实验1

DATA SEGMENT

DATA1 DB ' ','6','8','5','9',0DH,0AH,'$' ;数据1，以字节为单位存放字符对应ASCAL码

DATA2 DB ' ', '4','7','6','4',0DH,0AH,'$'; 0DH是回车，0AH是换行 ，'$'为结束符

DATA ENDS

CODE SEGMENT

  ASSUME CS:CODE,DS:DATA

START:

  MOV AX,DATA ;数据放入AX

  MOV DS,AX   ;数据放入DS

  LEA DX,DATA1 ;将DATA1的有效地址（偏移地址），放入DX寄存器

  MOV AH,09H   ;09号中断，显示DX内容作为首地址的字符串

  INT 21H

  LEA DX,DATA2 ;同上

  MOV AH,09H

  INT 21H

  LEA DI,DATA1 ;DATA1地址放入DI 变址寄存器

  LEA SI,DATA2 ;DATA2地址放入SI

  MOV CX,5 ;下面循环重复5次

BEGIN:

  SUB BYTE PTR [DI],30H ;BYTE PTR [DI],DI里的地址的字符转数字

  SUB BYTE PTR [SI],30H

  INC SI ;+1,处理下一位数据

  INC DI

LOOP BEGIN

  DEC SI ;-1 从最低位开始进行运算

  DEC DI

  MOV CX,5 ;5位，重复5次循环

  CLC ;进位清零

CALADD:

  MOV AL,BYTE PTR [DI] ;把DI存储的地址对应的数字放入AL

  ADC AL,BYTE PTR [SI] ;把SI存储的地址对应的数字放入AL

  AAA ;未组合的BCD调整指令

  MOV BYTE PTR [DI],AL ;两数相加，保存在DI存储的地址

  DEC SI ;下一个高位的运算

  DEC DI

LOOP CALADD

  INC DI ;+1 结果的最高位

  MOV CX,5 ;重复5次

DEALRESULT:

  ADD BYTE PTR [DI],30H ;数字转ASCAL字符

  INC DI ;下一位

LOOP DEALRESULT

  LEA DX,DATA1 ;输出显示结果

  MOV AH,09H

  INT 21H

  MOV AH,4CH ;终止进程，返回

  INT 21H

CODE ENDS

END START

实验2

data segment

;定义提示信息的字符串

MESS1 db 'input string1:',0DH,0AH,'$'

MESS2 db 'input string2:',0DH,0AH,'$'

MESS3 db 'MATCH.',0DH,0AH,'$'

MESS4 db 'NO MATCH.',0DH,0AH,'$'

MAXLEN1 db 10;定义string1

ACTLEN1 db ?  ;自动获得输入的长度

string1 db 9 dup(?),'$'   ;实际存放字符串的地方

MAXLEN2 db 10;定义string2

ACTLEN2 db ?   ;自动获得输入的长度

string2 db 9 dup(?),'$'  ;实际存放字符串的地方

CRLF   DB  0AH, 0DH,'$'     ;换行符

data ends

;定义栈段

stack segment stack 'stack'

STA db 128 DUP(?)

TOP EQU LENGTH STA

stack ends

code segment

ASSUME cs:code,ds:data,es:data,ss:stack

START: ;初始化相关段

        mov ax,data

        mov ds,ax

        mov es,ax

        mov ax,stack

        mov ss,ax

        mov sp,TOP

        ;显示提示输入信息MESS1

        LEA DX,MESS1

MOV AH,09H

INT 21H

        ;接收字符串string1

        LEA DX,MAXLEN1

MOV AH, 0AH

INT 21H

;换行

LEA DX, CRLF

MOV AH, 09H

INT 21H

        ;调用STRSHOW宏显示提示信息MESS2

        LEA DX,MESS2

MOV AH,09H

INT 21H

        ;接收字符串string2

        LEA DX,MAXLEN2

MOV AH, 0AH

INT 21H

LEA DX,CRLF;换行

MOV AH, 09H

INT 21H

;比较2个字符串

        LEA bx,ACTLEN1

        mov ah,[bx] ;ah中存放string1的实际长度

        LEA bx,ACTLEN2

        mov al,[bx] ;al中存放string2的实际长度

        sub ah,al;检查两个字符串长度是否一致

        jnz nomatch

        LEA SI,string1;取出字符串偏移量

        LEA DI,string2

        CLD; DF=0，向内存高地址增加

        CBW ;AL向AH做符号扩展 AX，记录长度

        mov cx,ax ;记录比较次数

        REPZ CMPSB  ;重复按位比较，一样零标志位为0

        jnz nomatch

match:;若匹配则输出提示信息MESS3

        LEA DX,MESS3

MOV AH,09H

INT 21H

        jmp done

nomatch:;若不匹配则输出提示信息MESS4

        LEA DX,MESS4

MOV AH,09H

INT 21H

        jmp done

done:;

        mov ax,4C00H

        int 21H

code ends

END START

实验3

DATA    SEGMENT

MESS    DB      'input 4\*16H: $'

ERROR   DB      'input error',0DH,0AH,'$'

DATA    ENDS

STACK   SEGMENT STACK  ;栈定义

STA     DB      128 DUP (?)

TOP     DW      ?

STACK   ENDS

CODE    SEGMENT

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

START: ;段初始化

  MOV AX,DATA

  MOV DS,AX

  MOV ES,AX

  MOV AX,STACK

  MOV SS,AX

  MOV AX,TOP

  MOV SP,AX

  ;提示输入4个16进制

  MOV AH,09H   ;09H号功能显示字符串

  MOV DX,OFFSET MESS

  INT 21H

  ;函数：键盘输入数据，并转换后数据存在DX中,为输入4位16进制对应的16个二进制

  CALL GETNUM

  ;输出显示，循位移环输出最低位

  MOV CX,16    ;循环16次(二进制位数)

  MOV     BX,DX ;转换结果从DX放入BX

NEXTSHOW:

  ROL     BX,01  ;循环左移1位，由最高位先输出

  MOV DL,BL ;输出的放入DL

  AND DL,01H   ;屏蔽高 7 位

  ADD DL,30H ;数字转字符输出

  MOV AH,02H ;输出DL

  INT 21H       ;显示某位二进制数

  LOOP NEXTSHOW

  ;结束进程，返回控制权限

  MOV     AX,4C00H

  INT     21H

GETNUM  PROC    NEAR  ;键盘接收子程序，2个数值转换2进制存DL中。连续2次，上一次因为左移保留在DH

  XOR     AX,AX

again:  ;AH＝01H，回车或空格结束

  MOV     AH,01H ;01H号功能输入一个字符

  INT 21H        ;键盘接收键盘码

  CMP AL,0DH     ;判回车键

  JZ  done

  CMP AL,20H     ;判空格键

  JZ  done

  ;判为 0 - 9 数字键,满足转换成二进制，存在AL

  CMP AL,30H ;与'0'比较

  JB  error0 ;输入字符<'0',error

  SUB AL,30H ;字符转二进制数字-30H

  CMP AL,0AH     ;并<10,判为 0 - 9 数字键

  JB  GETS ;AL的值已转为数字，ascll码已处理变为2进制数值

  ;判 A - F 字母键

  CMP AL,11H

  JB  error0 ;<'A',error

  SUB AL,07H ;字符转对应二进制数字-30-07H

  CMP AL,0FH     ;判 A - F 字母键

  JBE GETS ;AL的值已成功转为数字，

  ;判 a - f 字母键

  CMP AL,2AH ;与'a'比较

  JB  error0 ;<'a',error

  CMP AL,2FH     ;与'f'比较

  JA  error0 ;>'f',error

  SUB AL,20H ;字符转对应数字-30-07-20H

  ;AL存储的就是输入字符对应的2进制数字，放入DX

GETS:

  MOV     CL,04H ;控制左移次数

  SHL DX,CL      ;逻辑左移4位，空出4位，接上之前的

  ADD DL,AL      ;当前字符对应2进制数加入 DX 中

  JMP again ;输入下一个字符，判断转换

  ;输入错误

error0: PUSH    DX

  MOV AH,09

  MOV DX,OFFSET ERROR

  INT 21H        ;显示输入错误提示信息

  POP DX

  ;返回结果，并换行

done:   PUSH    DX

  ;CALL CRLF

  POP DX

  RET

GETNUM  ENDP

;换行函数

CRLF    PROC    near

MOV DL,0DH

    MOV AH,02H

    INT 21H       ;光标移到第一列

    MOV DL,0AH

    MOV AH,02H

    INT 21H       ;光标移到下一行

    RET

CRLF ENDP

CODE    ENDS

        END     START

实验4 排序实验

DATA0 SEGMENT ;排序数据段

   ORG 20H  ;从20H开始安排数据

   DATA1 DB 0Ch,09h,08h,07h,06h,05h,04h,03h,0Dh,02h ;data

  DB 1ah,19h,18h,07h,06h,05h,04h,03h,01h,12h

  DB 2ah,29h,28h,07h,06h,05h,04h,03h,01h,22h

  DB 3ah,39h,38h,07h,06h,05h,04h,03h,01h,32h

  DB 4ah,49h,48h,07h,06h,05h,04h,03h,01h,42h

DATA0 ENDS

DATAS SEGMENT ;文本数据段

   ORG 100H

   SORTNUM EQU 50   ;改到10显示得下，50题目要求

   MESS1 DB 'INPUT SEGMENT VALUE:$'

   MESS2 DB 'INPUT OFFSET VALUE:$'

   MESS3 DB 'success sort',13,10,'$'

   ERROR DB 'INPUT ERRORt',13,10,'$'

DATAS ENDS

STACK   SEGMENT STACK          ;栈定义

STA     DB      32 DUP (?)

TOP     DW      ?

STACK   ENDS

CODES SEGMENT

    ASSUME CS:CODES,DS:DATAS,SS:STACK

START:

;段初始化

MOV AX,DATAS

    MOV DS,AX  ;DS当前使用存放文本的数据段DATAS

    MOV AX,STACK

    MOV SS,AX

    MOV AX,TOP

    MOV SP,AX

    ;提示输入选择段

    MOV AH,09H          ;09H号功能显示字符串

    MOV DX,OFFSET MESS1

    INT 21H

    ;调用GETNUM，返回值放DL，最多输入4位，返回二进制表示保存在DX

    CALL GETNUM

    ;键入段地址送ES，提示数据位置

    MOV DH,00H ;DH有值清零

    MOV ES,DX

    CALL CRLF ;换行

    ;提示输入偏移值

    MOV AH,09H

    MOV DX,OFFSET MESS2

    INT  21H

    ;调用GETNUM，返回值放DX

    CALL GETNUM

    ;键入偏移量送SI，排序数据段位置

MOV DH,00H ;DH有值清零

    MOV SI,DX;数据开始位置，偏移段地址多少

;DS指向存放排序的数据段data0

mov AX,DATA0

MOV DS,AX

    MOV CX,SORTNUM  ;外循环次数SORTNUM—1送CX

    DEC CX ;cx-1

    XOR BX,BX ;偏移指针BX清0

AGAIN0:

MOV AL,[SI+BX] ;偏移量+偏移指针,从[SI+BX]单元取数送AL

MOV AH,BL     ;AH指向当前选择排序的数的位置，AH=BL

PUSH CX       ;CX入栈，外循环计数保存

MOV CX,SORTNUM   ;计算内循环次数送CX

SUB CX,BX        ;CX-BX,剩余需要排序的数

DEC CX

MOV DX,BX     ;内循环控制变量DX初始化

;选择BX之后（之前的已经排好），最小的数据位置保存在AX

AGAIN1:

INC DX  ;DX值加1

PUSH BX ;保存选择排序内循环开始位置

MOV BX,DX   ;BX指向下一个数据

CMP AL,[SI+BX]  ;比较AL与DX批示单元的值

JBE UUU ;比AL大跳转

MOV AL,[SI+BX] ;DX指示单元的值赋给AL

MOV AH,DL ;并修改指针AH，AH指向最小的位置

UUU:

POP BX

LOOP AGAIN1

;最小值与从前往后遍历的目前位置交换

MOV CL,[SI+BX] ;[SI+BX]旧值暂存CL

MOV [SI+BX],AL  ;[SI+BX]新值得到AL记录的最小值

MOV DL,AH ;DL,记录最小值位置

PUSH BX ;BX入栈

MOV BX,DX ;[SI+BX]变为最小值位置

MOV [SI+BX],CL ;最小值位置赋交换的值

POP BX ;恢复外循环位置

;当前位置完成排序，排后面一个

INC BX   ;偏移指针B加1

POP CX   ;一个循环结束后cx-1

LOOP AGAIN0

    PUSH DS  ;保存排序数据段位置

    MOV AX,DATAS ;临时换为输出数据段

    MOV     DS,AX

     ;输出完成排序

    MOV     AH,09H          ;09H号功能显示字符串

    MOV     DX,OFFSET MESS3

    INT     21H

    POP DS ;恢复排序数据段位置

    MOV CX,SORTNUM

    ;显示输出

AGAIN2:

MOV AL,[SI]

    CALL DISP1 ;循环调DISP1，将排序后的数依次显示出

    CALL CRLF

    INC SI

LOOP AGAIN2

;结束进程，返回控制权限

    MOV     AX,4C00H

    INT     21H

DISP1 PROC    near   ;输出2位16进制数

PUSH CX

MOV BL,AL ;AL当前要输出的数

MOV DL,BL ;DL存当前要输出的数

MOV CL,04

ROL DL,CL ;DL循环左移4位

AND DL,0FH ;DL保留低位

CALL DISPL ;输出第1个数字位

MOV DL,BL ;DL存当前要输出的数

AND DL,0FH ;

CALL DISPL ;输出第二个数字位

POP CX

RET

DISP1 ENDP

DISPL PROC    near   ;转字符，DL低四位保存需要输出的数

ADD DL,30H

CMP DL,3AH

JB DDD

ADD DL,27H

DDD:

MOV AH,02H

INT 21H

RET

DISPL ENDP

;换行函数

CRLF    PROC    near

MOV DL,0DH

    MOV AH,02H

    INT 21H       ;光标移到第一列

    MOV DL,0AH

    MOV AH,02H

    INT 21H       ;光标移到下一行

    RET

CRLF ENDP

GETNUM  PROC    NEAR  ;键盘接收子程序，2个数值转换2进制存DL中。连续2次，上一次因为左移保留在DH

  XOR     AX,AX

again:  ;AH＝01H，回车或空格结束

  MOV     AH,01H ;01H号功能输入一个字符

  INT 21H        ;键盘接收键盘码

  CMP AL,0DH     ;判回车键

  JZ  done

  CMP AL,20H     ;判空格键

  JZ  done

  ;判为 0 - 9 数字键,满足转换成二进制，存在AL

  CMP AL,30H ;与'0'比较

  JB  error0 ;输入字符<'0',error

  SUB AL,30H ;字符转二进制数字-30H

  CMP AL,0AH     ;并<10,判为 0 - 9 数字键

  JB  GETS ;AL的值已转为数字，ascll码已处理变为2进制数值

  ;判 A - F 字母键

  CMP AL,11H

  JB  error0 ;<'A',error

  SUB AL,07H ;字符转对应二进制数字-30-07H

  CMP AL,0FH     ;判 A - F 字母键

  JBE GETS ;AL的值已成功转为数字，

  ;判 a - f 字母键

  CMP AL,2AH ;与'a'比较

  JB  error0 ;<'a',error

  CMP AL,2FH     ;与'f'比较

  JA  error0 ;>'f',error

  SUB AL,20H ;字符转对应数字-30-07-20H

  ;AL存储的就是输入字符对应的2进制数字，放入DX

GETS:

  MOV     CL,04H ;控制左移次数

  SHL DX,CL      ;逻辑左移4位，空出4位，接上之前的

  ADD DL,AL      ;当前字符对应2进制数加入 DX 中

  JMP again ;输入下一个字符，判断转换

  ;输入错误

error0: PUSH    DX

  MOV AH,09

  MOV DX,OFFSET ERROR

  INT 21H        ;显示输入错误提示信息

  POP DX

  ;返回结果，并换行

done:   PUSH    DX

  ;CALL CRLF

  POP DX

  RET

GETNUM  ENDP

CODES ENDS

END START

实验5 BIOS

DATA SEGMENT

DATA ENDS

STACK SEGMENT STACK 'STACK'

STA DB 128 DUP(?)

TOP EQU LENGTH STA

STACK ENDS

CODE SEGMENT

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

START:

;初始化相关寄存器

  MOV AX,DATA

MOV DS,AX

MOV AX,STACK

MOV SS,AX

MOV SP,TOP

MOV BL,0;BL中存放要打印字符的ASCII码

MOV DH,0;设置行号

OUTER:

MOV DL,0;设置列号

INNER:

;输出ASCLL表的一个字符

MOV AH,02H ;BH＝0显示页码,DH＝行(Y坐标),DL＝ 列(X坐标)

INT 10H;调用BIOS中断设置光标位置

CMP BL,0AH;判断是否为换行符

JZ CHANGE

CMP BL,0DH;判断是否为回车符

JZ CHANGE

MOV AL,BL;将BL->AL用于显示

SHOW:

;输出字符

MOV CL,1;设置输出字符字数

MOV AH,0AH ;AL=字符，BH=页码，CX=多次打印字符

INT 10H;调用BIOS中断输出字符

;输出间隔的空格

INC DL;列号+1

INC BL;下一个字符ASCII码

MOV AH,02H;设置光标位置

INT 10H

MOV AL,0

MOV AH,0AH

INT 10H;输出空格间隔

;输出32个为一行，总共16行

INC DL;列号+1

CMP DL,32;判断一列是否完成

JNZ INNER

INC DH

CMP DH,16;判断行数是否达标

JNZ OUTER

MOV AX,4C00H

INT 21H;结束程序

CHANGE:;若为特殊字符则将其替换为0

        MOV AL,0

JMP SHOW

CODE ENDS

END START

实验5 DOS

DATA SEGMENT

DATA ENDS

STACK SEGMENT STACK 'STACK'

STA DB 128 DUP(?)

TOP EQU LENGTH STA

STACK ENDS

CODE SEGMENT

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

START:

;初始化相关寄存器

    MOV AX,DATA

MOV DS,AX

MOV AX,STACK

MOV SS,AX

MOV SP,TOP

;置行、列计数初值,字符0开始

MOV BL,0;BL中存放要打印字符的ASCII码

MOV CX,16;设置外层循环次数，即行数

OUTER:

PUSH CX

MOV CX,16;设置内层循环次数，即列数

INNER:

CMP BL,0AH;判断是否为换行符

JZ CHANGE

CMP BL,0DH;判断是否为回车符

JZ CHANGE

MOV DL,BL;显示码→DL

SHOW:

;输出DL里的内容

MOV AH,2

INT 21H ;int 21 02号功能，输出值放DL

;输出空格间隔

MOV DL,0

INT 21H;输出空格间隔

INC BL;下一个字符ASCII码

LOOP INNER

CALL CRLF;换行函数

POP CX

LOOP OUTER

MOV AX,4C00H;退出程序

INT 21H

CHANGE:;是控制码则将其替换为0

MOV DL,0

JMP SHOW

;换行函数

CRLF    PROC    NEAR

MOV DL,0DH

    MOV AH,02H

    INT 21H       ;光标移到第一列

    MOV DL,0AH

    MOV AH,02H

    INT 21H       ;光标移到下一行

    RET

CRLF ENDP

CODE ENDS

END START

泰戈尔在诗中说，天空没有翅膀的影子，但我已飞过，艾青对朋友说，也许有人到达不了彼岸，但我们共同拥有大海。也许你们没有显赫的成绩，但运动场上留下了你们的足迹。也许你们没有奖品，但我们心中留下了你们拼搏的身影。所有的努力都是为了迎接这一刹那，所有的拼搏都是为了这一声令下。

206001829 计算机科学与技术1班 罗宇涛