**1. 编一个程序，实现从键盘输入一个长度不超过300个字符的串（以回车键作为输入结束），然后在下一行以倒序输出所输入的字符。**

DATAS SEGMENT

string db 301 dup('$')

ctrl db 0ah,0dh,'$'

DATAS ENDS

STACKS SEGMENT

;此处输入堆栈段代码

STACKS ENDS

CODES SEGMENT

ASSUME CS:CODES,DS:DATAS,SS:STACKS

START:

MOV AX,DATAS

MOV DS,AX

lea bx,string

Mov cx,0

lp1:mov ah,1

int 21h

cmp al,0dh

je lp2

mov [bx],al

inc bx

Inc cx

jmp lp1

lp2:dec bx

mov dl,[bx]

mov ah,2

int 21h

loop lp2

exit:MOV AH,4CH

INT 21H

CODES ENDS

END START

**2. 编写一个程序，从键盘输入一个0～65535之间的10进制无符号数，然后以16进制形式显示出所输入的数。**

data segment

string db 0dh

num db 4 dup(?),'H$'

tab db '0123456789ABCDEF'

data ends

code segment

assume cs:code,ds:data

begin: mov ax,data

mov ds,ax ;初始化代码段

xor bx,bx

mov cx,10 ;进制

next: mov ah,1

int 21h

cmp al,0dh

jz conv ;如果是回车，表示输入结束，转换开始

push ax ;保存输入值，当然还有AH,因为堆栈的存取必须以字为单位

mov ax,bx

mul cx

mov bx,ax ;将先前的结果向上推一位

pop ax ;取回本次输入

and al,0fh ;屏蔽掉无用位，类SUB AL,30H

xor ah,ah ;高位归零

add bx,ax ;合并本次输入

jmp next

conv: mov ax,bx ;开始转换

mov ch,4

lea bx,tab ;bx直接定址表

mov cl,4

lea si,num

lopa: rol ax,cl ;把高4位移到低4位

push ax

and ax,000fh ;取出低4位

xlat

mov [si],al ;按地址由低到高的顺序将结果由高到底存放

inc si

pop ax

dec ch

jnz lopa

lea dx,string

mov ah,9

int 21h ;回车换行

mov ah,4ch

int 21h

code ends

end begin

**3. 编写一个程序，计算 1 + 2 + 3 + 4 + … + 100的累加和，并把累加和以2进制形式显示出来（要使用循环累加方法，不能使用公式S=N\*(N+1)/2=50\*101=5050）。**

DATAS SEGMENT

sum dw 0

DATAS ENDS

STACKS SEGMENT

;此处输入堆栈段代码

STACKS ENDS

CODES SEGMENT

ASSUME CS:CODES,DS:DATAS,SS:STACKS

START:

MOV AX,DATAS

MOV DS,AX

mov ax,0

mov bx,1

lp1:

add ax,bx

inc bx

cmp bx,101

je next

jmp lp1

next:

mov cx,16

mov bx,ax

lp3:

shl bx,1

jc lp2

mov ah,2

mov dl,0

add dl,30h

int 21h

dec cx

cmp cx,0

je exit

jmp lp3

lp2:

mov ah,2

mov dl,1

add dl,30h

int 21h

dec cx

cmp cx,0

je exit

jmp lp3

exit:MOV AH,4CH

INT 21H

CODES ENDS

END START

**已知数据块BUFA中存放15个互不相同的字数据，BUFB中存放20个互不相同的字数组，编一个程序，将既在BUFA中出现，又在BUFB中出现的数据存放到BUFC开始的缓冲区中。**

DATAS SEGMENT

BUFA db 1,3,5,7,9,21,21,21,21,21,21,21,21,21,21

count1 dw $-BUFA

BUFB db 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20

count2 dw $-BUFB

BUFC db 15 dup(?)

DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES SEGMENT

ASSUME CS:CODES,DS:DATAS,SS:STACKS

START:

MOV AX,DATAS

MOV DS,AX

lea di,BUFC

mov cx,count1

mov bx,0

L:

push cx

mov al,BUFA[bx]

inc bx

mov cx,count2

lea si,BUFB

L2:

mov dl,[si]

inc si

cmp al,dl

jz L3

loop L2

jmp L4

L3:

mov [di],dl

inc di

L4:

pop cx

loop L

MOV AH,4CH

INT 21H

CODES ENDS

END START

**5. LINK指向一个字线性表，其中的首单元保存线性表的长度，编一个程序，将该线性表中的0FFFFH项删除。**

DATAS SEGMENT

link dw 7,0ffffh,2,3,4,5,6,8

buffer dw 7 dup(0)

DATAS ENDS

CODES SEGMENT

ASSUME CS:CODES,DS:DATAS

START:

MOV AX,DATAS

MOV DS,AX

MOV BX,0

LEA SI,LINK

MOV CX,[SI]

LEA DI,BUFFER+2

L: ADD SI,2

MOV AX,[SI]

CMP AX,0FFFFH

JNZ CUN

LOOP L

JMP FUZHI

CUN:MOV [DI],AX

ADD DI,2

INC BX

LOOP L

FUZHI:

LEA SI,BUFFER

MOV [SI],BX

LEA DI,LINK

MOV CX,BX

L2: ADD SI,2

ADD DI,2

MOV AX,[SI]

MOV [DI],AX

LOOP L2

MOV AH,4CH

INT 21H

CODES ENDS

END START

**编写一个程序，从键盘输入一个不大于16位的2进制数，然后以8进制形式显示出所输入的数。**

DATA SEGMENT

x dw 0ah,0dh,'$'

InputBuffer db 17,0,17 dup (0)

UserInput dw 0

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START:

MOV AX,DATA

MOV DS,AX

call GetUserInput ;输入子程序

mov UserInput,ax ;取得输入值16进制,也是2进制

lea dx,x

mov ah,9

int 21h

mov bx,UserInput ;取得输入值

mov cl,3 ;移位数

mov ch,0 ;计数

next:

mov ax,bx

and ax,0000000000000111b ;取低3(111)位,即八进位

or al,'0' ;转ascii

push ax ;保存

inc ch ;计数

shr bx,cl ;左移3位,即下一位

or bx,bx ;是否完了

jnz next ;未

mov cl,ch ;取得计数

mov ch,0 ;清0

next1:

pop dx ;弹出dl

mov ah,2 ;印出结果

int 21h

loop next1

mov ah,7 ;暂停

int 21h

quit:

MOV AH,4CH ;带返回码结束功能

INT 21H ;返回系统

;-----------------------

GetUserInput:

mov dx,offset InputBuffer ;指向输出缓冲

mov ax,0c0ah ;输入函数,先清空键盘缓冲

int 21h

mov si,offset InputBuffer + 2

mov cx,0

mov cl,[si-1] ;取实际输入数

jcxz GetUserInput ;无输入

Call GetValue ;取值子程序,ax传回该值,cf=1表示错误,可能输入非数字

jc GetUserInput ;输入错误,非数字

ret

;--------------------------------------------------------------------------

;转值子程序,把输入的字符转成16进制(2进制)

;输入:ds:si数字字符串起点,以0dh结束

;输出:ax=转换后的16进制值,cf=1表示有非数字字符

GetValue:

push bx

push di

xor bx,bx

mov di,10

GetV10:

lodsb ;指向起点

mov ah,0 ;清除

cmp al,0dh ;回车?

jz Getvx

cmp al,'0' ;以下比较0-1,否则cf=1离开

jb Getvy

cmp al,'1'

ja Getvy

sub al,'0' ;ascii -> 值

shr al,1 ; cf=1,1 cf=0,0

rcl bx,1 ;一位接一位的旋入bx

jmp short GetV10

Getvx: mov ax,bx ;ascii转值后由ax转回

clc ;成功cf=0

jmp short Getvz

Getvy:

stc ;错误 cf=1

Getvz:

pop di

pop bx

ret

;------------------------------------

CODE ENDS

END START

**以下面形式定义一个长整数，其所占用字节数由N得到，比如128位的数5746352413DE89674523BC9A78563412H定义成：**

**NUM DB 12H,34H,56H,78H,9AH,0BCH,23H,45H**

**DB 67H,89H,0DEH,13H,24H,35H,46H,57H**

**N DB $-NUM**

**试编写一个程序，实现对N字节的NUM的值求补，并把求补后的值仍保存在NUM中。**

DATAS SEGMENT

NUM DB 12H,34H,56H,78H,9AH,0BCH,23H,45H

DB 67H,89H,0DEH,13H,24H,35H,46H,57H

N DW $-NUM

DATAS ENDS

STACKS SEGMENT

STACKS ENDS

CODES SEGMENT

ASSUME CS:CODES,DS:DATAS,SS:STACKS

START:

MOV AX,DATAS

MOV DS,AX

lea si,NUM

mov cx,N

stc ;将CF置1

L:

mov al,[si]

not al ;求补即是取反+1（not为取反）

adc al,0

mov [si],al

inc si

loop L

MOV AH,4CH

INT 21H

CODES ENDS

END START

1. **编写一个程序，将内存地址0B800H:0开始的32KB内容保存在文件VRAM.SCN中。（不考虑文件处理过程中的错误）**

DATA SEGMENT

PATHNM DB'VRAM.SCN',0

HANDLE DW 0

TIP1 DB 'ERROR OPEN!$'

TIP2 DB 'ERROR WRITE!$'

TIP3 DB 'OK!$'

DATA ENDS

STACK SEGMENT STACK

DB 100H DUP(?)

STACK ENDS

CODE SEGMENT

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

MAIN PROC NEAR

START:

MOV AX,DATA

MOV DS,AX

LEA DX,PATHNM

xor cx,cx ;normal

mov ax,3c00h

INT 21H

JC ERROR1

PUSH DS ;save

MOV HANDLE,AX

MOV BX,AX

MOV DX,0B800H

MOV DS,DX

XOR DX,DX ;b800:0

MOV CX,8000H

MOV AH,40H

INT 21H

POP DS ;restore

JC ERROR2

JMP OK

ERROR1:

MOV AH,09H

LEA DX,TIP1

INT 21H

JMP EXIT

ERROR2:

MOV AH,09H

LEA DX,TIP2

INT 21H

JMP EXIT

OK:

MOV AH,09H

LEA DX,TIP3

INT 21H

MOV BX,HANDLE

MOV AH,3EH ;close file

INT 21H

EXIT:

MOV AH,4CH

INT 21H

MAIN ENDP

CODE ENDS

END MAIN