



```

;
;  cghelper - 简单的学生成绩统计工具。
;              (汇编语言程序设计课程 上机大作业)
;
;  1953610    7/24/2020
;

.model compact
.stack 64

.data

; 用于保存数据的内存空间

Stu          Struc
recname      db 10 dup(?), '$'

0000 000A[    ]
    ??

    24
000B 0007[    ]
    ??

    24
0013 00          score      db 0
0014          Stu          Ends

0000 003C[    ]
    0001[
    000A[
    ??
    ]
    24
    ]
    0001[
    0007[
    ??
    ]
    24
    ]
    00
    ]

04B0 00          reccount   db 0

; 输入缓冲区

04B1 00 00      infield     db ?, ?, 14 dup(?)
    000E[
    ??
    ]

; 提示文本
```



```
04C1 0D 0A 24          emptyln    db 0dh, 0ah, '$'

04C4 23 20 43 6C 61 73  mnmsg_cpt  db "% Class Grade Helper", '$'
      73 20 47 72 61 64
      65 20 48 65 6C 70
      65 72 24

04D9 20 28 63 29 20 32  mnmsg_cpr  db " (c) 2020 strear.", 0dh, 0ah, '$'
      30 32 30 20 73 74
      72 65 61 72 2E 0D
      0A 24

04ED 20 20 52 65 61 64  opmsg_pmt  db " Ready: Choose an operation.", 0dh, 0ah, '$'
      79 3A 20 43 68 6F
      6F 73 65 20 61 6E
      20 6F 70 65 72 61
      74 69 6F 6E 2E 0D
      0A 24

050D 0A                opmsg       db 0ah
050E 20 20 31 2C 20 49  db " 1, Input record(s). 2, List all.", 0dh, 0ah
      6E 70 75 74 20 72
      65 63 6F 72 64 28
      73 29 2E 20 20 32
      2C 20 4C 69 73 74
      20 61 6C 6C 2E 0D
      0A

0533 20 20 33 2C 20 53  db " 3, Show analysis. 4, Clear.", 0dh, 0ah
      68 6F 77 20 61 6E
      61 6C 79 73 69 73
      2E 20 20 20 20 34
      2C 20 43 6C 65 61
      72 2E 0D 0A

0555 20 20 30 2C 20 51  db " 0, Quit.", 0dh, 0ah, 0ah, '$'
      75 69 74 2E 0D 0A
      0A 24

0563 20 20 3D 3E 20 24  selpmpt    db " => $"
0569 0D 20 20 45 6E 74  pausemsg    db 0dh, " Enter to continue...$"
      65 72 20 74 6F 20
      63 6F 6E 74 69 6E
      75 65 2E 2E 2E 24

; 提示文本: 输入记录

0581 20 20 48 6F 77 20  inmsg_askc db " How many records in total? $"
      6D 61 6E 79 20 72
      65 63 6F 72 64 73
      20 69 6E 20 74 6F
      74 61 6C 3F 20 24

059F 20 20 45 6E 74 65  inmsgprt0  db " Enter info required for student # $"
      72 20 69 6E 66 6F
      20 72 65 71 75 69
      72 65 64 20 66 6F
      72 20 73 74 75 64
      65 6E 74 20 23 24
```



```
05C3 2E 0D 0A 0A 24      inmsgprt1 db ".", 0dh, 0ah, 0ah, '$'
05C8 20 20 20 20 4E 61    inmsg_n   db "      Name | $"
      6D 65 20 7C 20 24
05D4 20 20 20 20 20 20    inmsg_i   db "      ID | $"
      49 44 20 7C 20 24
05E0 20 20 20 53 63 6F    inmsg_s   db "      Score | $"
      72 65 20 7C 20 24
05EC 0A 20 20 2A 20 52    inmsg_d   db 0ah, " * Record added.", 0dh, 0ah, '$'
      65 63 6F 72 64 20
      61 64 64 65 64 2E
      0D 0A 24

0601 20 20 2A 20 50 6C    inerr_num db " * Please enter a number below. Try again.", 0dh
      65 61 73 65 20 65    , 0ah, '$'
      6E 74 65 72 20 61
      20 6E 75 6D 62 65
      72 20 62 65 6C 6F
      77 2E 20 54 72 79
      20 61 67 61 69 6E
      2E 0D 0A 24
062F 20 20 2A 20 50 6C    inerr_score db " * Please enter a number below in [0, 100]. Try
      65 61 73 65 20 65    again.", 0dh, 0ah, '$'
      6E 74 65 72 20 61
      20 6E 75 6D 62 65
      72 20 62 65 6C 6F
      77 20 69 6E 20 5B
      30 2C 20 31 30 30
      5D 2E 20 54 72 79
      20 61 67 61 69 6E
      2E 0D 0A 24
0669 20 20 2A 20 43 61    inerr_getc db " * Can't save so many records. Try again.", 0dh,
      6E 27 74 20 73 61    0ah, '$'
      76 65 20 73 6F 20
      6D 61 6E 79 20 72
      65 63 6F 72 64 73
      2E 20 54 72 79 20
      61 67 61 69 6E 2E
      0D 0A 24

0696 20 20 2A 20 41 6C    clmsg     db " * All records are cleared.", 0dh, 0ah, '$'
      6C 20 72 65 63 6F
      72 64 73 20 61 72
      65 20 63 6C 65 61
      72 65 64 2E 0D 0A
      24

      ; 提示文本: 列出记录

06B5 20 20 20 49 44 20    lsmg_hdr  db "      ID      Name      Score", 0dh, 0ah
      20 20 20 20 20 4E
      61 6D 65 20 20 20
      20 20 20 20 53 63
```



```

    6F 72 65 0D 0A
06D2  20 20 20 2D 2D 2D          db "  ----- $"
    2D 2D 2D 2D 2D 2D
    2D 2D 2D 2D 2D 2D
    2D 2D 2D 2D 2D 2D
    2D 2D 2D 24

06EE  0D 0A 20 20 2A 20          lsmmsg_cnt0 db 0dh, 0ah, " * Totally $"
    54 6F 74 61 6C 6C
    79 20 24

06FD  20 72 65 63 6F 72          lsmmsg_cnt1 db " record(s).", 0dh, 0ah, '$'
    64 28 73 29 2E 0D
    0A 24

; 提示文本: 分析记录

070B  20 20 2A 20 43 75          azerr_empty db " * Currently no records. Enter some and try again.", 0dh, 0ah, '$'
    72 72 65 6E 74 6C
    79 20 6E 6F 20 72
    65 63 6F 72 64 73
    2E 20 45 6E 74 65
    72 20 73 6F 6D 65
    20 61 6E 64 20 74
    72 79 20 61 67 61
    69 6E 2E 0D 0A 24

0741  20 20 2A 20 54 6F          azmsg_cnt0 db " * Totally $"
    74 61 6C 6C 79 20
    24

074E  20 73 63 6F 72 65          azmsg_cnt1 db " score records calcuted.", 0dh, 0ah, 0ah
    20 72 65 63 6F 72
    64 73 20 63 61 6C
    63 75 74 65 64 2E
    0D 0A 0A

0769  20 20 2A 20 53 74          db " * Students in all segments counts as follows:",
    0dh, 0ah, 0ah, '$'
    75 64 65 6E 74 73
    20 69 6E 20 61 6C
    6C 20 73 65 67 6D
    65 6E 74 73 20 63
    6F 75 6E 74 73 20
    61 73 20 66 6F 6C
    6C 6F 77 73 3A 0D
    0A 0A 24

079C  20 20 20 20 45 78          azmsg_ex db "      Excellent: $"
    63 65 6C 6C 65 6E
    74 3A 20 24

07AC  20 20 20 20 20 20          azmsg_gd db "      Good: $"
    20 20 20 47 6F 6F
    64 3A 20 24

07BC  20 20 20 20 20 20          azmsg_md db "      Medium: $"
    20 4D 65 64 69 75
    6D 3A 20 24

07CC  20 20 20 20 20 20          azmsg_ps db "      Pass: $"
    20 20 20 50 61 73
```



```

    73 3A 20 24
07DC 20 20 20 20 20 20      azmsg_fl    db "          Fail: $"
    20 20 20 46 61 69
    6C 3A 20 24
07EC 20 20 20 20 20 20      azmsg_ab    db "          Absent: $"
    20 41 62 73 65 6E
    74 3A 20 24
07FC 20 74 6F 74 61 6C      scmsg_prt0  db " total, $"
    2C 20 24
0805 25 2E 0D 0A 24          scmsg_prt1  db "%.", 0dh, 0ah, '$'
080A 0D 0A 20 20 2A 20      azmsg_max   db 0dh, 0ah, " * The highest score is $"
    54 68 65 20 68 69
    67 68 65 73 74 20
    73 63 6F 72 65 20
    69 73 20 24
0826 2C 20 6C 6F 77 65      azmsg_min   db ", lowest $"
    73 74 20 24
0830 2C 20 61 6E 64 20      azmsg_avg   db ", and the average $"
    74 68 65 20 61 76
    65 72 61 67 65 20
    24
0843 2E 0D 0A 24          azmsg_end   db ". ", 0dh, 0ah, '$'

.code

; 主程序

0000      Entry:
0000 1E      push ds          ; 返回地址入栈, 用于正常退出
0001 33 C0    xor ax, ax
0003 50      push ax

0004 B8 ---- R      mov ax, @data      ; 初始化数据寄存器
0007 8E D8      mov ds, ax
0009 8E C0      mov es, ax

000B E8 0032 R      call putbanner      ; 开始运行, 显示程序名称

000E
000E E8 0045 R      .main_cycle:
0011 E8 0058 R      call putmenu      ; 显示菜单
                                call getop      ; 请求操作

0014 85 DB      test bx, bx
0016 74 11      jz .main_exit
0018 2E: FF 97 0028 R      call word ptr CS:optcase[bx-2]

001D 8D 16 04C1 R      lea dx, emptyln
0021 E8 0086 R      call print
0024 E8 0099 R      call pause
0027 EB E5      jmp .main_cycle

0029
0029 CB      .main_exit:      ; 正常退出
                                retf
```



```
002A 0158 R 0288 R 0303 R      optcase    dw op_input, op_list, op_analyze, op_clear
      027B R

0032                          putbanner:
0032 B3 0F                      mov bl, 0fh
0034 8D 16 04C4 R              lea dx, mnmsg_cpt
0038 E8 008B R                  call printc

003B B3 08                      mov bl, 08h
003D 8D 16 04D9 R              lea dx, mnmsg_cpr
0041 E8 008B R                  call printc
0044 C3                          ret

0045                          putmenu:
0045 B3 0C                      mov bl, 0ch
0047 8D 16 04ED R              lea dx, opmsg_pmt
004B E8 008B R                  call printc

004E B3 07                      mov bl, 07h
0050 8D 16 050D R              lea dx, opmsg
0054 E8 008B R                  call printc
0057 C3                          ret

0058                          getop:
0058 B3 09                      mov bl, 09h
005A 8D 16 0563 R              lea dx, selpmpt
005E E8 008B R                  call printc

0061                          .getop_cycle:
0061 B4 08                      mov ah, 08h
0063 CD 21                      int 21h
                                ; 提示用户选择一个选项
                                ; 读一个输入

0065 8A D0                      mov dl, al
0067 2C 30                      sub al, '0'
0069 7C F6                      jl .getop_cycle
006B 3C 04                      cmp al, 4
006D 7F F2                      jg .getop_cycle
                                ; 选项序号

006F 32 E4                      xor ah, ah
0071 02 C0                      add al, al
0073 50                          push ax
                                ; 找出该功能的入口地址

0074 B4 02                      mov ah, 02h
0076 CD 21                      int 21h
                                ; 回显输入

0078 B3 07                      mov bl, 07h
007A 8D 16 04C1 R              lea dx, emptyln
007E E8 0086 R                  call print
0081 E8 008B R                  call printc
                                ; 清除输出颜色

0084 5B                          pop bx
0085 C3                          ret
                                ; 返回选项
```



; 功能调用的封装

```
0086      print:
0086 B4 09      mov ah, 09h
0088 CD 21      int 21h
008A C3      ret

008B      printc:
008B B8 0900     mov ax, 0900h
008E B7 00      mov bh, 0h
0090 B9 07D0     mov cx, 07d0h
0093 CD 10      int 10h
0095 E8 0086 R   call print
0098 C3      ret

0099      pause:
0099 8D 16 0569 R lea dx, pausemsg
009D E8 0086 R   call print

00A0 B4 08      mov ah, 08h
00A2 CD 21      int 21h
00A4 3C 0D      cmp al, 0dh
00A6 75 F1      jne pause

00A8 B4 02      mov ah, 02h
00AA B2 0D      mov dl, 0dh
00AC CD 21      int 21h
00AE C3      ret

00AF      parsenum:
00AF 8A 16 04B2 R mov dl, infield[1]
00B3 32 F6      xor dh, dh
00B5 8B FA      mov di, dx
00B7 B2 0A      mov dl, 10

00B9 33 C0      xor ax, ax
00BB 33 F6      xor si, si
00BD EB 09 90    jmp .parsen_loopin

00C0      .parsen_loop:
00C0 F6 E2      mul dl
00C2 02 84 04B2 R add al, infield[si+1]
00C6 2C 30      sub al, '0'

00C8      .parsen_loopin:
00C8 46      inc si
00C9 3B F7      cmp si, di
00CB 7E F3      jle .parsen_loop
00CD C3      ret

00CE      putdecimal_right:      ; 十进制输出AL, 右对齐
00CE 50      push ax
00CF BB 000A     mov bx, 10
00D2 EB 0F 90    jmp .putdecimal_spc_loopin
```



```
00D5          .putdecial_spc_loop:
00D5 32 E4      xor ah, ah
00D7 F6 F3      div bl
00D9 84 C0      test al, al
00DB 75 06      jnz .putdecial_spc_loopin

00DD 50          push ax
00DE B4 02      mov ah, 2h
00E0 CD 21      int 21h
00E2 58          pop ax

00E3          .putdecial_spc_loopin:
00E3 E2 F0      loop .putdecial_spc_loop
00E5 58          pop ax

00E6          putdecial:                ; 十进制输出AL (递归方法)
00E6 BB 000A     mov bx, 10
00E9 E8 00ED R  call .putdecial
00EC C3         ret

00ED          .putdecial:
00ED 50          push ax
00EE 32 E4      xor ah, ah                ; 将数字分为AH 和AL 两部分
00F0 F6 F3      div bl                ; AL = 商, AH = 余数
00F2 84 C0      test al, al           ; 是前导0 吗?
00F4 74 03      je .pdec_digin        ; - 如果不是, 显示该位余数
00F6 E8 00ED R  call .putdecial           ; 继续处理商

00F9          .pdec_digin:
00F9 80 C4 30     add ah, '0'
00FC 8A D4      mov dl, ah
00FE B4 09      mov ah, 09h           ; 设置输出颜色
0100 B9 0001     mov cx, 01h
0103 CD 10      int 10h
0105 B4 02      mov ah, 02h           ; 输出数字
0107 CD 21      int 21h
0109 58          pop ax
010A C3         ret

010B          input:
010B E8 0086 R   call print                ; 输入提示语
010E 52          push dx

010F 8D 16 04B1 R lea dx, infield
0113 B4 0A      mov ah, 0Ah
0115 CD 21      int 21h

0117 5A          pop dx
0118 FF D3      call bx                ; 检查输入
011A 72 EF      jc input

011C 8D 16 04C1 R lea dx, emptyln
0120 E8 0086 R   call print
0123 C3         ret
```




```
0124                                inchk_num:
0124 8A 0E 04B2 R                    mov cl, infield[1]      ; 检查非空
0128 84 C9                          test cl, cl
012A 74 28                          jz inchk_bad

012C 32 ED                          xor ch, ch
012E 8B F1                          mov si, cx
0130 EB 0D 90                       jmp .inchkn_loopin

0133                                .inchkn_loop:
0133 8A 84 04B3 R                    mov al, infield[si+2]
0137 3C 30                          cmp al, '0'
0139 72 0D                          jnae .inchkn_fail
013B 3C 39                          cmp al, '9'
013D 77 09                          jnbe .inchkn_fail

013F                                .inchkn_loopin:
013F 4E                              dec si
0140 83 FE 00                       cmp si, 0
0143 7D EE                          jge .inchkn_loop
0145 EB 0F 90                       jmp inchk_good

0148                                .inchkn_fail:
0148 52                              push dx
0149 8D 16 0601 R                   lea dx, inerr_num
014D E8 0086 R                     call print
0150 5A                              pop dx
0151 EB 01 90                       jmp inchk_bad

0154                                inchk_bad:
0154 F9                              stc
0155 C3                              ret

0156                                inchk_good:
0156 F8                              clc
0157 C3                              ret

; 操作

; 输入

0158                                op_input:
0158 8D 16 0581 R                   lea dx, inmsg_askc      ; 确定本次输入数据总数
015C 8D 1E 0194 R                   lea bx, .inchk_getc
0160 C6 06 04B1 R 03               mov infield, 3
0165 E8 010B R                     call input
0168 E8 01E5 R                     call .opin_prepcycle

016B EB 1E 90                       jmp .opin_newcycle

016E                                .opin_cycle:
016E 50                              push ax
```



```
016F 51                push cx

0170 8D 16 059F R        lea dx, inmsgprt0      ; 提示即将输入的数据序数
0174 E8 0086 R          call print
0177 8A C1              mov al, cl
0179 E8 00E6 R          call putdecal
017C 8D 16 05C3 R        lea dx, inmsgprt1
0180 E8 0086 R          call print

0183 E8 01FD R          call .opin_one      ; 输入一条数据
0186 83 C5 14          add bp, size Stu

0189 59                pop cx
018A 58                pop ax

018B                  .opin_newcycle:
018B 41                inc cx
018C 3B C8              cmp cx, ax
018E 7E DE              jle .opin_cycle

0190 A2 04B0 R          mov reccount, al      ; 数据输入完毕, 返回
0193 C3                ret

0194                  .inchk_getc:
0194 E8 0124 R          call inchk_num
0197 72 BB              jc inchk_bad

0199 52                push dx
019A E8 00AF R          call parsenum
019D 5A                pop dx
019E 02 06 04B0 R        add al, reccount

01A2 32 E4              xor ah, ah
01A4 8B F8              mov di, ax
01A6 3C 3C              cmp al, length recpool
01A8 76 AC              jbe inchk_good

01AA 52                push dx
01AB 8D 16 0669 R        lea dx, inerr_getc
01AF E8 0086 R          call print
01B2 5A                pop dx
01B3 EB 9F              jmp inchk_bad

01B5                  .inchk_score:
01B5 E8 0124 R          call inchk_num
01B8 72 9A              jc inchk_bad

01BA 80 F9 03           cmp cl, 3
01BD 72 97              jb inchk_good

01BF 80 3E 04B3 R 31     cmp infield[2], '1'
01C4 72 90              jb inchk_good
01C6 77 11              ja .inchk_score_fail

01C8 80 3E 04B4 R 30     cmp infield[3], '0'
```



```
01CD 77 0A          ja .inchk_score_fail

01CF 80 3E 04B5 R 30    cmp infield[4], '0'
01D4 77 03          ja .inchk_score_fail
01D6 E9 0156 R        jmp inchk_good

01D9              .inchk_score_fail:
01D9 52            push dx
01DA 8D 16 062F R      lea dx, inerr_score
01DE E8 0086 R        call print
01E1 5A            pop dx
01E2 E9 0154 R        jmp inchk_bad

01E5              .opin_prepcycle:
01E5 32 E4          xor ah, ah
01E7 A0 04B0 R        mov al, reccount
01EA B3 14          mov bl, size Stu
01EC F6 E3          mul bl
01EE 8B E8          mov bp, ax
01F0 81 C5 0000 R      add bp, offset recpool

01F4 8B C7          mov ax, di
01F6 32 ED          xor ch, ch
01F8 8A 0E 04B0 R      mov cl, reccount
01FC C3            ret

01FD              .opin_one:
01FD 8D 16 05C8 R      lea dx, inmsg_n          ; 姓名
0201 8D 1E 0156 R      lea bx, inchk_good
0205 C6 06 04B1 R 0B    mov infield, size recname+1
020A E8 010B R        call input

020D 8A 0E 04B2 R      mov cl, infield+1        ; 从输入缓冲转移到数据区
0211 32 ED          xor ch, ch
0213 8D 36 04B3 R      lea si, infield+2
0217 8B FD          mov di, bp
0219 83 C7 00        add di, Stu.recname
021C FC            cld
021D F3/ A4          rep movsb

021F B0 20          mov al, ' '          ; 清空原有内容
0221 B1 0A          mov cl, size recname
0223 2A 0E 04B2 R      sub cl, infield+1
0227 F3/ AA          rep stosb

0229 8D 16 05D4 R      lea dx, inmsg_i          ; 学号
022D 8D 1E 0124 R      lea bx, inchk_num
0231 C6 06 04B1 R 08    mov infield, size id+1
0236 E8 010B R        call input

0239 8A 0E 04B2 R      mov cl, infield+1        ; 从输入缓冲转移到数据区
023D 32 ED          xor ch, ch
023F 8D 36 04B3 R      lea si, infield+2
0243 8B FD          mov di, bp
0245 83 C7 12        add di, Stu.id+size id
```



```
0248 2B F9          sub di, cx
024A FC          cld
024B F3/ A4       rep movsb

024D B0 20       mov al, ' '          ; 清空原有内容
024F 8B FD       mov di, bp
0251 83 C7 0B     add di, Stu.id
0254 B1 07       mov cl, size id
0256 2A 0E 04B2 R sub cl, infield+1
025A F3/ AA       rep stosb

025C 8D 16 05E0 R lea dx, inmsg_s      ; 成绩
0260 8D 1E 01B5 R lea bx, .inchk_score
0264 C6 06 04B1 R 04 mov infield, 4
0269 E8 010B R    call input

026C E8 00AF R    call parsenum          ; 从输入缓冲转移到数据区
026F 3E: 88 46 13 mov DS:[bp+Stu.score], al

0273 8D 16 05EC R lea dx, inmsg_d
0277 E8 0086 R    call print
027A C3          ret

; 清空

027B          op_clear:
027B C6 06 04B0 R 00 mov reccount, 0

0280 8D 16 0696 R lea dx, clmsg
0284 E8 0086 R    call print

0287 C3          ret

; 列表

0288          op_list:
0288 8D 16 06B5 R lea dx, lsmsg_hdr
028C E8 0086 R    call print

028F 8D 36 0000 R lea si, recpool
0293 32 C9       xor cl, cl
0295 EB 32 90     jmp .opls_newcycle

0298          .opls_cycle:
0298 B2 20       mov dl, ' '          ; 输出空格
029A B4 02       mov ah, 2h
             rept 3
             int 21h
             endm
029C CD 21       1          int 21h
029E CD 21       1          int 21h
02A0 CD 21       1          int 21h
```



```
02A2 8D 54 0B          lea dx, [si+Stu.id]      ; 学号
02A5 E8 0086 R        call print

02A8 B2 20            mov dl, ' '              ; 输出空格
02AA B4 02            mov ah, 2h
02AC CD 21            int 21h

02AE 8D 14            lea dx, [si+Stu.recname]; 姓名
02B0 E8 0086 R        call print

02B3 B2 20            mov dl, ' '              ; 输出空格
02B5 B4 02            mov ah, 2h
02B7 CD 21            int 21h
02B9 CD 21            int 21h

02BB 8A 44 13          mov al, [si+Stu.score] ; 成绩
02BE 51                push cx
02BF B9 0003            mov cx, 3
02C2 E8 00CE R        call putdecalf_right
02C5 59                pop cx

02C6 83 C6 14          add si, size Stu

02C9                  .opls_newcycle:
02C9 8D 16 04C1 R      lea dx, emptyln
02CD E8 0086 R        call print

02D0 FE C1            inc cl
02D2 8A C1            mov al, cl
02D4 32 E4            xor ah, ah
02D6 B3 14            mov bl, 20                ; 当输出数据达到该值时, 暂停一下
02D8 F6 F3            div bl
02DA 84 E4            test ah, ah
02DC 75 0A            jnz .opls_nopause

02DE 8D 16 04C1 R      lea dx, emptyln
02E2 E8 0086 R        call print
02E5 E8 0099 R        call pause

02E8                  .opls_nopause:
02E8 3A 0E 04B0 R      cmp cl, reccount
02EC 7E AA            jle .opls_cycle

02EE 8D 16 06EE R      lea dx, lsmmsg_cnt0      ; 列表完毕, 显示数据总数
02F2 E8 0086 R        call print

02F5 A0 04B0 R          mov al, reccount
02F8 E8 00E6 R        call putdecalf

02FB 8D 16 06FD R      lea dx, lsmmsg_cnt1
02FF E8 0086 R        call print

0302 C3                ret
```



```

; 统计

0303                                op_analyze:
0303 80 3E 04B0 R 00                cmp reccount, 0           ; 确认非空
0308 75 12                        jnz .opaz_init

030A 8D 16 070B R                lea dx, azerr_empty
030E E8 0086 R                  call print
0311 C3                        ret

; 存放临时数据

0312 00                        .opaz_ex_c db ?
0313 00                        .opaz_gd_c db ?
0314 00                        .opaz_md_c db ?
0315 00                        .opaz_ps_c db ?
0316 00                        .opaz_fl_c db ?
0317 00                        .opaz_ab_c db ?

0318 00                        .opaz_max db ?
0319 00                        .opaz_min db ?
031A 0000                      .opaz_sum dw ?

031C                                .opaz_init:
031C 2E: C6 06 0312 R 00        mov .opaz_ex_c, 0
0322 2E: C6 06 0313 R 00        mov .opaz_gd_c, 0
0328 2E: C6 06 0314 R 00        mov .opaz_md_c, 0
032E 2E: C6 06 0315 R 00        mov .opaz_ps_c, 0
0334 2E: C6 06 0316 R 00        mov .opaz_fl_c, 0
033A 2E: C6 06 0317 R 00        mov .opaz_ab_c, 0
0340 2E: C6 06 0318 R 80        mov .opaz_max, -128
0346 2E: C6 06 0319 R 7F        mov .opaz_min, 127
034C 2E: C7 06 031A R 0000      mov .opaz_sum, 0

0353 32 E4                    xor ah, ah
0355 A0 04B0 R                mov al, reccount
0358 FE C8                    dec al
035A B3 14                    mov bl, size Stu
035C F6 E3                    mul bl
035E 8B C8                    mov cx, ax

0360                                .opaz_cycle:
0360 8B D9                    mov bx, cx
0362 8A 87 0013 R            mov al, recpool[bx].score
0366 32 E4                    xor ah, ah
0368 2E: 01 06 031A R        add .opaz_sum, ax

036D 2E: 3A 06 0318 R        cmp al, .opaz_max
0372 7E 04                    jng .opaz_notmax
0374 2E: A2 0318 R          mov .opaz_max, al

0378                                .opaz_notmax:
0378 84 C0                    test al, al
037A 74 0B                    jz .opaz_notmin
037C 2E: 3A 06 0319 R        cmp al, .opaz_min
```



```
0381 7D 04                jnl .opaz_notmin
0383 2E: A2 0319 R        mov .opaz_min, al

0387                .opaz_notmin:
0387 3D 005A                cmp ax, 90
038A 7C 08                jnge .opaz_notex
038C 2E: FE 06 0312 R    inc .opaz_ex_c
0391 EB 39 90            jmp .opaz_newcycle

0394                .opaz_notex:
0394 3D 0050                cmp ax, 80
0397 7C 08                jnge .opaz_notgd
0399 2E: FE 06 0313 R    inc .opaz_gd_c
039E EB 2C 90            jmp .opaz_newcycle

03A1                .opaz_notgd:
03A1 3D 0046                cmp ax, 70
03A4 7C 08                jnge .opaz_notmd
03A6 2E: FE 06 0314 R    inc .opaz_md_c
03AB EB 1F 90            jmp .opaz_newcycle

03AE                .opaz_notmd:
03AE 3D 003C                cmp ax, 60
03B1 7C 08                jnge .opaz_notps
03B3 2E: FE 06 0315 R    inc .opaz_ps_c
03B8 EB 12 90            jmp .opaz_newcycle

03BB                .opaz_notps:
03BB 85 C0                test ax, ax
03BD 74 08                jz .opaz_absent
03BF 2E: FE 06 0316 R    inc .opaz_fl_c
03C4 EB 06 90            jmp .opaz_newcycle

03C7                .opaz_absent:
03C7 2E: FE 06 0317 R    inc .opaz_ab_c

03CC                .opaz_newcycle:
03CC 83 E9 14                sub cx, size Stu
03CF 7D 8F                jge .opaz_cycle

; 统计循环结束, 显示结果

03D1 8D 16 0741 R        lea dx, azmsg_cnt0 ; 总人数
03D5 E8 0086 R        call print
03D8 A0 04B0 R        mov al, reccount
03DB 32 E4                xor ah, ah ; 保留用于计算
03DD 8B F0                mov si, ax
03DF 8B FE                mov di, si ; 用于四舍五入
03E1 D1 EF                shr di, 1
03E3 E8 00E6 R        call putdec1
03E6 8D 16 074E R        lea dx, azmsg_cnt1
03EA E8 0086 R        call print

03ED 2E: 8A 3E 0312 R    mov bh, .opaz_ex_c ; 得优人数
03F2 84 FF                test bh, bh
```



```
03F4 74 07                jz .opaz_noex
03F6 8D 16 079C R          lea dx, azmsg_ex
03FA E8 04AE R            call .opaz_segcount

03FD                    .opaz_noex:
03FD 2E: 8A 3E 0313 R      mov bh, .opaz_gd_c
0402 84 FF                test bh, bh
0404 74 07                jz .opaz_nogd
0406 8D 16 07AC R          lea dx, azmsg_gd
040A E8 04AE R            call .opaz_segcount

040D                    .opaz_nogd:
040D 2E: 8A 3E 0314 R      mov bh, .opaz_md_c
0412 84 FF                test bh, bh
0414 74 07                jz .opaz_nomd
0416 8D 16 07BC R          lea dx, azmsg_md
041A E8 04AE R            call .opaz_segcount

041D                    .opaz_nomd:
041D 2E: 8A 3E 0315 R      mov bh, .opaz_ps_c
0422 84 FF                test bh, bh
0424 74 07                jz .opaz_nops
0426 8D 16 07CC R          lea dx, azmsg_ps
042A E8 04AE R            call .opaz_segcount

042D                    .opaz_nops:
042D 2E: 8A 3E 0316 R      mov bh, .opaz_fl_c
0432 84 FF                test bh, bh
0434 74 07                jz .opaz_nofl
0436 8D 16 07DC R          lea dx, azmsg_fl
043A E8 04AE R            call .opaz_segcount

043D                    .opaz_nofl:
043D 2E: 8A 3E 0317 R      mov bh, .opaz_ab_c
0442 84 FF                test bh, bh
0444 74 0F                jz .opaz_noab

0446 8D 16 07EC R          lea dx, azmsg_ab
044A E8 04AE R            call .opaz_segcount

044D 32 E4                xor ah, ah
044F 2E: A0 0317 R          mov al, .opaz_ab_c
0453 2B F0                sub si, ax                ; 计算平均分时不考虑缺考

0455                    .opaz_noab:
0455 8D 16 080A R          lea dx, azmsg_max
0459 E8 0086 R            call print
045C 2E: A0 0318 R          mov al, .opaz_max
0460 E8 00E6 R            call putdecval

0463 8D 16 0826 R          lea dx, azmsg_min
0467 E8 0086 R            call print
046A 2E: A0 0319 R          mov al, .opaz_min
046E 3C 7F                cmp al, 127
0470 75 02                jne .opaz_minavl
```




```
0472 32 C0                xor al, al
0474                .opaz_minavl:
0474 E8 00E6 R            call putdecal

0477 8D 16 0830 R          lea dx, azmsg_avg
047B E8 0086 R            call print

047E 85 F6                test si, si
0480 75 05                jnz .opaz_avgavl
0482 32 F6                xor dh, dh
0484 EB 1B 90            jmp .opaz_avgdigit
; 全部缺考时, 没有平均成绩可给出

0487                .opaz_avgavl:
0487 2E: A1 031A R          mov ax, .opaz_sum
048B BB 000A            mov bx, 10
048E F7 E3              mul bx
0490 03 C7              add ax, di
0492 F7 F6              div si
0494 F6 F3              div bl
0496 8A F4              mov dh, ah
0498 E8 00E6 R          call putdecal

049B B4 02                mov ah, 02h
049D B2 2E                mov dl, '.'
049F CD 21                int 21h

04A1                .opaz_avgdigit:
04A1 8A C6                mov al, dh
04A3 E8 00E6 R          call putdecal

04A6 8D 16 0843 R          lea dx, azmsg_end
04AA E8 0086 R            call print
04AD C3                  ret

04AE                .opaz_segcount:
04AE E8 0086 R            call print
04B1 8A C7                mov al, bh
04B3 E8 00E6 R          call putdecal
04B6 8A F8                mov bh, al
04B8 8D 16 07FC R          lea dx, scmsg_ptr0
04BC E8 0086 R            call print

04BF 32 E4                xor ah, ah
04C1 8A C7                mov al, bh
04C3 BB 03E8            mov bx, 1000
04C6 F7 E3              mul bx
04C8 03 C7              add ax, di
04CA F7 F6              div si
04CC B3 0A                mov bl, 10
04CE F6 F3              div bl
04D0 E8 00E6 R          call putdecal

04D3 8A DC                mov bl, ah
04D5 B4 02                mov ah, 02h
```



```
04D7 B2 2E          mov dl, '.'
04D9 CD 21          int 21h
04DB 80 C3 30       add bl, '0'
04DE 8A D3          mov dl, bl
04E0 CD 21          int 21h

04E2 8D 16 0805 R    lea dx, scmsg_prt1
04E6 E8 0086 R       call print
04E9 C3             ret

end Entry
```



Symbols-1

Structures and Records:

Name	Width	# fields	Mask	Initial
	Shift	Width		
STU	0014	0003		
RECNAME	0000			
ID	000B			
SCORE	0013			

Segments and Groups:

Name	Length	Align	Combine	Class
DGROUP	GROUP			
_DATA	0847	WORD	PUBLIC	'DATA'
STACK	0040	PARA	STACK	'STACK'
_TEXT	04EA	WORD	PUBLIC	'CODE'

Symbols:

Name	Type	Value	Attr
AZERR_EMPTY	L BYTE	070B	_DATA
AZMSG_AB	L BYTE	07EC	_DATA
AZMSG_AVG	L BYTE	0830	_DATA
AZMSG_CNT0	L BYTE	0741	_DATA
AZMSG_CNT1	L BYTE	074E	_DATA
AZMSG_END	L BYTE	0843	_DATA
AZMSG_EX	L BYTE	079C	_DATA
AZMSG_FL	L BYTE	07DC	_DATA
AZMSG_GD	L BYTE	07AC	_DATA
AZMSG_MAX	L BYTE	080A	_DATA
AZMSG_MD	L BYTE	07BC	_DATA
AZMSG_MIN	L BYTE	0826	_DATA
AZMSG_PS	L BYTE	07CC	_DATA
CLMSG	L BYTE	0696	_DATA
EMPTYLN	L BYTE	04C1	_DATA
ENTRY	L NEAR	0000	_TEXT
GETOP	L NEAR	0058	_TEXT
INCHK_BAD	L NEAR	0154	_TEXT
INCHK_GOOD	L NEAR	0156	_TEXT
INCHK_NUM	L NEAR	0124	_TEXT
INERR_GETC	L BYTE	0669	_DATA
INERR_NUM	L BYTE	0601	_DATA
INERR_SCORE	L BYTE	062F	_DATA
INFIELD	L BYTE	04B1	_DATA
INMSGPRT0	L BYTE	059F	_DATA
INMSGPRT1	L BYTE	05C3	_DATA
INMSG_ASKC	L BYTE	0581	_DATA



Symbols-2

INMSG_D	L BYTE 05EC	_DATA	
INMSG_I	L BYTE 05D4	_DATA	
INMSG_N	L BYTE 05C8	_DATA	
INMSG_S	L BYTE 05E0	_DATA	
INPUT	L NEAR 010B	_TEXT	
LSMSG_CNT0	L BYTE 06EE	_DATA	
LSMSG_CNT1	L BYTE 06FD	_DATA	
LSMSG_HDR	L BYTE 06B5	_DATA	
MNMSG_CPR	L BYTE 04D9	_DATA	
MNMSG_CPT	L BYTE 04C4	_DATA	
OPMSG	L BYTE 050D	_DATA	
OPMSG_PMT	L BYTE 04ED	_DATA	
OPTCASE	L 0002 002A	_TEXT	
OP_ANALYZE	L NEAR 0303	_TEXT	
OP_CLEAR	L NEAR 027B	_TEXT	
OP_INPUT	L NEAR 0158	_TEXT	
OP_LIST	L NEAR 0288	_TEXT	
PARSENUM	L NEAR 00AF	_TEXT	
PAUSE	L NEAR 0099	_TEXT	
PAUSEMSG	L BYTE 0569	_DATA	
PRINT	L NEAR 0086	_TEXT	
PRINTC	L NEAR 008B	_TEXT	
PUTBANNER	L NEAR 0032	_TEXT	
PUTDECAL	L NEAR 00E6	_TEXT	
PUTDECAL_RIGHT	L NEAR 00CE	_TEXT	
PUTMENU	L NEAR 0045	_TEXT	
RECCOUNT	L BYTE 04B0	_DATA	
RECPool	L 0000	_DATA	Length = 003C
SCMSG_PRT0	L BYTE 07FC	_DATA	
SCMSG_PRT1	L BYTE 0805	_DATA	
SELPMP	L BYTE 0563	_DATA	
.GETOP_CYCLE	L NEAR 0061	_TEXT	
.INCHK_FAIL	L NEAR 0148	_TEXT	
.INCHK_LOOP	L NEAR 0133	_TEXT	
.INCHK_LOOPIN	L NEAR 013F	_TEXT	
.INCHK_GETC	L NEAR 0194	_TEXT	
.INCHK_SCORE	L NEAR 01B5	_TEXT	
.INCHK_SCORE_FAIL	L NEAR 01D9	_TEXT	
.MAIN_CYCLE	L NEAR 000E	_TEXT	
.MAIN_EXIT	L NEAR 0029	_TEXT	
.OPAZ_ABSENT	L NEAR 03C7	_TEXT	
.OPAZ_AB_C	L BYTE 0317	_TEXT	
.OPAZ_AVGAVL	L NEAR 0487	_TEXT	
.OPAZ_AVGDIGIT	L NEAR 04A1	_TEXT	
.OPAZ_CYCLE	L NEAR 0360	_TEXT	
.OPAZ_EX_C	L BYTE 0312	_TEXT	
.OPAZ_FL_C	L BYTE 0316	_TEXT	
.OPAZ_GD_C	L BYTE 0313	_TEXT	



Symbols-3

.OPAZ_INIT	L NEAR 031C	_TEXT
.OPAZ_MAX	L BYTE 0318	_TEXT
.OPAZ_MD_C	L BYTE 0314	_TEXT
.OPAZ_MIN	L BYTE 0319	_TEXT
.OPAZ_MINAVL	L NEAR 0474	_TEXT
.OPAZ_NEWCYCLE	L NEAR 03CC	_TEXT
.OPAZ_NOAB	L NEAR 0455	_TEXT
.OPAZ_NOEX	L NEAR 03FD	_TEXT
.OPAZ_NOFL	L NEAR 043D	_TEXT
.OPAZ_NOGD	L NEAR 040D	_TEXT
.OPAZ_NOMD	L NEAR 041D	_TEXT
.OPAZ_NOPS	L NEAR 042D	_TEXT
.OPAZ_NOTEX	L NEAR 0394	_TEXT
.OPAZ_NOTGD	L NEAR 03A1	_TEXT
.OPAZ_NOTMAX	L NEAR 0378	_TEXT
.OPAZ_NOTMD	L NEAR 03AE	_TEXT
.OPAZ_NOTMIN	L NEAR 0387	_TEXT
.OPAZ_NOTPS	L NEAR 03BB	_TEXT
.OPAZ_PS_C	L BYTE 0315	_TEXT
.OPAZ_SEGCOUNT	L NEAR 04AE	_TEXT
.OPAZ_SUM	L WORD 031A	_TEXT
.OPIN_CYCLE	L NEAR 016E	_TEXT
.OPIN_NEWCYCLE	L NEAR 018B	_TEXT
.OPIN_ONE	L NEAR 01FD	_TEXT
.OPIN_PREPCYCLE	L NEAR 01E5	_TEXT
.OPLS_CYCLE	L NEAR 0298	_TEXT
.OPLS_NEWCYCLE	L NEAR 02C9	_TEXT
.OPLS_NOPAUSE	L NEAR 02E8	_TEXT
.PARSEN_LOOP	L NEAR 00C0	_TEXT
.PARSEN_LOOPIN	L NEAR 00C8	_TEXT
.PDEC_DIGIN	L NEAR 00F9	_TEXT
.PUTDECAL	L NEAR 00ED	_TEXT
.PUTDECAL_SPC_LOOP	L NEAR 00D5	_TEXT
.PUTDECAL_SPC_LOOPIN	L NEAR 00E3	_TEXT
@CODE	TEXT	_TEXT
@CODESIZE	TEXT	0
@CPU	TEXT	0101h
@DATASIZE	TEXT	1
@FILENAME	TEXT	cghelper
@VERSION	TEXT	510

784 Source Lines

787 Total Lines

130 Symbols

47262 + 414684 Bytes symbol space free

0 Warning Errors

0 Severe Errors