器，中断向量表的概念）
  - ✔中断向量表就是各类型中断处理程序的入口地址表
  - ✔响应中断过程
    - 取中断类型号N
    - 当前FLAGS、CS、IP 的内容入栈
    - 清 IF, TF 标志位为0
    - 取内存单元（0: N*4）字内容送IP
    - 取内存单元（0: N*4+2）字内容送CS
    - 转中断子程序

    此时 CS:IP 指向中断程序入口，开始执行中断程序。
- 调用程序和子程序之间参数传递的不同方式
- 外部设备输入输出方式
  - DMA; 中断传送方式；程序查询方式
- 各种寻址方式 - 第二章知识点
- 串操作
- ret返回指令 - call 与中断
- 标志位，各种指令对于标志位的变化
- 第一章，补码，结合第二章考察
- 实验操作基本步骤，masm，link，debug

指令禁忌 ⛔

读程序 - 实验示例

- 串操作 scasb, movsb
- 明确串操作各个寄存器里的内容

程序改错 🤡

程序填空 - 作业、实验

- 把一个数组中的最大值和最小值最终存放在AH和AL中

完整编程 - 作业、实验

- 屏幕提示字符串“input char:", 然后输入单个字符，之后显示该字符的下一个字符
- 写子程序，完成功能：十六进制显示BX寄存器内容
- 对100个元素的带符号字数组A从小到大排序
- 写一个宏汇编和子程序分别实现输出二进制数字（参考二进制转十六进制）
- 编写完整程序，输出字母 a~z
- 写子程序，输出bx寄存器中的二进制数

- 宏汇编，求一个双字长数组的所有元素之和

寻址方式

```
; 数1的程序
; 在 ADDR 单元中存放着数Y的地址，试编制一程序把Y中1的个数存入COUNT单元中
datarea segment
  v1   db  1,2,3
  v2   db 'adfad'
  v3   dw 'a','b'
  addr  dw number
  number dw 11111111b
  count  dw ?
datarea ends
;
prognam segment
main proc far
  assume cs:prognam,ds:datarea
start:
  push ds
  sub  ax,ax
  push ax
  mov  ax,datarea
  mov  ds,ax
  mov  cx,0
  mov  bx,addr ; mov bx,[0000h] <=> mov bx,0000h
  mov  bx,[0002h]
  mov  bx,word ptr v1  ; warning -- match
  mov  bx,offset number
  mov  ax,2[bx] ; mov ax,[addr] <=> mov ax, addr
repeat:
  test ax,0ffffh ; test Y
  jz   exit      ; if Y=0, get exit
  jns  shift     ; if MSB=0, C unchanged
  inc  cx
shift:
  shl  ax,1
  jmp  repeat
exit:
  mov  count,cx
  ret
main endp
prognam ends
  end start
```

DEBUG - 常用指令

统计

& 分类统计字符个数

```
; 实验2.3 分类统计字符个数
datarea segment
      letter db ?
      digit  db ?
      other  db ?
      string label byte
             max db 80
             act db ?
             str db 80 dup(?)
      mess0 db 13,10,'please enter the string:','$'
      mess1 db 13,10,'num_letter:','$'
      mess2 db 13,10,'num_digit :','$'
      mess3 db 13,10,'num_others:','$'
datarea ends
prognam segment
      assume cs:prognam,ds:datarea
start:
      push ds
      sub  ax,ax
      push ax
      mov  ax,datarea
      mov  ds,ax
      mov  es,ax
; 置零
      mov  letter,0
      mov  digit,0
      mov  other,0
; enter the string
      lea  dx,mess0
      mov  ah,09H
      int  21H
; read the input string
      lea  dx,string
      mov  ah,0aH
      int  21H
;
      sub  ch,ch          ; ch 置零
      mov  cl,[string+1]  ; 设置循环次数为 string 的长度+1
      lea  si,string+2    ; 设置开始索引值为 string+2
digitseg:
      mov  al,[si]        ; 把索引值为 si 的字符送入 al
      cmp  al,'0'
      jb   otherseg       ; 小于 0 为其他，  ASCII 30h-39h
      cmp  al,'9'
      ja   letter1seg     ; 大于 9 为 字母或其他（ja：无符号大于则跳转）
      inc  digit          ; 结果为数字 digit1++
      jmp  loop1
letter1seg:
      cmp  al,'A'          ; 在 'A'-'Z' 之间为字母
      jb   otherseg        ; 小于 'A' 的为其他

      cmp  al,'Z'
```

```
      ja   letter2seg      ; 大于 'Z' 的可能为字母或其他
      inc  letter          ; 是字母，则 letter++
      jmp  loop1
letter2seg:
      cmp  al,'a'
      jb   otherseg        ; 小于 'a'(且大于'Z')的为其他
      cmp  al,'z'
      ja   otherseg        ; 大于 'z' 的为其他
      inc  letter          ; 是字母，则 letter++
      jmp  loop1
otherseg:
      inc  other           ; 是其他字符，other++
loop1:
      inc  si              ; 字符的索引值加一
      dec  cl              ; 手动减一 loop计数
      cmp  cl,0            ; 如果 cl = 0 则 loop 结束
      jz   print           ; 打印输出
      jne  digitseg        ; 重复分类过程
print:
;     输出字母的个数
      lea  dx,mess1
      mov  ah,09H
      int  21H
      mov  al,letter
      call disp
;     输出数字的个数
      lea  dx,mess2
      mov  ah,09H
      int  21H
      mov  al,digit
      call disp
;     输出其他字符的个数
      lea  dx,mess3
      mov  ah,09H
      int  21H
      mov  al,other
      call disp
exit:
      mov  ah,04ch    ; terminate current process
      int  21H        ; invoke the interrupt
disp:
;     输出十进制的数字(最多两位，由于输入字符最多为80个的限制)
;     原先存储在 al 中
      mov  ah,0       ; 高位置零
      mov  bl,10      ; 除数 10
      div  bl         ; div 无符号：div src 16位操作：商ax=(dx,ax)/src,余数dx
      add  al,30h     ; 结果商放在 al 中，+30h 表示该数字的 ASCII 码
      mov  dl,al      ; 将结果的十进制的 ASCII 码存入 dl 以备显示
      mov  bh,ah      ; 结果余数放在 ah 中，
      mov  ah,02h     ; 显示输出 dx，则显示 dl 中的数据
      int  21H        ; invoke the interrupt
      mov  al,bh      ; 将余数放入 al 中

      add  al,30H     ; +30h 得到余数的十进制的 ASCII 码
```

```
      mov  dl,al      ; 存入 dl 以备显示
      mov  ah,02h     ; 显示输出 dx, 则显示 dl 中的数据
      int  21h        ; invoke the interrupt
      ret
prognam  ENDS
     end start
```

数1的程序

```
; 数1的程序
; 在 ADDR 单元中存放着数Y的地址，试编制一程序把Y中1的个数存入COUNT单元中
datarea segment
  addr dw number
  number dw 11111111b
  count  dw ?
datarea ends
;
prognam segment
main proc far
  assume cs:prognam,ds:datarea
start:
  push ds
  sub  ax,ax
  push ax
  mov  ax,datarea
  mov  ds,ax
  mov  cx,0
  mov  bx,addr
  mov  ax,[bx] ; how about ax,[addr]
repeat:
  test ax,0ffffh ;test Y
  jz   exit      ; if Y=0, get exit
  jns  shift     ; if MSB=0, C unchanged
  inc  cx
shift:
  shl  ax,1
  jmp  repeat
exit:
  mov  count,cx
  ret
main endp
prognam ends
  end start
```

统计学生成绩

```
; 实验2.3 统计学生成绩 result
; ***************************************************************
datarea segment ;define data segment

    grade   dw  56,69,84,82,73,88,99,63,100,80
```

```
    s5      dw  0
    s6      dw  0
    s7      dw  0
    s8      dw  0
    s9      dw  0
    s10     dw  0
datarea ends
;*************************************************************************
prognam segment ;define code segment
;-------------------------------------------------------------------------
main    proc   far
        assume cs: prognam, ds: datarea
start:
;set up stack for return
        push   ds
        sub    ax,ax
        push   ax
;set DS register to current data segment
        mov    ax,datarea
        mov    ds,ax
; main part of prognam goes here
        mov    s5,0      ; initialize counter
        mov    s6,0
        mov    s7,0
        mov    s8,0
        mov    s9,0
        mov    s10,0
        mov    cx,10    ; initialize loop count value
        mov    bx,offset grade   ;initialize first addr
compare:
    mov    ax,[bx]
        cmp    ax,60      ; < 60
        jl     five
        cmp    ax,70      ; 60 - 70
        jl     six
        cmp    ax,80      ; 70 - 80
        jl     seven
        cmp    ax,90      ; 80 - 90
        jl     eight
        cmp    ax,100     ; 90 - 100
        jne    nine
        inc    s10        ; =100 s10++
        jmp    change_addr
nine:
        inc    s9         ; s9++
       jmp    change_addr
eight:
        inc    s8         ; s8++
       jmp    short change_addr
seven:

        inc    s7         ; s7++
```

```
        jmp     short change_addr
six:
         inc     s6          ; s6++
        jmp     short change_addr
five:
         inc     s5          ; s5++
        jmp     short change_addr
change_addr:
        add     bx,2
         loop    compare
         ret                 ; return to DOS
main    endp
;--------------------------------------------------------------------
prognam ends
;********************************************************************
         end     start      ; end assembly
```

输入字符串，对非数字计数

```
; 5.11 从键盘输入一系列以 $ 为结束符的字符串，然后对其中的非数字字符计数，并显示出计数结果
datarea segment
    digit  db  ?
    string label byte
           max db 80
           act db ?
           str db 80 dup(?)
datarea ends
;************************************************************************
prognam segment
;------------------------------------------------------------------------------
main proc far
    assume cs: prognam, ds: datarea
start:
    push ds
    sub  ax,ax
    push ax
;
    mov  ax,datarea
    mov  ds,ax
    mov  es,ax
;
    push bx
    push dx
    push si
;
; input string
    lea  dx,string
    mov  ah,0ah
    int  21h
    lea  si,string+2   ; 索引值
    mov  digit,0          ; 计数

startcount:
```

```
    mov  al,[si]
    cmp  al,'0'
    jb   nextloop
    cmp  al,'9'
    ja   nextloop
    inc  digit
nextloop:
    cmp  al,'$'
    je   disp
    inc  si
    jmp  startcount
disp:
    mov  al,digit
    mov  ah,0
    sub  si,ax      ; si-ax 得到非数字的字符个数
    mov  ax,si
    mov  ah,0       ; 高位置零
    mov  bl,10      ; 除数 10
    div  bl         ; div 无符号：div src 16位操作：商ax=(dx,ax)/src,余数dx
    add  al,30h     ; 结果商放在 al 中，+30h 表示该数字的 ASCII 码
    mov  dl,al      ; 将结果的十进制的 ASCII 码存入 dl 以备显示
    mov  bh,ah      ; 结果余数放在 ah 中，
    mov  ah,02h     ; 显示输出 dx，则显示 dl 中的数据
    int  21H        ; invoke the interrupt
    mov  al,bh      ; 将余数放入 al 中
    add  al,30H     ; +30h 得到余数的十进制的 ASCII 码
    mov  dl,al      ; 存入 dl 以备显示
    mov  ah,02h     ; 显示输出 dx，则显示 dl 中的数据
    int  21h        ; invoke the interrupt
;
    pop  si
    pop  dx
    pop  bx
    pop  ax
    ret
    mov  ah,4ch
    int  21h
main endp
;-------------------------------------------------------------------------
prognam ends
;*************************************************************************
     end  start
```

发声🔊

& 枪声 🔫

```
prognam segment
main  proc  far

    assume  cs:prognam
```

```
    org     100h  ; start of program
start:
    mov   cx,50d   ;set number of shots
new_shot:
    push  cx       ; save count
    call  shoot    ; sound of shot
    mov   cx,4000h ; set up silent delay
silent:
    loop  silent   ; silent delay
    pop   cx       ; get shots count back
    loop  new_shot ; loop till shots done
    mov   al,48h   ; 01001000
    out   61h,al   ; reset output port
    int   20h      ; return to DOS
main  endp
shoot   proc  near
    mov   dx,1400h  ; initialize value of wait
    mov   bx,20h   ; set count
    in    al,61h   ; get port 61
    and   al,11111100b ;AND off bits 0,1
sound:
    xor   al,2     ; toggle bit #1 in AL
    out   61h,al   ; output to port 61
    add   dx,9248h ; add random pattern
    mov   cl,3     ; set to rotate 3 bits
    ror   dx,cl    ; rotate it
    mov   cx,dx    ; put in CX
    and   cx,1ffh  ; mask off upper 7 bits
    or    cx,10    ; ensure not too short
wait1:
    loop  wait1     ; wait
    dec   bx       ; done enough?
    jnz   sound
; turn off sound
    and   al,11111100b ; AND off bits 0,1
    out   61h,al       ; turn off bits 0,1
    ret
shoot  endp
prognam ends
    end start
```

乐曲程序🎼

```
;*************************************************************************************
datasg segment
       mus_freq   dw 262,294,330,262,262,294,330,262      ;频率决定音高
                  dw 330,349,392,330,349,392,392,440
                  dw 392,349,330,262,392,440,392,349
                  dw 330,262,294,196,262,294,196,262
       mus_time   dw 25,25,25,25,25,25,25,25,25,25
                  dw 50,25,25,50,12,12,12,12,25,25
                  dw 12,12,12,12,25,25,25,25,50,25,25,50
    mes    db 31h,32h,33h,31h,31h,32h,33h,31h,33h,34h,35h,33h,34h,35h,35h,36h
```

```
      db 35h,34h,33h,31h,35h,36h,35h,34h,33h,31h,32h,35h,31h,32h,35h,31h,'$'

datasg ends
;*********************************************************************************
;*********************************************************************************
codesg segment
    main    proc   far
         assume cs:codesg,ds:datasg
   org    100h     ;链接器将之后的程序放在0100h的开始位置
;-----------------------------------------------------
 start:
   push ds
   sub  ax,ax
   push ax
   mov  ax,datasg
   mov  ds,ax
;
   lea  di,mus_freq   ;频率
   lea  bx,mus_time   ;节拍
   lea  si,mes    ;音符
   mov  cx,32d;
;-----------------------------------------------------
 new_shot:
    push cx
    call sound
    add  di,2
    add  bx,2

      push ax
      mov  ax,[bx]
      push bx
      mov  bx,ax
      mov  ax,80d
      mul  bx
      pop  bx
      mov  [bx],ax
      pop ax

   add  si,1
   mov  cx,0ffffh
;-----------------------------------------------------
 silent:
   loop silent
   pop  cx
   loop new_shot
   mov  al,48h
   out  61h,al

   mov  ah,4ch
   int  21h

   ret

 main endp
```

```
;-------------------------------------------------
sound proc near

   mov  ah,02h    ;利用中断21h的02功能调用，显示音符简谱
   and  si,00ffh
   mov  dx,[si]
   int  21h

   in  al,61h
   and  al,11111100b
  sing:
      xor  al,2    ;将PB1置1，打开扬声器
   out  61h,al

   push ax
   call widnth
   pop  ax
   mov  cx,dx
  waits:
      loop waits    ;循环延迟，由此控制开关电路所产生脉冲的频率
      dec WORD PTR [bx]  ;实现相应节拍的声音
   jnz  sing


      and  al,11111100b  ;关闭扬声器
   out  61h,al
   ret       ;一轮实现了一个音的输出

sound endp
;-------------------------------------------------
;-------------------------------------------------
widnth proc near     ;控制脉宽的计数值
   mov  ax,2801
   push bx     ;保存的是节拍
   mov  bx,50
   mul  bx
   div  WORD PTR [di]  ;除以了相应的频率
   mov  dx,ax
   pop  bx
   ret
widnth endp
;-------------------------------------------------
codesg ends
;*********************************************************************************
 end start
;*********************************************************************************
```

乐曲程序-示例

```
; 实验3.1 乐曲程序
datasg segment
; 音乐频率
```

```
      mus_freq  dw 262,294,330,262,262,294,330,262
                dw 330,349,392,330,349,392,392,440
                dw 392,349,330,262,392,440,392,349
                dw 330,262,294,296,262,294,196,262
; 音乐节拍 (2节拍 = 50; 1节拍等于25; 1/2节拍等于12)
      mus_beat  dw 25,25,25,25,25,25,25,25,25,25
                dw 50,25,25,50,12,12,12,12,25,25
                dw 12,12,12,12,25,25,25,25,50,25,25,50
datasg ends
;-------------------------------------------------------------
codesg segment
main proc far
  assume cs:codesg,ds:datasg
  org    100h
start:
; save old data segment
    push  ds
    sub   ax,ax
    push  ax
; set ds register to current data segment
    mov   ax,datasg
    mov   ds,ax
; set specific data address
    lea   di,mus_freq ;频率
    lea   si,mus_beat ;节拍
; loop 32 times
    mov   cx,32d      ;音符
new_shot:
    push  cx
    call  sound       ; make sound
    add   di,2        ; next frequency
    add   si,2        ; next beat
;节拍乘80
    push ax
    mov  bx,[si]
    mov  ax,80d
    mul  bx
    mov  [si],ax
    pop  ax
; set silent loop times
    mov   cx,0ffffh  ; silent loop
silent:
    loop  silent
    pop   cx
; finish silent loop
    loop  new_shot
; finish new_shot loop
    mov   al,48h
    out   61h,al  ;恢复61h端口
    mov   ah,4ch
    int   21h
    ret

main endp
```

```
; sound 子程序
sound proc near
    in   al,61h
    and  al,11111100b ; 关断定时器通道2的门控
sing:
    xor  al,2         ;触发61h端口第一位
    out  61h,al
    push ax
    call widnth
    pop  ax
    mov  cx,dx
waits:
    loop waits
; loop 控制开关电路所产生脉冲的频率
    dec  WORD PTR [si]
; loop 次数为节拍数*80, 表示脉冲持续的时间
    jnz  sing
;
    and  al,11111100b
    out  61h,al      ; 恢复61端口
    ret
sound endp
; 将音符的频率转化为控制脉冲宽度的计数值
; dx = 1/(2*freq)
widnth proc near
    mov  ax,2801
    push bx
    mov  bx,50
    mul  bx
    div  WORD PTR [di]
    mov  dx,ax  ; dx = 2801*50 / freq
    pop  bx
    ret
widnth endp
codesg ends
    end start
```

显示📺

表格显示ASCII码

```
; 用15*16的表格显示ascii码
datarea segment
  recol db 13,10,'$' ;换行
  space db '$'
datarea ends

prognam segment
main proc far
  assume cs:prognam,ds:datarea,es:datarea
start:

  push ds
```

```
  sub  ax,ax
  push ax
  mov  ax,datarea
  mov  ds,ax
  ;
  mov  dl,10h
  mov  cx,15
outside:
  push cx
  mov  cx,16
inside:
  mov  ah,02h
  int  21h
  inc  dl
  push dx
  lea  dx,space
  mov  ah,09h
  int  21h
  pop  dx
  loop inside
  push dx
  lea  dx,recol
  mov  ah,09h
  int  21h
  pop  dx
  pop  cx
  loop outside
  ret
main endp
prognam ends
  end start
```

串指令，movsb

```
;把数据段的字符串传送到附加段
datarea segment
  mess1 db 'personal computer $'
datarea ends
;
extra segment
  mess2 db 17 dup(?)
extra ends
;
code segment
  assume cs:code,ds:datarea,es:extra
main proc far
start:
  push ds
  sub  ax,ax
  push ax
  mov  ax,datarea
  mov  ds,ax
  mov  ax,extra
```

```
  mov  es,ax
  lea  si,mess1
  lea  di,mess2
  mov  cx,17
  cld
  rep movsb
  ret
main endp
code ends
  end start
```

进制转换🔃

二进制转十六进制

```
; 二进制到十六进制的转换程序
prognam segment
main proc far
  assume cs:prognam
start:
  push ds
  sub  ax,ax
  push ax
  mov  ch,4
rotate:
  mov  cl,4
  rol  bx,cl
  mov  al,bl
  and  al,0fh ; mask off left digit, now we have the lowest four byte in al
  add  al,30h ; convert to ascii
  cmp  al,3ah ; is it > 9 ?
  jl   printit
  add  al,7h  ; digit is A to F
printit:
  mov  dl,al
  mov  ah,2
  int  21H
  dec  ch
  jnz  rotate
  ret
main endp
prognam ends
  end
```

十进制转十六进制

```
decihex SEGMENT
  assume cs:decihex

main proc FAR
```

```
repeat:  call decibin ; keyboard to binary
         call crlf
         call binihex
         call crlf
         jmp  repeat
main endp
;
decibin   proc near
    mov bx,0
newchar:
    mov ah,1
    int 21h
    sub al,30h
    jl  exit
    cmp al,9d
    jg  exit
    cbw
    xchg ax,bx
    mov  cx,10d
    mul  cx
    xchg ax,bx
    add  bx,ax
    jmp  newchar
exit:
    ret
decibin  endp
;
binihex  proc near
    mov  ch,4
rotate:
    mov  cl,4
    rol  bx,cl
    mov  al,bl
    and  al,0fh   ; mask off left digit
    add  al,30H
    cmp  al,3ah   ; is it > 9 ?
    jl   printit
    add  al,7h
printit:
    mov  dl,al    ; put ascii char in dl
    mov  ah,2
    int  21H
    dec  ch
    jnz  rotate
    ret
binihex   endp
;
crlf  proc  near
    mov  dl,0DH
    mov  ah,2
    int  21h
    mov  dl,0ah

    mov  ah,2
```

```
    int  21h
    ret
crlf  endp
;
decihex  ends
    end main
```

十六进制转二进制

```
; 5.9 从键盘接收一个四位的十六进制数，并在终端上显示与它等值的二进制数
datarea segment
      str db 0dh,0ah,'$'
datarea ends
;*****************************************************************
prognam segment
;-----------------------------------------------------------------
main proc far
    assume cs: prognam, ds:datarea
start:
    push ds
    sub  ax,ax
    push ax
;
    mov  ax,datarea
    mov  ds,ax
;
    mov  ch,4  ; loop 4次 4位16进制数
    mov  cl,4  ; 移位次数
    mov  bx,0
input:
    shl  bx,cl  ; 左移四位
    mov  ah,01h ; 输入0~9, a~f, A~F的数据 存入al
    int  21h
    cmp  al,'0'
    jb   input  ; 不可以小于 0
    cmp  al,'f'
    ja   input   ; 不可以大于 f
    cmp  al,'9'
    jbe  num     ; 输入的字符为0~9
    cmp  al,'@'
    jbe  input
    cmp  al,'F'
    jbe  Convert1  ;输入的字符为A~F
    cmp  al,'a'
    jb   input
    cmp  al,'f'
    jbe  Convert2
num:
    and  al,0fh   ;转换为: 1010B～1111B  或 0000B～1001B
    jmp  binary
```

```
Convert1:
    and  al,0fh
    add  al,9
    jmp  binary
Convert2:
    sub  al,20h  ; 小写字母转大写字母
    and  al,0fh  ;
    add  al,9
    jmp  binary
binary:
    or   bl,al   ; 将键盘输入的数进行组合
    dec  ch
    jnz  input
disp:
    mov  cx,16   ; loop 16次
    lea  dx,str
    mov  ah,09h
    int  21h
display:
    mov  dl,0
    rol  bx,1    ;循环左移
    rcl  dl,1    ;带进位循环右移
    or   dl,30H
    mov  ah,2
    int  21h
    loop display
;
    ret
main endp
;--------------------------------------------------------------
prognam ends
;**************************************************************
    end start
```

十六进制转ASCII

```
; 5.17 把 AX 中的十六进制数转换为ASCII码,并将对应的ASCII码依次存放到MEM数组的四个字节中。
datarea segment
    mem  db 4 dup(?)
datarea ends
;*****************************************************************************
prognam segment
;-----------------------------------------------------------------------------
main proc far
     assume cs: prognam, ds:datarea
start:
     push ds
     sub  ax,ax
     push ax
     mov  ax,datarea
     mov  ds,ax
     mov  ax,abdeh ;(Ax) = abdeh
     mov  si,01h
```

```
     mov  bl,al
     and  bl,0fh
     cmp  bl,0ah
     jl   number1
     add  bl,07h
number1:
     add  bl,30h
     mov  mem[0],bl
     mov  cl,04h
deal:
     cmp  si,04h
     jz   quit
     shr  ax,cl
     mov  bl,al
     and  bl,0fh
     cmp  bl,0ah
     jl   number2
     add  bl,07h
number2:
     add  bl,30H
     mov  mem[si],bl
     inc  si
     jmp  deal
quit:
     mov  ah,04ch
     int  21h
main endp
;-----------------------------------------------
prognam ends
;****************************************
     ends start
```

比较&查找👀

数组中插入一元素

```
; 将正数N插入一个已整序的字数组的正确位置。该数组的首地址和末地址分别为ARRAY_HEAD和ARRAY_END，其中所有数均为正数且已按递增的次序排列
; 数组中插入一元素程序
datarea  segment
  x           dw ?
  array_head  dw 3,5,15,23,37,49,52,65,78,99
  array_end   dw 105
  n           dw 2
datarea  ends
;
prognam  segment
main proc far
  assume cs:prognam,ds:datarea
start:
  push  ds

  sub   ax,ax
```

```
  push  ax
  mov   ax,datarea
  mov   ds,ax
  mov   ax,n
  mov   array_head-2,0ffffh ; -1 的补码
  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  short compare
insert:
  mov  array_end[si+2],ax
  ret
main  endp
prognam  ends
  end start
```

查找匹配字符串

```
;查找匹配字符串
;*******************************
datarea segment
    mess1 db 'Enter keyword:','$'
    mess2 db 'Enter Sentence:','$'
    mess3 db 'Match at location:','$'
    mess4 db 'No match!',13,10,'$ '
    mess5 db 'match.',13,10,'$ '
    mess6 db 'H of the sentence.',13,10,'$ '
;
stoknin1 label byte
    max1 db 10
    act1 db ?
    stokn1 db 10 dup(?)
;
stoknin2 label byte
    max2 db 50
    act2 db ?
    stokn2 db 50 dup(?)
datarea ends
;*********************************
prognam segment
;--------------------------------
main proc far
    assume cs:prognam ,ds:datarea,es:datarea
start:
    push ds             ; 原先的 ds 入栈
    sub ax,ax           ; ax 清零
    sub bx,bx           ; bx 清零
    push ax             ; 原先的 ax 入栈
    mov ax,datarea      ; 把数据段放入 ax
```

```
    mov ds,ax           ; 把 ax 赋给 ds
    mov es,ax           ; 把 ax 赋给 es
;MAIN PART-------------;输入关键字
    lea dx,mess1        ; lea= load effective address mess1
    mov ah,09           ; 显示字符串
    int 21h             ; call dos
    lea dx,stoknin1     ; load effective address
    mov ah,0ah          ; 输入字符串
    int 21h             ; call dos
    cmp act1,0          ; 输入为空，退出
    je exit
a10:                    ; 输入句子
    call crlf           ; 调用回车换行子程序
    lea dx,mess2        ; load effective address
    mov ah,09           ; 显示字符串
    int 21h             ; call dos
    lea dx,stoknin2     ;
    mov ah,0ah          ; 把输入存储在 dx
    int 21h             ; call dos
    cmp act2,0          ; 如果输入句子等于0的话是nmatch
    je nmatch           ; jump to nmatch if equal
    mov al,act1         ; 把 act1 存到 al
    cbw                 ; 将 AL 扩展成 AX
    mov cx,ax           ; 把 ax 赋值给 cx cx 存储关键字的长度
    push cx             ; cx 入栈
    mov al,act2         ;
    sub al,act1         ;
    js nmatch           ; 如果输入句子长度小于keyword是No match
    mov di,0            ; di = 0 代表目标句子的index
    mov si,0            ; si = 0 代表源关键字的index
    lea bx,stokn2;      ;
    inc al              ;  al ++
a20:
    mov ah,[bx+di]      ; 开始比较
    cmp ah,stokn1[si]   ; 不等则转到bx+1
    jne a30             ; bx+1
    inc si              ; si 加一
    inc di              ; di 加一
    dec cx              ; cx 减一 一个字符相等
    cmp cx,0            ; 全部相等
    je match
    jmp a20
a30:
    inc bx              ; bx 加一
    dec al              ; al 减一
    cmp al,0            ; 如果 al = 0 表示全部比较完了
    je nmatch           ; 没有匹配
    mov si,0            ;
    mov di,0
    pop cx
    push cx
    jmp a20

exit:
```

```
    call crlf
    ret
nmatch:             ;no match则输出No match
    call crlf
    lea dx,mess4
    mov ah,09
    int 21h
    jmp a10         ; jump->输入句子
match:              ; match则输出位置信息
    call crlf
    lea dx,mess3
    mov ah,09
    int 21h
    sub bx,offset stokn2
    inc bx
    call trans
    lea dx,mess6         ;
    mov ah,09
    int 21h
    jmp a10
crlf proc near           ;回车，换行
    mov dl,0dh
    mov ah,2
    int 21h
    mov dl,0ah
    mov ah,2
    int 21h
    ret
crlf   endp
trans proc near          ;转换为16进制，参考书上例6.3
    mov ch,4         ;number of digits
rotate:
    mov cl,4         ;set count to 4bits
    rol bx,cl        ;left digit to right
    mov al,bl        ;mov to al
    and al,0fh       ;mask off left digit
    add al,30h       ;convert hex to ASCII
    cmp al,3ah       ;is it>9?
    jl printit       ;jump if digit=0 to 9
    add al,7h        ;digit is A to F
printit:
    mov dl,al        ;put ASCII char in DL
    mov ah,2         ;Display Output funct
    int 21h          ;call DOS
    dec ch           ;done 4 digits?
    jnz rotate       ;not yet
    ret          ;return from trans
trans endp           ;
main endp
;---------------------------------
prognam ends
;*********************************

    end start
```

求出最小偶数,存入AX

```
; 5.7 求出首地址为 DATA 的 100D 字数组中最小偶数，并把它存放在 AX 中
datarea segment
    data dw 10,9,14,7,6,5,4,3,2,1,90 dup(63)
datarea ends
;*******************************************************
prognam segment
;-------------------------------------------------------
main proc far
    assume cs: prognam, ds: datarea
start:
;
      push ds
      sub  ax,ax
      push ax
;
      mov  ax,datarea
      mov  ds,ax
;
      push bx
      push cx
      push dx
      push si
;
      mov bx,offset data
      mov si,0                    ; data index
      mov cx,100                  ; loop 100 times
      mov ax,[bx][si]
;
checker:
      mov dx,[bx][si]
      push cx
      mov cx,dx
      rcr cx,1  ;带进位循环右移一位
      pop cx
      jc continue  ;奇数跳转
      cmp ax,dx
      jle continue ; 小于等于跳转
      mov ax,dx
continue:
      add si,2
      loop checker
; 测试结果，输出最小偶数对应的 ASCII码
      mov dx,ax
      mov ah,2     ;display
      int 21h
;
      pop si
      pop dx
      pop cx
      pop bx
```

```
;
      ret
main endp
;--------------------------------------------------------
prognam ends
;********************************************************
    end   start
```

求出绝对值最大数

```
; 5.15 数据段定义了一个有 n 个字数据的数组 M，是编写一程序求出 M 中绝对值最大的数，
; 把它放在数据段的 M+2n 单元中，并将该数的偏移地址存放在 M + 2 (n+1) 单元中。
datarea segment
    m dw -4,7,9,13,-3,0,234,2,5,22,43
    n equ $ - m - 4
datarea ends
;*************************************************************************************
prognam segment
;-------------------------------------------------------------------------------------
main proc far
     assume cs: prognam, ds: datarea
start:
     push ds
     sub  ax,ax
     push ax
;
     mov  ax,datarea
     mov  ds,ax
     mov  es,ax
     mov  si,00h
     mov  ax,00h
     mov  bx,00h
check:
     cmp  si,n
     jz   quit
     mov  cx,m[si]
     jge  positive
     neg  cx
positive:
     cmp  ax,cx
     jge  pass
     mov  ax,cx
     mov  bx,si
pass:
     add  si,2h
     jmp  check
quit:
     mov  m[si],ax
     add  bx,offset m
     add  si,2h
     mov  m[si],bx
     mov  ah,4ch
     int  21H
```

```
main endp
;--------------------------------------------------------------------------------
prognam ends
;********************************************************************************
     end start
```

去掉字符串中的0, 压缩、补零

```
; 5.12 有一个首地址为 MEM 的 100D 字数组，编程序删除数组中所有为零项，并将后续项向前压缩，最后
; 将数组的剩余部分补上零
datarea segment
    mem  dw 1,0,0,1,1,1,1,1,0,1
    temp dw 100dh dup(?)
    ; 用 temp 缓存非零数
datarea ends
;*********************************************************************************************
prognam segment
;---------------------------------------------------------------------------------------------
main proc far
     assume cs: prognam, ds: datarea
start:
     push ds
     sub  ax,ax
     push ax
;
     mov  ax,datarea
     mov  ds,ax
     mov  es,ax
;
     mov  si,offset mem
     mov  di,offset temp
     mov  bx,offset mem
     add  bx,201ch          ; 100dh * 2 + 2
     cld                    ; df = 0
check:
     cmp  si,bx
     jz   clear
     cmp  word ptr [si], 0000h
     jz   pass
     movsw
pass:
     add  si,2h
     jmp  check
clear:
     mov  cx,di
     mov  di,offset mem
     sub  cx,offset temp
     mov  si,offset temp
     cld                      ; df = 0
     rep  movsw
     sub  di,offset mem
     mov  cx,201ch
     sub  cx,di
```

```
loop1:
     mov  mem[di],00h
     add  di,2h
     dec  cx                ; cx 每次减2
     loop loop1
quit:
     mov  ah,04ch
     int  21H
main endp
;-------------------------------------------------------------------------
prognam  ends
;*************************************************************************
      end start
```

比较三个补码数，显示结果

```
; 5.21 要求比较数组 ARRAY 中的三个16位补码数，并根据比较结果在终端上显示如下信息:
;   (1)三个数都不相等显示0 (2) 两个相等显示1 (3) 三个相等显示2
.model small
.data
    array dw 0100h, 0100h, 0010h
    outputinfo db 'Output: $'
.code
start:
    mov ax, @data
    mov ds, ax
    mov es, ax

    mov dx, offset outputinfo
    mov ah, 09h
    int 21h

    mov ax, array[0]
    mov bx, array[2]
    mov cx, array[4]

    cmp ax, bx
    jz asb           ;; ax == bx
    cmp bx, cx       ;; ax != bx
    jz asb           ;; bx == cx
    cmp ax, cx       ;; ax != bx, bx != cx
    jz asc
    mov ah, 02h
    mov dl, 30h
    int 21h
    jmp quit
asb:
    cmp ax, cx
    jnz show1        ;; ax == bx, ax != cx
    mov ah, 02h

    mov dl, 32h
```

```
    int 21h
    jmp quit
asc:
    cmp bx, cx      ;; ax == cx
    jnz show1       ;; bx != cx
    mov ah, 02h
    mov dl, 32h
    int 21h
    jmp quit
show1:
    mov ah, 02h
    mov dl, 31h
    int 21h
quit:
    mov ah, 4ch
    int 21h
    end start
```

奇偶数判断（分支）

```
; 5.23 已定义两个整数变量：A,B
; (1)odd->A even->B (2) odd+1->A/B (3) even, nothing

data segment
    A dw 2
    B dw 3
data ends

code segment
    assume cs:code, ds:data
start:
    mov ax, data
    mov ds, ax
    mov ax, A
    mov bx, B
    test ax, 0001h
    jz a_even           ;; A is even
    test bx, 0001h
    jz quit     ;; A is odd and B is even
        inc ax
        inc bx
        mov A, ax
        mov B, bx
        jmp quit
    a_even:
        test bx, 0001h
        jz quit         ;; A is even and B is even
        xchg ax, bx
        mov A, ax
        mov B, bx
quit:
```

```
    mov ah, 4ch
    int 21h
code ends
    end start
```

学生成绩

```
; 5.18 把0-100D 之间的30个数存入以GRADE为首地址的30个字数组中，
; GRADE+i 表示学号为 i+1 的学生的成绩。另一个数组 RANK 为 30 个学生的名次表，
; 其中 RANK+i 的内容是学号为 i+1 的学生的名次。
; 编写一程序，根据GRADE 中的学生成绩，将学生名词填入RANK数组中。
; not do it
data segment
    grade dw 0110h, 1000h, 100dh, 0001h, 0110h, 1001h, 0110h, 1002h, 002dh, 0010h, 0100h, 0200h,
0300h, 0400h, 0500h, 0600h, 0700h, 0800h, 0101h, 0202h, 0203h, 0104h, 0305h, 0506h, 0607h,
0207h, 080ah, 070dh, 0fffh, 0ffeh
    temp dw 30 dup(?)
    rank db 30 dup(1)
    outputinfo1 db 'Student $'
    outputinfo2 db ' score : $'
    outputinfo3 db ' rank: $'
    nextline db 0dh, 0ah, '$'
data ends

code segment
    assume cs:code, ds:data
start:
    mov ax, data
    mov ds, ax
    mov es, ax

    mov si, offset grade
    mov di, offset temp
    mov cx, 1eh
    rep movsw            ;; copy to temp

    mov si, 00h
compare:
    cmp si, 3ch
    jz outputpart
    mov di, 00h
    mov bl, 01h
    mov ax, temp[si]
    nextcmp:
        cmp di, 3ch
        jz setrank
        cmp ax, temp[di]
        jae pass
        inc bl
        pass:

            add di, 2h
```

```
        jmp nextcmp
        setrank:
            shr si, 1
            mov rank[si], bl
            shl si, 1
    add si, 2h
    jmp compare

    ;; output part

outputpart:
    mov si, 00h
    mov di, 00h

outputs:
    cmp si, 3ch     ;; 1eh * 2 = 3ch
    jge quit
    mov cl, 0ah
    mov ax, di          ;; 输出 'Student xx score:
    div cl
    add ax, 3030h       ;; to ascii
    mov bx, ax
    mov dx, offset outputinfo1
    mov ah, 9h
    int 21h
    mov ah, 2h
    mov dl, bl
    int 21h             ;; 十位
    mov dl, bh          ;; 个位
    int 21h
    mov dx, offset outputinfo2
    mov ah, 9h
    int 21h

    mov ah, 02h
    mov cl, 04h
    mov bx, temp[si]
    mov ch, 00h
printnumber:
    cmp ch, 04h
    jz printleft
    mov dl, bh
    and dl, 0f0h
    shr dl, cl
    cmp dl, 09h
    jle number
    add dl, 07h
    number:
        add dl, 30h
    int 21h

    shl bx, cl
```

```
    inc ch
    jmp printnumber
printleft:
    mov dx, offset outputinfo3
    mov ah, 9h
    int 21h              ;; 输出 ' rank : '
    mov bl, 0ah
    mov ah, 00h
    mov al, rank[di]
    div bl
    mov bx, ax
    mov ah, 02h
    mov dl, bl
    add dl, 30h
    int 21h
    mov dl, bh
    add dl, 30h
    int 21h
    mov ah, 9h
    mov dx, offset nextline
    int 21h
    add si, 02h
    inc di
    jmp outputs
quit:
    mov ah, 4ch
    int 21h
code ends
    end start
```