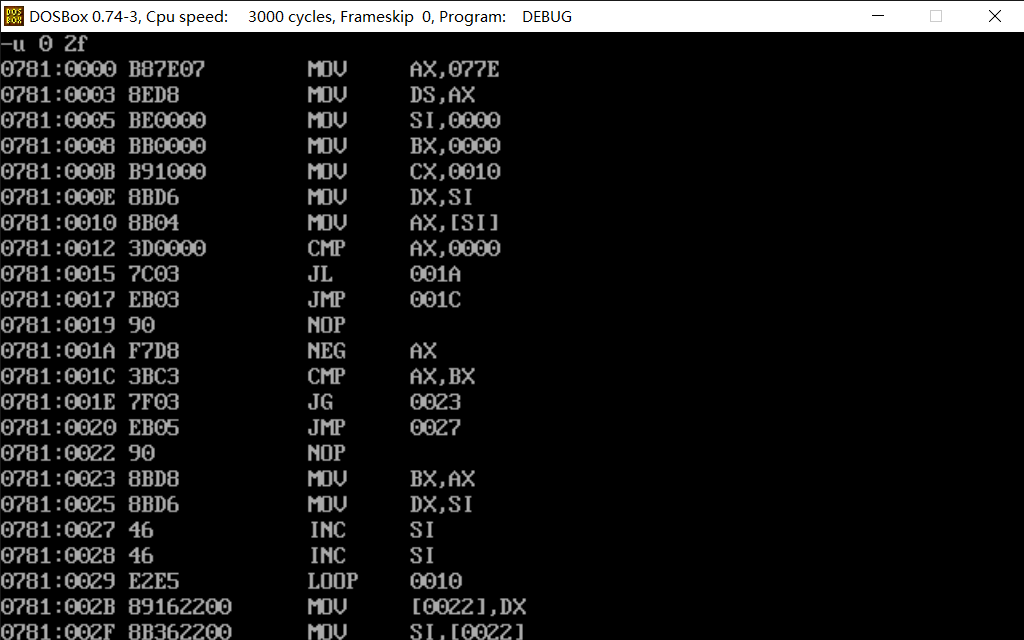
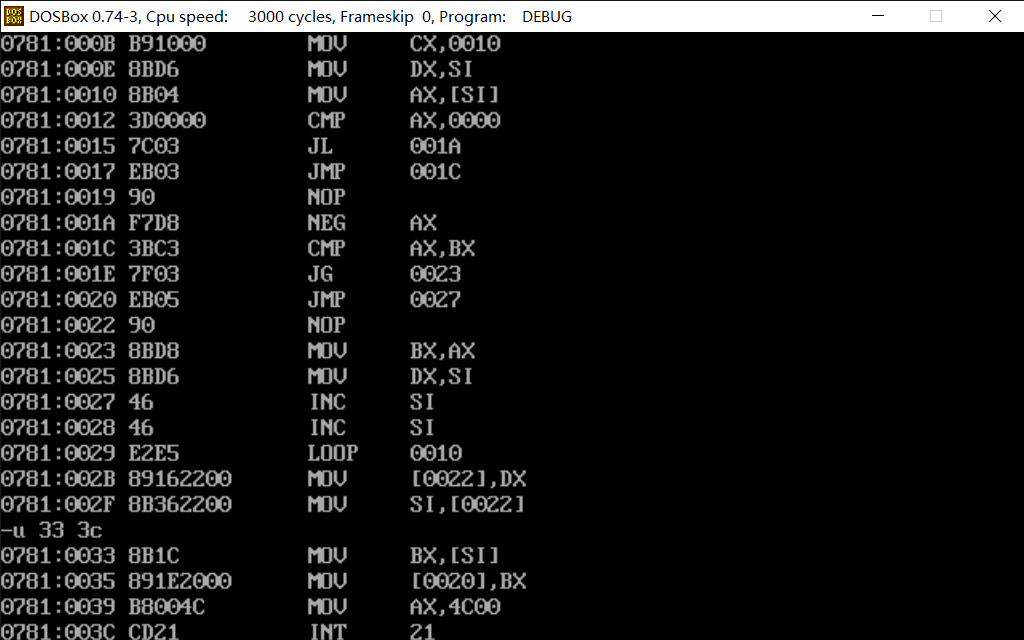
第2次上机

|  |  |  |
| --- | --- | --- |
| 班级 | 学号 | 姓名 |
|  |  |  |

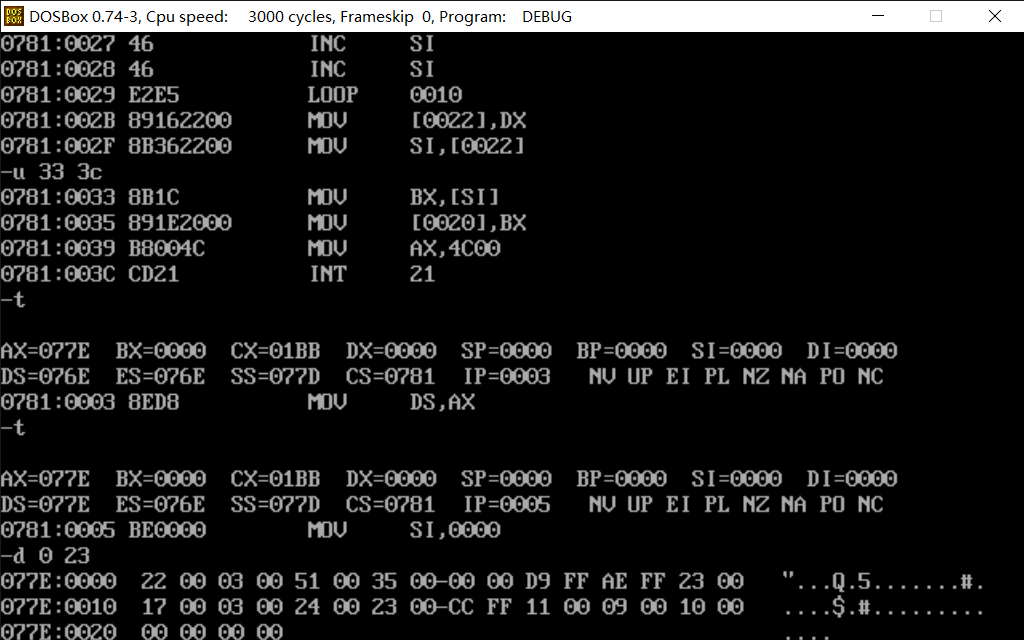
1、 循环程序设计

（1）反汇编的截图

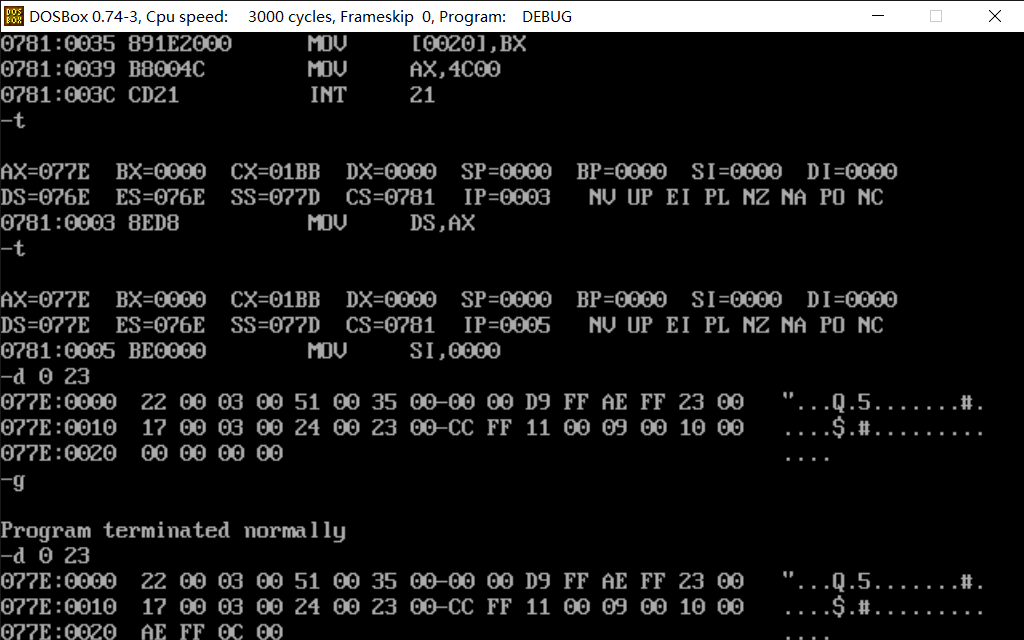




（2）在进行计算前，显示数组M开始的n+2个字的内存值的截图（只能显示这n+2个字的内存值，多显示、少显示均扣分）



（3）执行完计算后，显示数组M开始的n+2个字的内存值的截图（只能显示这n+2个字的内存值，多显示、少显示均扣分）



（4）源代码

title Max\_abs

data segment

M dw 22h, 03h, 51h, 35h, 00h, 0ffd9h, 0ffaeh, 23h, 17h, 03h, 24h, 23h, 0ffcch, 11h, 09h, 10h

n equ ($-M)/2

max\_num dw ?

offset\_address dw ?

data ends

code segment

assume cs:code, ds:data

main proc

; assign the data segment base address to DS

mov ax, data

mov ds, ax

; TODO ...

; | add your code between arrows |

; v ---------------------------- v

mov si, offset M

mov bx, 0 ;用于存储当前的最大值

mov cx, n ;数组大小

mov dx, si ;用于存储当前最大值的偏移地址

start: mov ax, [si]

cmp ax, 0

jl negative

jmp positive

negative: neg ax

positive: cmp ax, bx

jg change

jmp lop

change: mov bx, ax

mov dx, si

lop: inc si

inc si

loop start

result: mov offset\_address, dx

mov si, offset\_address

mov bx, [si]

mov max\_num, bx

; ^ ---------------------------- ^

; | The END |

; method 2: return to dos

mov ax, 4c00h

int 21h

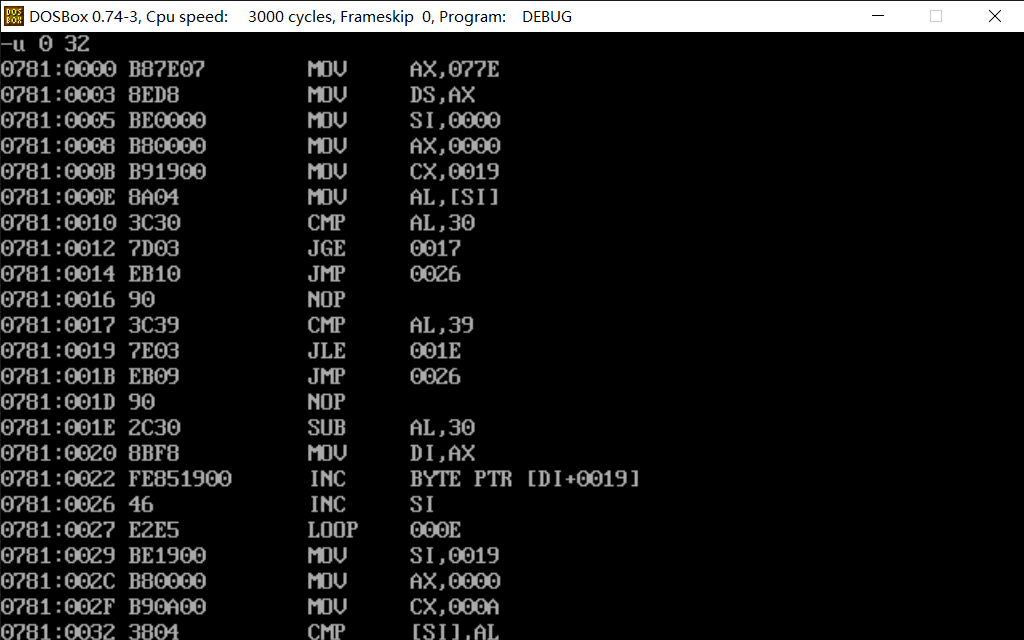
main endp

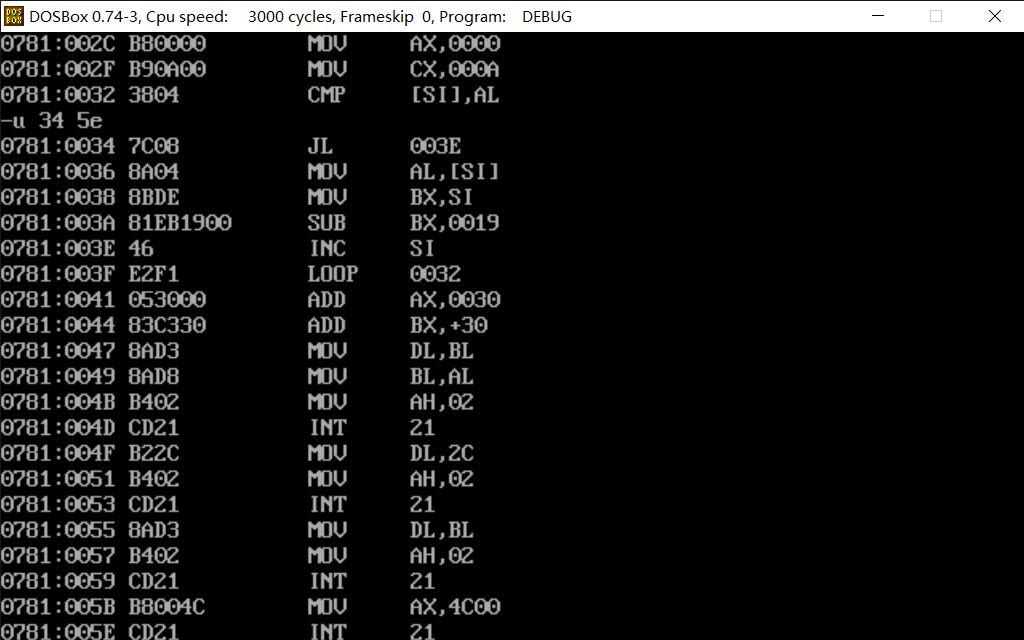
code ends

end main

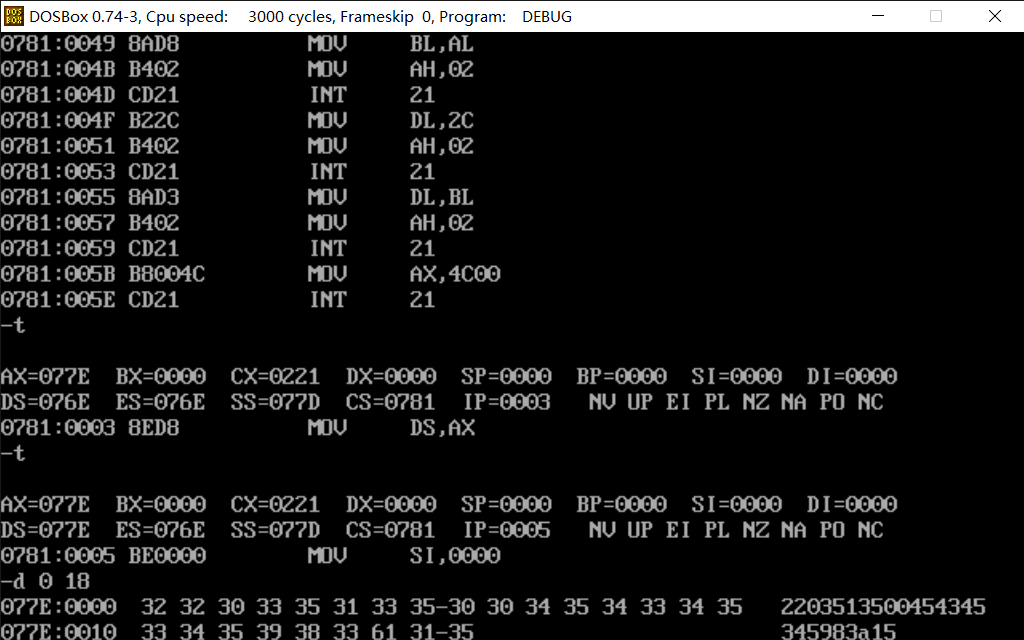
2、 分支程序设计

（1）反汇编的截图

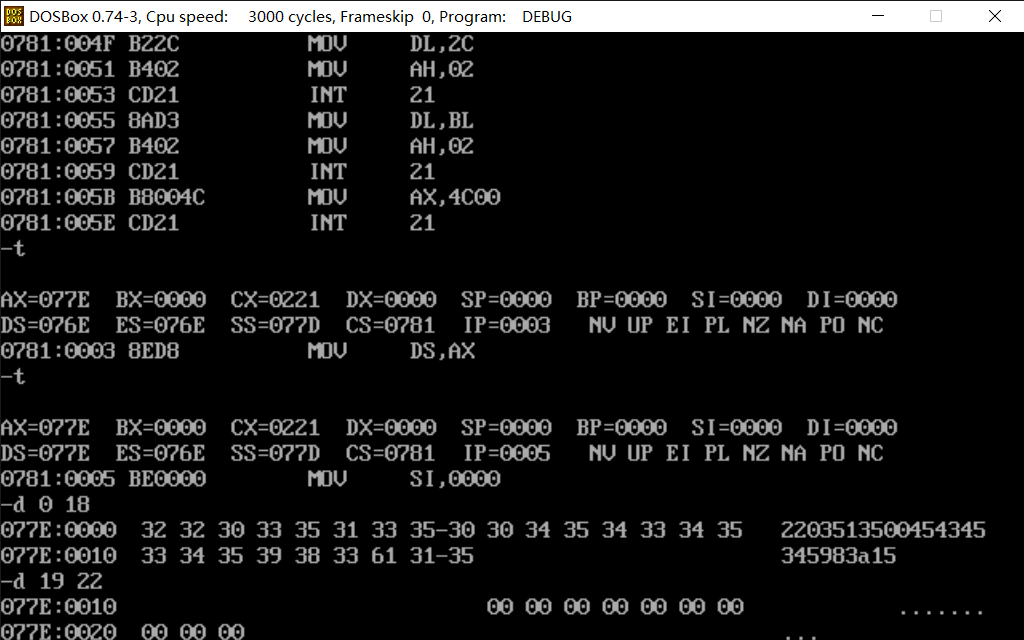




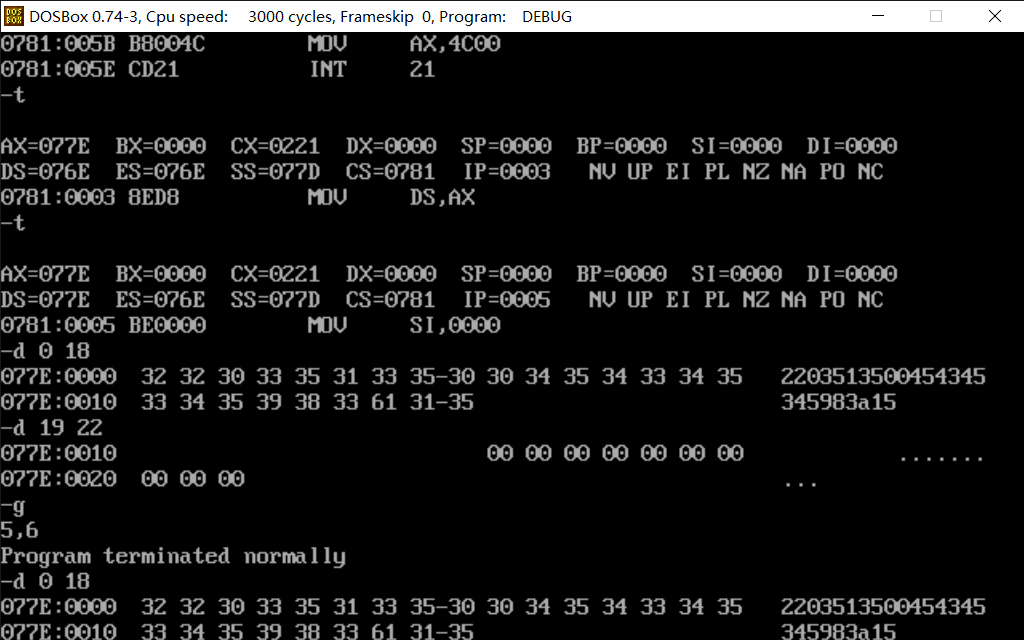
（2）在进行计算前，显示在数据段中定义的含学号的字符串的内存值的截图（只能显示该完整的字符串，多显示、少显示均扣分）



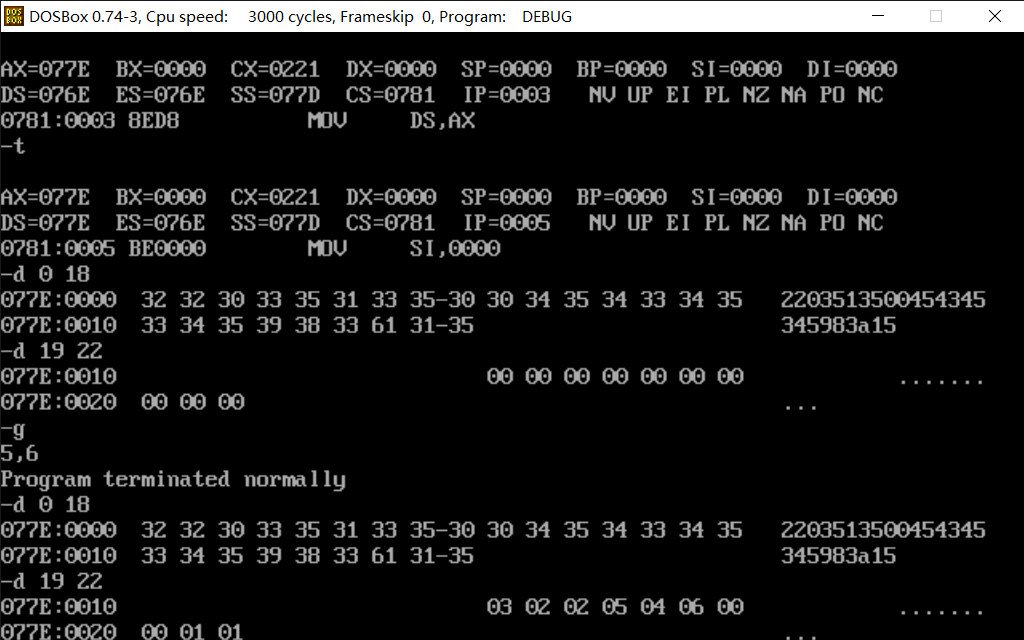
（3）在进行计算前，显示在数据段中定义的COUNT数组的内存值的截图（只能显示完整的COUNT数组内容，多显示、少显示均扣分）



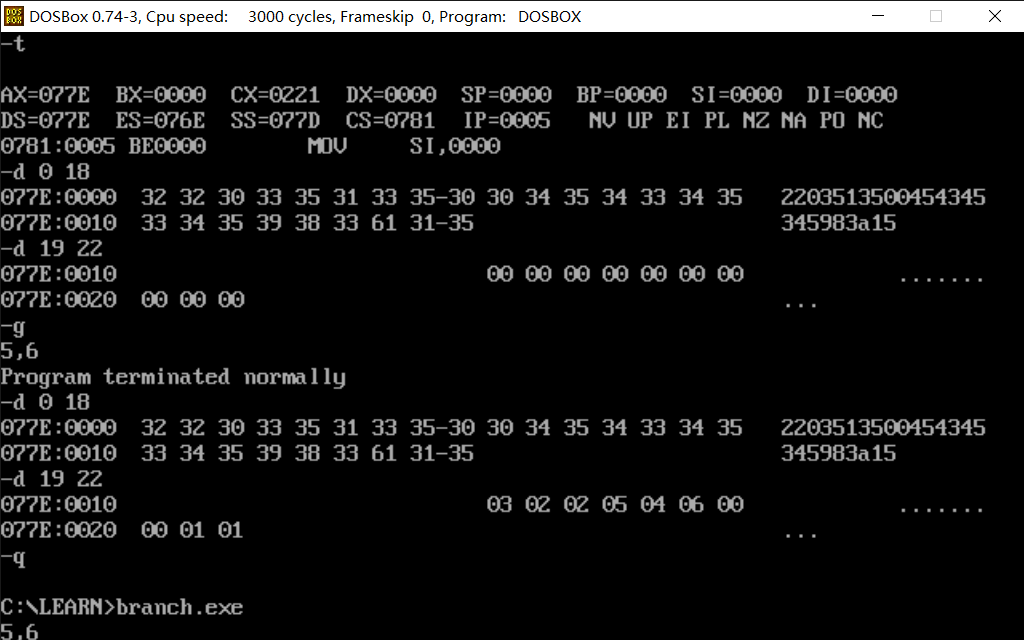
（4）执行完计算后，显示在数据段中定义的含学号的字符串的内存值的截图（只能显示该完整的字符串，多显示、少显示均扣分）



（5）执行完计算后，显示在数据段中定义的COUNT数组的内存值的截图（只能显示完整的COUNT数组内容，多显示、少显示均扣分）



（6）程序在DOSBox下直接运行的截图



（7）源代码

title count

data segment

string db '2203513500454345345983a15'

n equ $-string

count db 10 dup(0)

data ends

code segment

assume cs:code, ds:data

main proc

; assign the data segment base address to DS

mov ax, data

mov ds, ax

; TODO ...

; | add your code between arrows |

; v ---------------------------- v

mov si, offset string

mov ax, 00h

mov cx, n ;字符数组大小

start: mov al, [si] ;用于存储字符

judge1: cmp al, 30h

jge judge2

jmp lop

judge2: cmp al, 39h

jle counting

jmp lop

counting: sub al, 30h

mov di, ax

inc count[di] ;递增相应的数组元素计数值

lop: inc si

loop start

mov si, offset count

mov ax, 00h

mov cx, 0ah ;COUNT数组大小

compare: cmp [si], al

jl next

mov al, [si] ;存储最大计数值

mov bx, si

sub bx, offset count

;存储最大计数值对应的COUNT数组下标

next: inc si

loop compare

print: add ax, 30h ;ASCII码转换

add bx, 30h

mov dl, bl

;由于打印完字符后，al<——dl,故将al的值保存在bl里

mov bl, al

mov ah, 2

int 21h

mov dl, 2ch ;打印逗号

mov ah, 2

int 21h

mov dl, bl

mov ah, 2

int 21h

; ^ ---------------------------- ^

; | The END |

; method 2: return to dos

mov ax, 4c00h

int 21h

main endp

code ends

end main