**01**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1、11个同学的汇编成绩分别为（括号中为学号）：89（01）,76（02）,91（03）,93（04）,58（05）,64（06）,83（07）,81（08）,45（09）,72（10）,84（11)

（1）统计成绩位于100~90,89~80,79~70,69~60,59~0等区间的学生人数

（2）输出最高分，最低分，以及11个同学的平均成绩

（3）附加要求：将11个学生分数从高到低排序并将排序后的成绩和对应的学号输出

DATA SEGMENT

    BUF DB 89,76,91,93,58,64,83,81,45,72,84

    TJ  DB 5 DUP(0)           ;0位置放100到90的人数，1位置放90到80的人数

    INFO1 DB 0DH,0AH,'THE numberof people between100-90 90-80 80-70 70-60 59-0 is:$'

    INFO2 DB 0DH,0AH,'cong gao dao di pai xu hou ,11 ge ren de cheng ji xu lie wei:$'

DATA ENDS                     ;2位置放80到70的人数，3位置放70到60的人数

CODE SEGMENT                  ;4位置放60以下到人数

ASSUME CS:CODE,DS:DATA

START:MOV AX,DATA

      MOV DS,AX

      MOV CX,11

      LEA SI,BUF

      LEA DI,TJ

RS:   MOV AL,[SI]

      CMP AL,90           ;大于等于90分，跳转到S1

      JGE S1

      CMP AL,80       ;大于等于80分，跳转到S2

      JGE S2

      CMP AL,70           ;大于等于70分，跳转到S3

      JGE S3

      CMP AL,60           ;大于等于60分，跳转到S4

      JGE S4

      MOV BL,[DI+4]       ;小于60分,对应位置加一

      INC BL

      MOV [DI+4],BL

      INC SI

      LOOP RS

      JMP TJWB            ;CX=0，统计结束

S1:  MOV BL,[DI]          ;存放90分以上的人数的存储单元加一

     INC BL

     MOV [DI],BL

     INC SI

     LOOP RS

     JMP TJWB

S2:  MOV BL,[DI+1]

     INC BL

     MOV [DI+1],BL

     INC SI

     LOOP RS

     JMP TJWB

S3:  MOV BL,[DI+2]

     INC BL

     MOV [DI+2],BL

     INC SI

     LOOP RS

     JMP TJWB

S4:  MOV BL,[DI+3]

     INC BL

     MOV [DI+3],BL

     INC SI

     LOOP RS

     JMP TJWB

TJWB:

     LEA DX,INFO1

     MOV AH,9

     INT 21H

     MOV DL,0DH

     MOV AH,2

     INT 21H

     MOV DL,0AH

     MOV AH,2

     INT 21H

     LEA DI,TJ

     MOV CX,5

CTJ: MOV DL,[DI]              ;输出统计的结果，各分数段到人数

     INC DI

     ADD DL,30H

     MOV AH,2

     INT 21H

     MOV DL,20H

     MOV AH,2

     INT 21H

     LOOP CTJ

     MOV DL,0DH

     MOV AH,2

     INT 21H

     MOV DL,0AH

     MOV AH,2

     INT 21H

     MOV CX,11    ;统计完毕，开始排序，CX赋值为11

WXH: MOV SI,11    ;采用选择排序，

     SUB SI,CX    ;SI每次从11-CX开始做循环，当SI<=11时，内循环继续

     MOV BL,BUF[SI]    ;将每次内循环开始的第一个数放在BL中，当作目前最大数

     MOV DX,SI         ;DX用来记录找到的最大数的位置

NXH:

     INC SI            ;内循环每次SI加一

     CMP BL,BUF[SI]    ;比较SI所指向的数与当前的最大数的大小

     JL  GJL           ;若SI所指向的数大于当前的最大数，则将当前最大数换为SI所指向的数

     CMP SI,11         ;SI小于11时内循环继续

     JL  NXH           ;继续执行内循环

     JMP CHUNEI        ;内循环结束，出内循环

GJL: MOV BL,BUF[SI]    ;更改当前最大数

     MOV DX,SI         ;DX用以记录找到最大数的位置

     CMP SI,11         ;若SI小于等于11则继续执行内循环

     JLE NXH

     JMP CHUNEI        ;否则，出内循环

CHUNEI:MOV SI,11        ;找到本次内循环开始的位置

       SUB SI,CX

       XCHG BUF[SI],BL  ;将本次找到的最大数放到本次内循环的开始位置

       MOV DI,DX        ;同时更改原来最大数位置的内容，换为本次循环开始的位置中的数

       XCHG BL,BUF[DI]

       LOOP WXH         ;到外循环

FINISH:LEA SI,BUF

       MOV CX,11

       MOV BX,0AH

       LEA DX,INFO2           ;输出排序后到成绩

       MOV AH,9

       INT 21H

       MOV DL,0DH

       MOV AH,2

       INT 21H

       MOV DL,0AH

       MOV AH,2

       INT 21H

PRINT: MOV AL,BUF[SI]

       MOV AH,00H

       INC SI

       DIV BL

       MOV DH,AH

       MOV DL,AL

       ADD DL,30H

       MOV AH,2

       INT 21H

       MOV DL,DH

       ADD DL,30H

       MOV AH,2

       INT 21H

       MOV DL,20H

       MOV AH,2

       INT 21H

       LOOP PRINT

       MOV AH,4CH

       INT 21H

CODE ENDS

END START

**02**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

对随机输入的 11 名学生的成绩进行排序与分数段人数统计，输出最高 分，最低分，中间值以及排序后的成绩。

内容：定义存储空间存放成绩，输出不及格人数，60~69 分数段人数，……， 90~100 分数段人数，对成绩进行排序，输出排序后的结果，在显示器上显示 如下信息：

The score between 90 and 100 ： ××

……

The score between 0 and 59 ：××

Min is ××

Max is ××

The middle score is ××

Rank ordering : ×× ×× ××……

××表示一个十进制数

**03**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1、输入两个十进制的数，以十六进制输出两者的和，例如，输入3和4，输出07；输入6和7，输出0D；输入8和9，输出11

2、输入两个十六进制的数，以十进制输出两者的和。例如，输入3和5，输出08；输入6和7，输出13；输入9和A，输出19

;输入两个十进制的数，以十六进制输出两者的和，例如，输入3和4，输出07；输入6和7，输出0D；输入8和9，输出11

;输入两个十六进制的数，以十进制输出两者的和。例如，输入3和5，输出08；输入6和7，输出13；输入9和A，输出19

ASSUME CS:CODE,DS:DATA,SS:STACK

DATA SEGMENT

        NUM1 DB 0,0     ;存放第一种运算的两个加数

        NUM2 DB 0,0     ;存放第二种运算的加数

        SUM  DW 0,0     ;存放两个运算的和

        STR1 DB 'Please input two numbers (DEC) :',0AH,0DH,'$'

        SUM1 DB 'The sum is (HEX) :',0AH,0DH,'$'

        STR2 DB 'Please input two numbers (HEX) :',0AH,0DH,'$'

        SUM2 DB 'The sum is (DEC) :',0AH,0DH,'$'

        CRLF DB 0AH,0DH,'$'

DATA ENDS

STACK SEGMENT

        DB 128 DUP(0)

STACK ENDS

CODE SEGMENT

START:

        MOV AX,DATA

        MOV DS,AX

        MOV AX,STACK

        MOV SS,AX

;-------输出提示语句------

        LEA DX,STR1

        MOV AH,9

        INT 21H

INPUT\_DEC:

;-------输入两个十进制数-----

        MOV AH,01H

        INT 21H

        MOV NUM1[0],AL

        SUB NUM1[0],30H

        MOV AH,01H

        INT 21H

        MOV NUM1[1],AL

        SUB NUM1[1],30H

        LEA DX,CRLF

        MOV AH,9

        INT 21H

SUM\_DEC:

        MOV BL,NUM1[0]

        ADD BL,NUM1[1]

        ;SUB BX,30H

        MOV SUM[0],BX  ;将第一个计算结果的ASCII码存放在sum1

PRINTSUM\_HEX:

;-------提示输出答案16进制

        LEA DX,SUM1

        MOV AH,9

        INT 21H

        MOV AX,SUM[0]

        CMP AX,16

        JG  PRINT\_DIGIT      ;比9小，在0-9范围

        CMP AX,10

        JB PRINT\_DIGIT      ;在10-15范围内

        JMP  PRINT\_CHAR     ;比9大，比16小，输出字母

PRINT\_DIGIT:

        MOV BX,16       ;dx=10,除以10

        MOV CX,0        ;cx记录位数，最后输出字符的个数,记录循环次数

        MOV DX,0        ;dx记录余数

S1:                     ;将单个数字依次入栈

        DIV BX          ;ax=ax/16商，dx=ax/16余数

        PUSH DX         ;将余数dx入栈

        INC CX          ;位数加一

        CWD             ;将ax扩展到dx.ax

        CMP AX,0        ;将剩下的数ax与0比较

        JNE S1          ;若ax!=0，继续循环

S2:                     ;输出字符，cx为循环次数，一个数有多少位

        POP DX          ;栈顶的数据出栈，给dx

        ADD DL,30H      ;转化为字符

        MOV AH,02H

        INT 21H         ;dl字符输出

        LOOP S2         ;循环cx

        JMP INPUT\_HEX

PRINT\_CHAR:

        MOV DX,0

        SUB AX,10

        MOV DL,AL

        ADD DL,65

        MOV AH,02H

        INT 21H

        JMP INPUT\_HEX

INPUT\_HEX:

;------输出换行-----

        LEA DX,CRLF

        MOV AH,9

        INT 21H

;------输出提示字符串-----

        LEA DX,STR2

        MOV AH,9

        INT 21H

;-------输入加数-----

        MOV AH,01H

        INT 21H

        MOV NUM2[0],AL

        MOV AH,01H

        INT 21H

        MOV NUM2[1],AL

        LEA DX,CRLF

        MOV AH,9

        INT 21H

        LEA DX,SUM2

        MOV AH,9

        INT 21H

JUDGE PROC NEAR

        MOV CX,2

        MOV SI,0

        MOV BX,0

SUM\_HEX:

;-------比较判断是数字还是字母-----

        MOV AL,NUM2[SI]

        CMP AL,57               ;和9比较

        JG  TO\_DEC              ;是字母，变回数字

SUM\_AGAIN:

        SUB NUM2[SI],30H

AD:     INC SI

        LOOP SUM\_HEX

        JMP DEAL\_SUM

TO\_DEC:

        SUB AL,65

        ADD AL,48

;-------将字母对应的数字扩成对应十进制-----

        SUB AL,38

        SUB AL,32

        MOV NUM2[SI],AL

        JMP AD

    RET

JUDGE ENDP

DEAL\_SUM:

        MOV BL,NUM2[0]

        ADD BL,NUM2[1]

        MOV SUM[2],BX

        JMP PRINT\_DIGIT2

PRINT\_DIGIT2 PROC NEAR

        MOV AX,BX       ;AX存放着sum2

        MOV BX,10       ;dx=10,除以10

        MOV CX,0        ;cx记录位数，最后输出字符的个数,记录循环次数

        MOV DX,0        ;dx记录余数

T1:                     ;将单个数字依次入栈

        DIV BX          ;ax=ax/10商，dx=ax/10余数

        PUSH DX         ;将余数dx入栈

        INC CX          ;位数加一

        CWD             ;将ax扩展到dx.ax

        CMP AX,0        ;将剩下的数ax与0比较

        JNE T1          ;若ax!=0，继续循环

T2:                     ;输出字符，cx为循环次数，一个数有多少位

        POP DX          ;栈顶的数据出栈，给dx

        ADD DL,30H      ;转化为字符

        MOV AH,02H

        INT 21H         ;dl字符输出

        LOOP T2         ;循环cx

        JMP EXIT

    RET

PRINT\_DIGIT2 ENDP

EXIT:

        MOV AX,4C00H

        INT 21H

CODE ENDS

END START

**04**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

使用串操作指令 MOVSB 对一段内存单元中的内容（1，2，3，……，100） 进行转移，再使用串操作指令 CMPS 对转移的内容进行比较来判断传输是否正 确，若不正确则进行重新传输；接着对已经正确传输的 100 个数据进行无符 号型的累加，最后使用 BCD 调整码，最终将答案放入内存，并将其显示在屏幕上。

; 设置数据段

DATA SEGMENT

    Source DB 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100

    Destination DB 100 DUP(0)

    Sum DB 0

DATA ENDS

; 设置代码段

CODE SEGMENT

    ASSUME CS:CODE, DS:DATA, ES:DATA

START:

    MOV AX, DATA

    MOV DS, AX

    MOV ES, AX

; 使用 MOVSB 进行数据传输

    MOV SI, OFFSET Source

    MOV DI, OFFSET Destination

    MOV CX, 100

TransferLoop:

    MOVSB

    LOOP TransferLoop

; 使用 CMPS 进行数据比较

    MOV SI, OFFSET Source

    MOV DI, OFFSET Destination

    MOV CX, 100

CompareLoop:

    CMPSB

    JNE Retransmit

    LOOP CompareLoop

; 对数据进行无符号累加

    MOV SI, OFFSET Destination

    MOV CL, 100

    XOR AH, AH

    XOR AL, AL

AddLoop:

    MOV BL, [SI]

    ADD AL, BL

    INC SI

    LOOP AddLoop

    MOV Sum, AL

; 使用 BCD 调整码

    MOV AH, 0

    MOV AL, Sum

    MOV BX, 10

    DivBcd AH, BX, AL

; 将结果放入内存并显示在屏幕上

    MOV [Sum], AL

    ; 在此处添加代码以将结果显示在屏幕上

; 结束程序

    MOV AH, 4Ch

    INT 21h

CODE ENDS

END START

**05**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

计算 3X-2|Y|+|Z|，并将计算结果输出显示。

data segment

    input db 'x=1,y=2,z=-4',0dh,0ah,'3x+2|y|-|z|=$'

    x db 1h

    y db 2h

    z db -4h

data ends

; stack segment

;     db 200 dup(0)

; stack ends

code segment

    assume ds:data,cs:code;,ss:stack     ;假定每个段的段寄存器，但是需要自己赋值

begin:

    mov ax,data

    mov ds,ax

    ; mov ax,stack

    ; mov ss,ax

    lea dx,input

    mov ah,09h                  ;显示要计算的内容

    int 21H

    mov al,x                    ;将x赋给ax

    add al,x

    add al,x

    cmp y,0

    jge Ly                  ;将y与0比较，大于等于0就跳转到ly，直接相加

    mov bl,0

    sub bl,y                    ;bl为0，减去y就是|y|

    mov y,bl                    ;将bl的值返回给y，继续执行ly

ly:

    add al,y

    add al,y

    cmp z,0

    jge lz                      ;将z与0比较，大于等于0就跳转到lz，直接相加

    mov bl,0

    sub bl,z

    mov z,bl

lz:

    sub al,z

    mov dl,al                   ;将al中的值转移到dl中准备输出

    add dl,30H                  ;dl中的数字加上48，转化为ascii输出

    mov ah,2                    ;显示命令

    int 21H                     ;终止命令

    mov ah,4CH                  ;停止命令

    int 21H

code ends

    end begin

**06**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

题目：分类统计字符个数

内容：程序接受用户从键盘输入的一行字符（字符个数不超过 80 个字符，该 字符串以回车符结束），并按字母、数字及其他字符分类统计个数，然后将相 应的结果存放于 letter、digit 和 other 中，并在显示器上显示如下信息：

The counted result of the program:

letter:××

digit:××

other:××

××表示一个十进制数