程序流程图思想：

首先设定bx开始指向num（原数字数组），bp指向asis（第一辅助数组），si = 0，di = 0

程序主要有两个子程序组成.

Num:123 12666 1 8 3 38

第一个子程序功能为将16进制数字转化为10进制字符并且保存到一个数组当中，一共有六个数字，循环六次,ds:[bp+di]位置作为辅助保存数组。

Dtoc保存bx地址（push bx）、cx循环数（push cx）、push 0

1.：判断当前数字是不是为0，为0则跳出程序，否则开始2

2.取当前数字除以十的余数，并且将余数dl加上30h，push压栈dx，循环1

3.出栈到cx中，判断cx是不是为0，为0则跳到4，否则执行pop cx，ds:[bp+di] = cx，di++，循环3

4.ds:[bp+di] = 20h，pop cx、pop bx返回

Asis:123 12666 1 8 3 38 000000000

第二个子程序功能为显示字符串，即show\_str

Trans：令dx = 0803、指定行列，di = 0，回到字符串串首位置

1. 首先es = 0b800h，bx = dh乘以0a0h计算行数，bx += dl \* 02h计算列数，令si = 0，由es:[bx+si]来指定显存位置。
2. 判断ds:[bp+di]是否为0，为0则跳出循环，否则Es:[bx+si] = ds:[bp+di]，si++，Es:[bx+si] = 02h（绿色字），si++，di++，循环2

跳回主程序

assume cs:code

data segment

num dw 123,12666,1,8,3,38

asis dw 16 dup (0)

data ends

stack segment

dw 16 dup (0)

stack ends

code segment

start: mov ax,data

mov ds,ax

mov ax,stack

mov ss,ax

mov sp,10h

lea bx,num

lea bp,asis

mov si,0;si和bx共同推进num数组

mov di,0;di和bp共同推进asis数组

mov cx,6

s: mov ax,ds:[bx+si]

call dtoc;调用计算数组

add si,2h

loop s

mov dx,0803h

mov di,0

call show\_str

mov ax,4c00h

int 21h

dtoc: push bx

push cx

mov dx,0h

push dx

s2: mov cx,ax;判断x是不是等于0

jcxz back1;等于0返回

mov dx,0

mov bx,0ah

div bx;每次除以10

add dl,30h;ascii码变换

push dx

jmp s2

back1: pop cx

jcxz back2

mov ds:[bp+di], cx

inc di

jmp back1

back2: mov byte ptr ds:[bp+di],20h

inc di

pop cx

pop bx

ret

show\_str: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

s5: mov ch,0h

mov cl,ds:[bp+di]

jcxz back3;判断当前字符

mov ah,ds:[bp+di]

mov es:[bx+si],ah;保存字符

inc si

mov al,02h

mov es:[bx+si],al;保存字符属性

inc si

inc di;向前读取

jmp s5

back3: ret

code ends

end start

程序执行结果：

