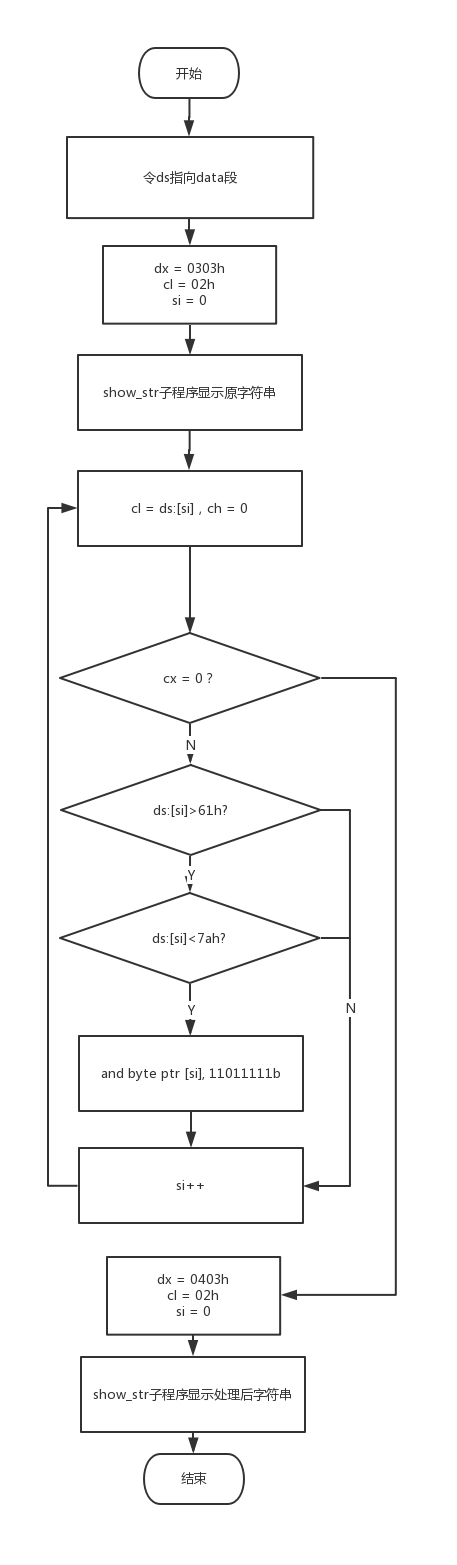
1. ds指向data段，dx = 0303h（第三行第三列），cl = 02h（黑底绿字），si = 0，调用show\_str子程序；
2. Show\_str子程序流程思想在实验十子程序一中，参数为dx，cl，ds:[si]指向一行字符串，功能为将一行字符串到不为0的位置之前全部显示在屏幕上；
3. 令si = 0，调用letterc子程序；
4. 进入letterc子程序；
5. Letterc子程序主要就是将一个字符串中，16进制ascii码在61h到7ah之间的小写字母与11011111b相与，使小写字母转变为大写字母。首先令cl = ds:[si]，令ch = 0，判断cx是否为0，为0跳转至（7）。不为0与61h比较，小于61h跳转至（6），否则与7ah比较，大于7ah跳转至（6），否则做与操作；
6. si增加，跳转回（5）；
7. 返回（8）；
8. 令dx = 0403h（第四行第三列），cl = 02h，si = 0，调用show\_str子程序；
9. 返回（10）；
10. 程序结束。



源程序：

assume cs:codesg

datasg segment

db "Beginner's All-purpose Symbolic Instruction Code.",0

datasg ends

codesg segment

begin: mov ax,datasg

mov ds,ax

mov dx,0703h

mov cl,02h

mov si,0

call show\_str

mov si,0

call letterc

mov dx,0803h

mov cl,02h

mov si,0

call show\_str

mov ax,4c00h

int 21h

letterc:mov cl,[si]

mov ch,0

jcxz ok

cmp byte ptr [si],61h

jb x1

cmp byte ptr [si],7Ah

ja x1

and byte ptr [si],11011111b

x1: inc si

jmp short letterc

ok: ret

show\_str:push dx

push cx;保存寄存器

mov ax,0b800h

mov es,ax;显存位置

mov al,0A0h

mul dh;计算出行的位置

mov bx,ax

mov al,2h

mul dl;计算出列的位置

add bx,ax

mov di,0;显存寄存器寻址

s: mov bp,cx

mov ch,0h

mov cl,ds:[si]

jcxz back;判断当前字符

mov ah,ds:[si]

mov es:[bx+di],ah;保存字符

inc di

mov cx,bp

mov al,cl

mov es:[bx+di],al;保存字符属性

inc si

inc di;向后读取

jmp s

back: pop cx

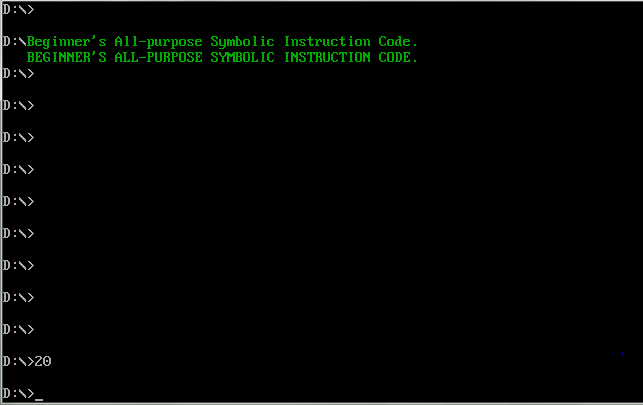
pop dx

ret

codesg ends

end begin

程序运行结果：



功能描述：

程序运行后可以在屏幕上第3行第3列显示出来原字符串，第4行第3列显示出来将原字符串中所有小写字母变为相应的大写字母的字符串。