第 2 次上机

班级	学号	姓名
计试 2201	2223312202	林圣翔

1、 循环程序设计

（1）反汇编的截图

TODO：你的截图

```
0777:0000 B87607        MOV     AX,0776
0777:0003 8ED8          MOV     DS,AX
0777:0005 B90400        MOV     CX,0004
0777:0008 8D3E0000      LEA     DI,[0000]
0777:000C 8B1D          MOV     BX,[DI]
0777:000E 83FB00        CMP     BX,+00
0777:0011 7302          JNB     0015
0777:0013 F7DB          NEG     BX
0777:0015 391E0A00      CMP     [000A],BX
0777:0019 7308          JNB     0023
0777:001B 891E0A00      MOV     [000A],BX
0777:001F 893E0C00      MOV     [000C],DI
0777:0023 83C702        ADD     DI,+02
0777:0026 83F900        CMP     CX,+00
0777:0029 7402          JZ      002D
0777:002B E2DF          LOOP    000C
0777:002D B8004C        MOV     AX,4C00
0777:0030 CD21          INT     21
0777:0032 7AFF          JPE     0033
```

（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）

TODO：你的截图

```
-d 0 d
0776:0000  22 00 23 00 31 00 22 00-02 00 00 00 00 00        ".#.1."......
```

（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）

TODO：你的截图

```
-d 0 d
0776:0000  22 00 23 00 31 00 22 00-02 00 31 00 04 00        ".#.1."...1...
```

（4）源代码

TODO：你的源代码

```
; empty asm file

title I love asm

data segment
    M dw 22H,23H,31H,22H,02H
    len equ ($-M)/2
    maxx dw 0
    maxi 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
        mov   cx,len-1
        lea   di,M
lop:    mov   bx,[di]
        cmp   bx,0
        jnc   comp
        neg   bx
comp:   cmp   maxx,bx
        jae   jump
        mov   maxx,bx
        mov   maxi,di
jump:   add   di,2
        cmp   cx,0
        je    l1
        loop  lop
l1:     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

TODO：你的截图

```
077B:0000 B87607          MOV     AX,0776
077B:0003 8ED8            MOV     DS,AX
077B:0005 8D3E0000        LEA     DI,[0000]
077B:0009 8B1D            MOV     BX,[DI]
077B:000B 83EB30          SUB     BX,+30
077B:000E 32FF            XOR     BH,BH
077B:0010 80872B0001      ADD     BYTE PTR [BX+002B],01
077B:0015 8A872B00        MOV     AL,[BX+002B]
077B:0019 38063500        CMP     [0035],AL
077B:001D 7707            JA      0026
077B:001F A23500          MOV     [0035],AL
077B:0022 881E3600        MOV     [0036],BL
077B:0026 83C701          ADD     DI,+01
077B:0029 83FF2B          CMP     DI,+2B
077B:002C 7402            JZ      0030
077B:002E E2D9            LOOP    0009
077B:0030 8006360030      ADD     BYTE PTR [0036],30
077B:0035 B402            MOV     AH,02
077B:0037 8A163600        MOV     DL,[0036]
077B:003B CD21            INT     21
```

```
077B:003D B402            MOV     AH,02
077B:003F B22C            MOV     DL,2C
077B:0041 CD21            INT     21
077B:0043 A03500          MOV     AL,[0035]
077B:0046 B400            MOV     AH,00
077B:0048 33D2            XOR     DX,DX
077B:004A BB0000          MOV     BX,0000
077B:004D 33C9            XOR     CX,CX
077B:004F BB0A00          MOV     BX,000A
077B:0052 F7F3            DIV     BX
077B:0054 52              PUSH    DX
077B:0055 33D2            XOR     DX,DX
077B:0057 41              INC     CX
077B:0058 3D0000          CMP     AX,0000
077B:005B 7402            JZ      005F
077B:005D EBF3            JMP     0052
077B:005F 5A              POP     DX
077B:0060 80C230          ADD     DL,30
077B:0063 B402            MOV     AH,02
077B:0065 CD21            INT     21
077B:0067 E2F6            LOOP    005F
077B:0069 B8004C          MOV     AX,4C00
077B:006C CD21            INT     21
077B:006E FF50B8          CALL    [BX+SI-48]
```

（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）

TODO：你的截图

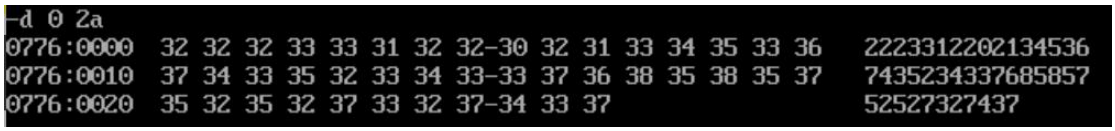
```
-d 0 2a
0776:0000  32 32 32 33 33 31 32 32-30 32 31 33 34 35 33 36   2223312202134536
0776:0010  37 34 33 35 32 33 34 33-33 37 36 38 35 38 35 37   7435234337685857
0776:0020  35 32 35 32 37 33 32 37-34 33 37                  52527327437
```

（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）

TODO：你的截图

（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）

TODO：你的截图



（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）

TODO：你的截图

（6）程序在 DOSBox 下直接运行的截图

TODO：你的截图

（7）源代码

TODO：你的源代码

```
1   ; empty asm file
2
3   title I love asm
4
5   data segment
6       string db '22233122021345367435234337685857525273274 37'
7       len equ ($-string)
8       count  db 10 dup(?)
9       maxx db ?
10      maxi db ?
11      number db 16 dup(?)
12  data ends
13  code segment
14      assume cs:code, ds:data
```

```
15      main    proc
16          ; assign the data segment base address to DS
17          mov   ax, data
18          mov   ds, ax
19          lea   di, byte ptr string
20  lop:    mov   bx, [di]
21          sub   bx, 30h
22          xor   bh, bh
23          add   count[bx],1
24          mov   al, count[bx]
25          cmp   maxx, al
26          ja    l1
27          mov   maxx, al
28          mov   maxi, bl
29  l1:     add   di, 1
30          cmp   di, len
31          je    print
32          loop  lop
33  print:
34          add   maxi, 48
35          mov   ah, 2h
36          mov   dl, maxi
37          int   21h
38          mov   ah, 2h
39          mov   dl, 44
40          int   21h
41  l2:     mov   al, maxx
```

```
42          mov   ah, 0
43          xor   dx, dx
44          mov   bx, 0
45          xor   cx, cx
46          mov   bx, 10
47  div_next:
48          div   bx
49          push  dx
50          xor   dx, dx
51          inc   cx
52          cmp   ax, 0
53          jz    display_begin
54          jmp   div_next
55  display_begin:
56          pop   dx
57          add   dl, 48
58          mov   ah,  2
59          int   21h
60          loop display_begin
61          mov   ax, 4c00h
62          int   21h
63      main    endp
64  code ends
65  end main
```