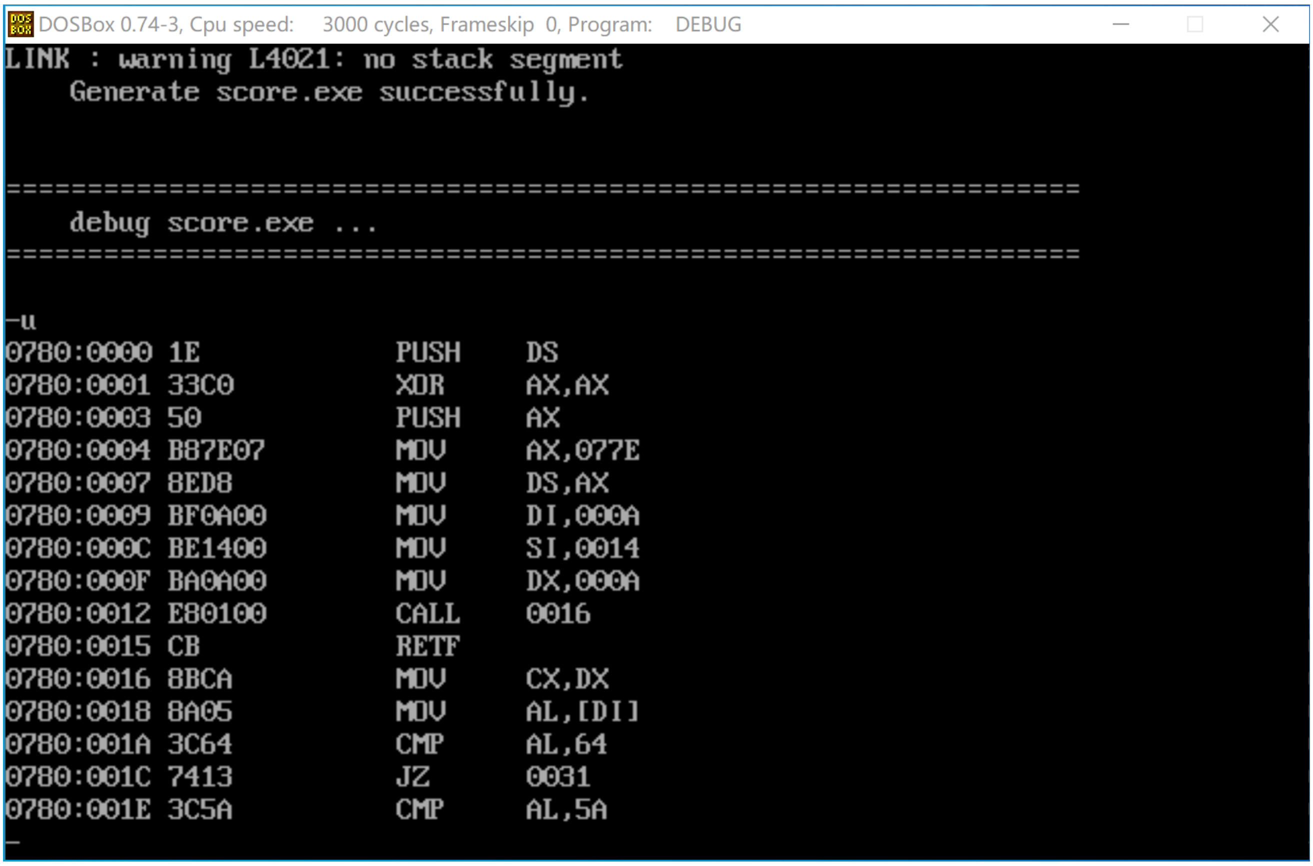
汇编语言第3次上机

班级	学号	姓名
计算机2205	2204112913	李雨轩

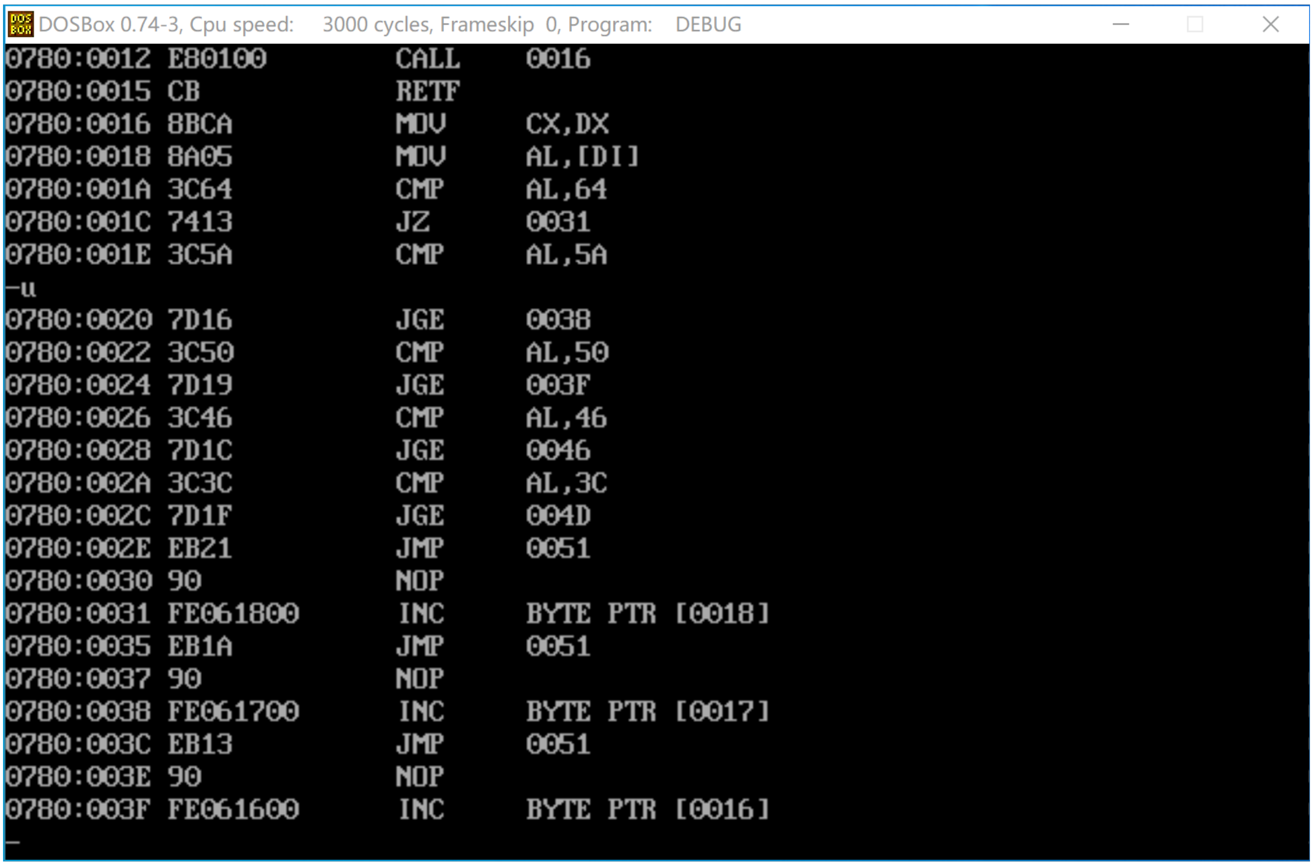
1. 子程序设计

> *设有10个学生的成绩分别是76, 69, 84, 90, 73, 88, 99, 63, 100和80分。试编制一个子程序统计60~ 69分，70~ 79分，80~ 89分，90~99分和100分的人数并分别存放到`S6， S7， S8， S9`和`S10`单元中。数据段中至少需要定义以下内容：*
>
> *（1） `ID db '2186123456'`(说明：以学号2186123456为例，此处应更换为自己的学号)*
> *（2） `array db ...`(存放10个学生的成绩) （3） 定义`S6, S7, S8, S9, S10`的内存单元*

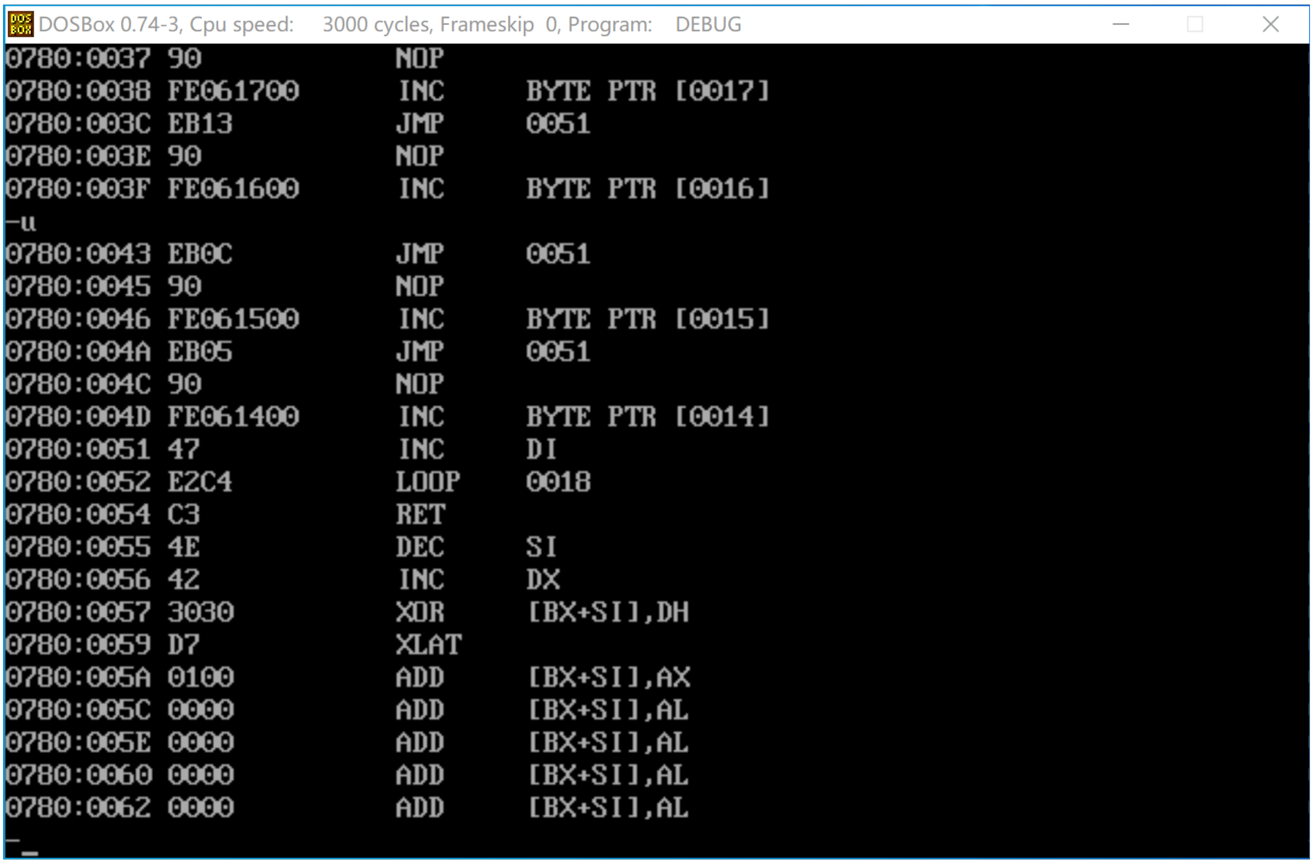
1.1 反汇编的截图


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
LINK : warning L4021: no stack segment
    Generate score.exe successfully.


==================================================================
    debug score.exe ...
==================================================================

-u
0780:0000 1E            PUSH    DS
0780:0001 33C0          XOR     AX,AX
0780:0003 50            PUSH    AX
0780:0004 B87E07        MOV     AX,077E
0780:0007 8ED8          MOV     DS,AX
0780:0009 BF0A00        MOV     DI,000A
0780:000C BE1400        MOV     SI,0014
0780:000F BA0A00        MOV     DX,000A
0780:0012 E80100        CALL    0016
0780:0015 CB            RETF
0780:0016 8BCA          MOV     CX,DX
0780:0018 8A05          MOV     AL,[DI]
0780:001A 3C64          CMP     AL,64
0780:001C 7413          JZ      0031
0780:001E 3C5A          CMP     AL,5A
-
```

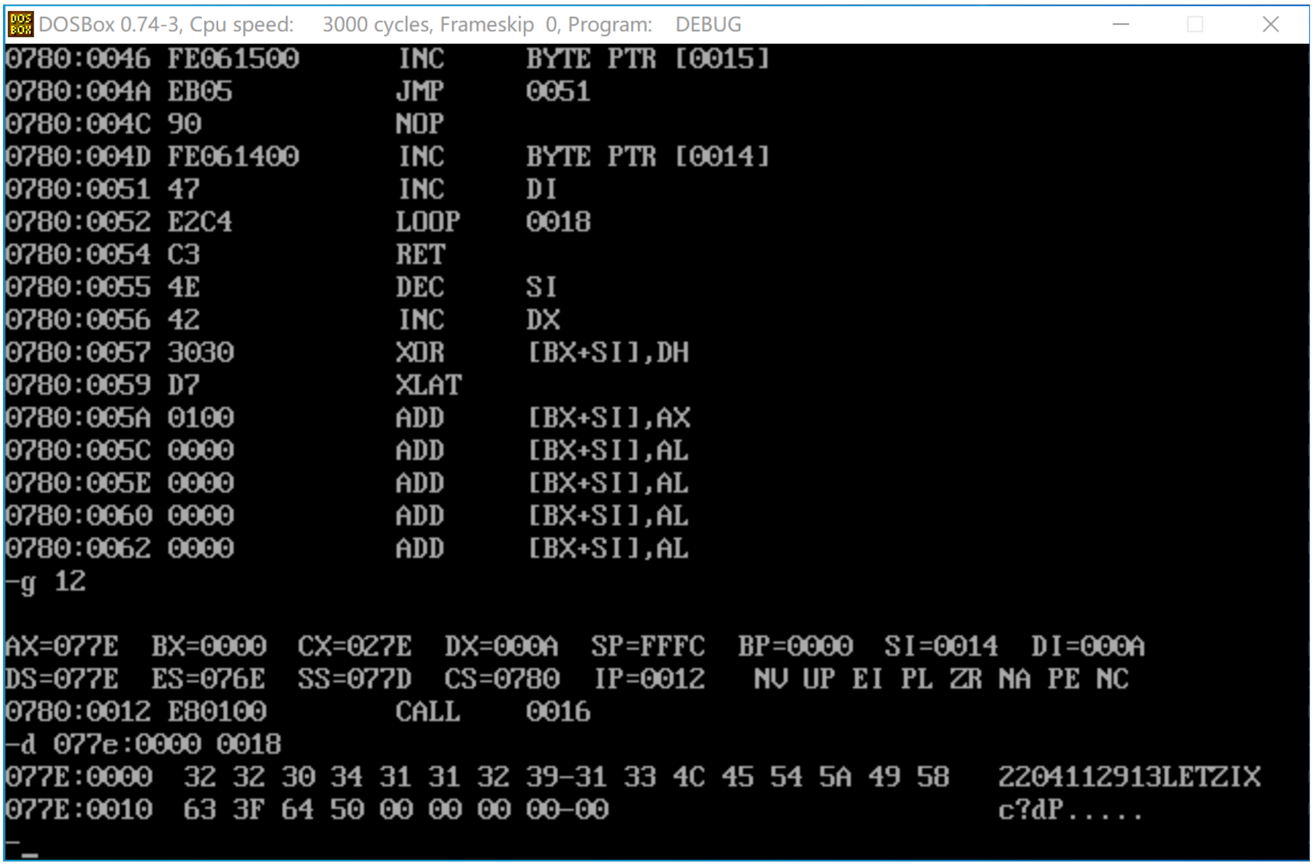

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
0780:0012 E80100        CALL    0016
0780:0015 CB            RETF
0780:0016 8BCA          MOV     CX,DX
0780:0018 8A05          MOV     AL,[DI]
0780:001A 3C64          CMP     AL,64
0780:001C 7413          JZ      0031
0780:001E 3C5A          CMP     AL,5A
-u
0780:0020 7D16          JGE     0038
0780:0022 3C50          CMP     AL,50
0780:0024 7D19          JGE     003F
0780:0026 3C46          CMP     AL,46
0780:0028 7D1C          JGE     0046
0780:002A 3C3C          CMP     AL,3C
0780:002C 7D1F          JGE     004D
0780:002E EB21          JMP     0051
0780:0030 90            NOP
0780:0031 FE061800      INC     BYTE PTR [0018]
0780:0035 EB1A          JMP     0051
0780:0037 90            NOP
0780:0038 FE061700      INC     BYTE PTR [0017]
0780:003C EB13          JMP     0051
0780:003E 90            NOP
0780:003F FE061600      INC     BYTE PTR [0016]
-
```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

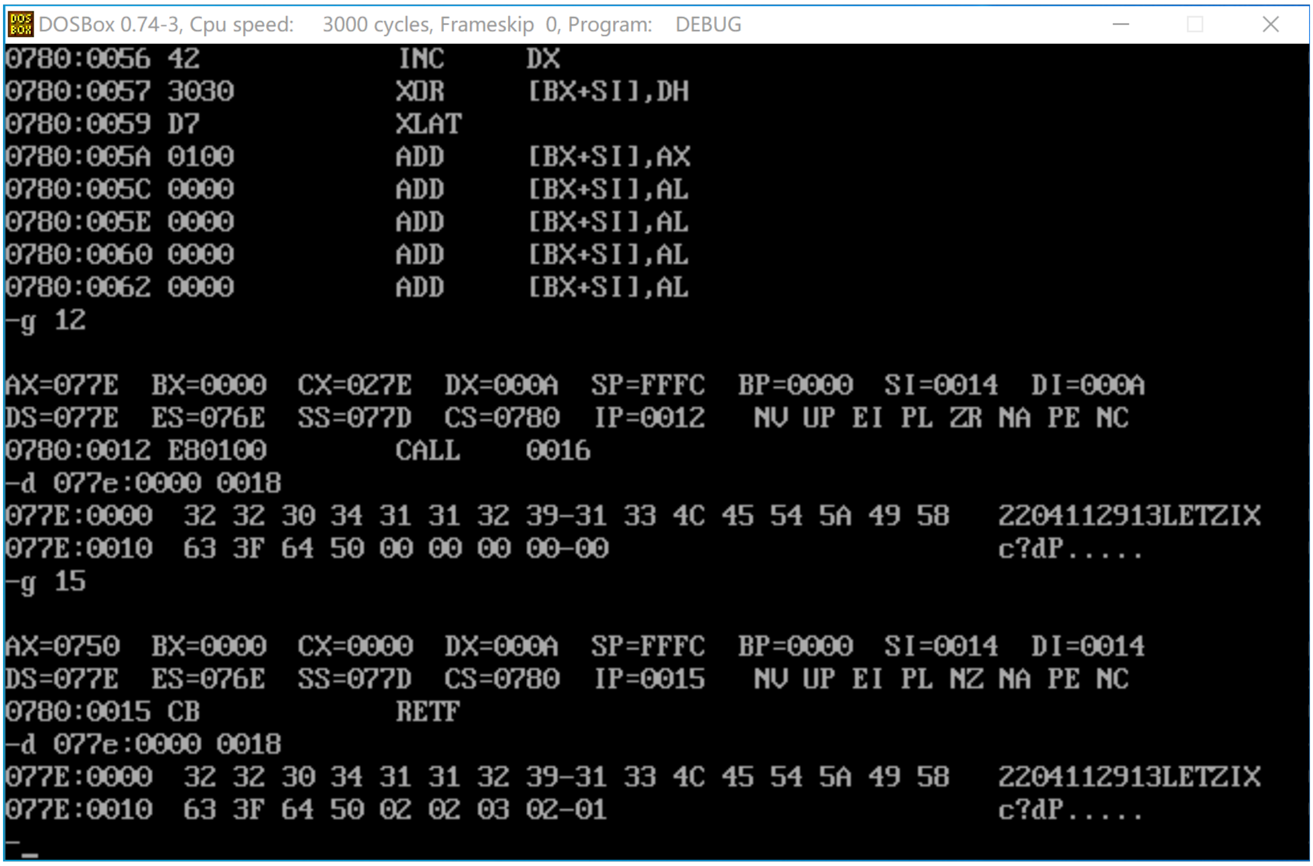
```
0780:0037 90            NOP
0780:0038 FE061700      INC     BYTE PTR [0017]
0780:003C EB13          JMP     0051
0780:003E 90            NOP
0780:003F FE061600      INC     BYTE PTR [0016]
-u
0780:0043 EB0C          JMP     0051
0780:0045 90            NOP
0780:0046 FE061500      INC     BYTE PTR [0015]
0780:004A EB05          JMP     0051
0780:004C 90            NOP
0780:004D FE061400      INC     BYTE PTR [0014]
0780:0051 47            INC     DI
0780:0052 E2C4          LOOP    0018
0780:0054 C3            RET
0780:0055 4E            DEC     SI
0780:0056 42            INC     DX
0780:0057 3030          XOR     [BX+SI],DH
0780:0059 D7            XLAT
0780:005A 0100          ADD     [BX+SI],AX
0780:005C 0000          ADD     [BX+SI],AL
0780:005E 0000          ADD     [BX+SI],AL
0780:0060 0000          ADD     [BX+SI],AL
0780:0062 0000          ADD     [BX+SI],AL
-_
```

1.2 在进行计算前，显示ID、array以及S6——S10的内存值的截图（多显示、少显示均扣分）


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
0780:0046 FE061500      INC     BYTE PTR [0015]
0780:004A EB05          JMP     0051
0780:004C 90            NOP
0780:004D FE061400      INC     BYTE PTR [0014]
0780:0051 47            INC     DI
0780:0052 E2C4          LOOP    0018
0780:0054 C3            RET
0780:0055 4E            DEC     SI
0780:0056 42            INC     DX
0780:0057 3030          XOR     [BX+SI],DH
0780:0059 D7            XLAT
0780:005A 0100          ADD     [BX+SI],AX
0780:005C 0000          ADD     [BX+SI],AL
0780:005E 0000          ADD     [BX+SI],AL
0780:0060 0000          ADD     [BX+SI],AL
0780:0062 0000          ADD     [BX+SI],AL
-g 12

AX=077E  BX=0000  CX=027E  DX=000A  SP=FFFC  BP=0000  SI=0014  DI=000A
DS=077E  ES=076E  SS=077D  CS=0780  IP=0012   NV UP EI PL ZR NA PE NC
0780:0012 E80100        CALL    0016
-d 077e:0000 0018
077E:0000  32 32 30 34 31 31 32 39-31 33 4C 45 54 5A 49 58   2204112913LETZIX
077E:0010  63 3F 64 50 00 00 00 00-00                        c?dP.....
-_
```

1.3 执行完计算后，显示ID、array以及S6——S10的内存值的截图（多显示、少显示均扣分


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
0780:0056 42            INC     DX
0780:0057 3030          XOR     [BX+SI],DH
0780:0059 D7            XLAT
0780:005A 0100          ADD     [BX+SI],AX
0780:005C 0000          ADD     [BX+SI],AL
0780:005E 0000          ADD     [BX+SI],AL
0780:0060 0000          ADD     [BX+SI],AL
0780:0062 0000          ADD     [BX+SI],AL
-g 12

AX=077E  BX=0000  CX=027E  DX=000A  SP=FFFC  BP=0000  SI=0014  DI=000A
DS=077E  ES=076E  SS=077D  CS=0780  IP=0012   NV UP EI PL ZR NA PE NC
0780:0012 E80100        CALL    0016
-d 077e:0000 0018
077E:0000  32 32 30 34 31 31 32 39-31 33 4C 45 54 5A 49 58   2204112913LETZIX
077E:0010  63 3F 64 50 00 00 00 00-00                        c?dP.....
-g 15

AX=0750  BX=0000  CX=0000  DX=000A  SP=FFFC  BP=0000  SI=0014  DI=0014
DS=077E  ES=076E  SS=077D  CS=0780  IP=0015   NV UP EI PL NZ NA PE NC
0780:0015 CB            RETF
-d 077e:0000 0018
077E:0000  32 32 30 34 31 31 32 39-31 33 4C 45 54 5A 49 58   2204112913LETZIX
077E:0010  63 3F 64 50 02 02 03 02-01                        c?dP.....
-_
```

1.4 源代码

```
name Score
title Count students score

MSG MACRO COUNT
    S&COUNT DB 0
ENDM

data segment
    ID  db  '2204112913'
    array db 76, 69, 84, 90, 73, 88, 99, 63, 100, 80
    counts label byte
    X = 6
    REPT 5
        MSG %X
        X = X + 1
    ENDM
data ends

code segment
    assume cs:code, ds:data

    main proc far
        push ds
        xor ax, ax
        push ax
        mov ax, seg data
        mov ds, ax

        mov di, offset array
        mov si, offset counts
        mov dx, counts - array
        call countScore

        ret
    main endp

    countScore proc near
        mov cx, dx

count_scores:
        mov al, [di]

        cmp al, 100
        je  increment_S10
        cmp al, 90
        jge increment_S9
        cmp al, 80
        jge increment_S8
        cmp al, 70
        jge increment_S7
        cmp al, 60
        jge increment_S6
        jmp next_student

increment_S10:
        inc byte ptr S10
        jmp next_student
increment_S9:
        inc byte ptr S9
        jmp next_student
increment_S8:
        inc byte ptr S8
        jmp next_student
increment_S7:
        inc byte ptr S7
        jmp next_student
increment_S6:
        inc byte ptr S6
next_student:
        inc di
        loop count_scores
        ret
    countScore endp

code ends
    end main
```

2. 高级汇编语言技术

> *试使用条件汇编和重复汇编编写一段程序，完成以下功能：根据给定名为X的字符串长度汇编以下指令，如果X的长度为n，当n<=5时，汇编n次；当n>5时，汇编6次。* `ADD AX, AX`
>
> *数据段中至少需要定义以下内容：*
>
> *（1）* `ID db '2186123456'` *(说明：以学号2186123456为例，此处应更换为自己的学号)（2）X 是一个目标字符串，根据下面的测试场景需求进行定义；（3）从这里定义自己认为必要的变量*
>
> *测试场景要求：*
>
> *（1）X 是一个长度为5 的字符串，例如：'23456'；学号的后5位（2）X 是一个长度为10的字符串，例如：'2186123456'；完整的学号（3）测试时，只能改变数据段中X的定义内容（通过注释符号），其它数据段、代码段中的内容必须保持不变。*

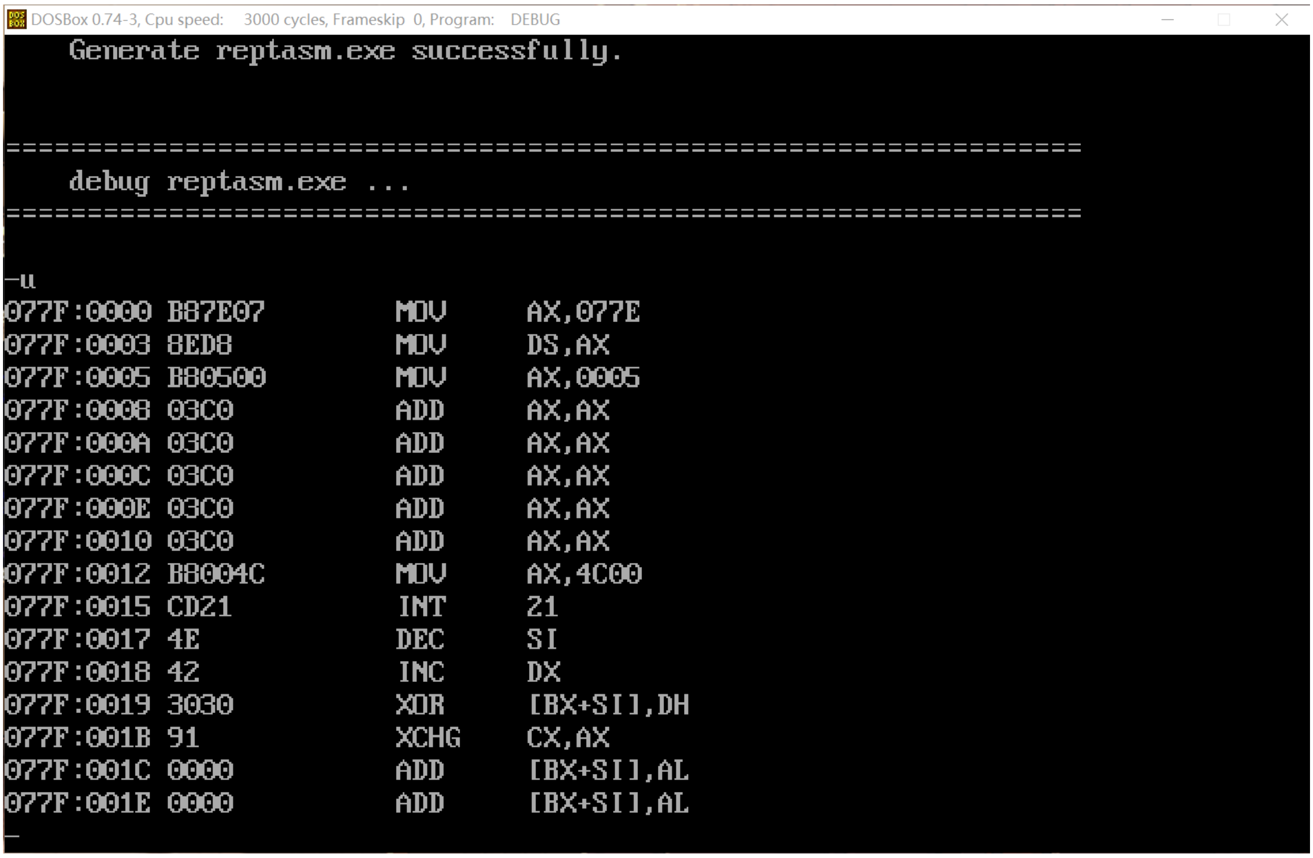
2.1 场景1的.lst文件的截图

ASM3\REPTASM.LST

```
 Microsoft (R) Macro Assembler Version 5.00                 4/21/24 23:38:09
conditional Assembly and repitition Assembly               Page     1-1


                                name REPTASM
                                title conditional Assembly and repitition Assem
                                bly

 0000                           data segment
 0000  32 32 30 34 31 31 32         ID db '2204112913'
       39 31 33

 000A                               strBegin label byte
 000A  32 32 30 34 31               X db '22041'
 000F                               strEnd label byte

 000F                           data ends

 0000                           code segment
                                    assume cs:code, ds:data
 0000                               main proc far
 0000  B8 ---- R                        mov ax, seg data
 0003  8E D8                            mov ds, ax
 0005  B8 0005                          mov ax, strEnd-strBegin

                                    IF strEnd-strBegin LE 5 ; if length of stri
                                ng is less than or equal to 5
                                        REPT strEnd-strBegin
                                            add ax, ax
                                        ENDM
 0008  03 C0                 1              add ax, ax
 000A  03 C0                 1              add ax, ax
 000C  03 C0                 1              add ax, ax
 000E  03 C0                 1              add ax, ax
 0010  03 C0                 1              add ax, ax
                                    ENDIF

 0012  B8 4C00                          mov ax, 4c00h
 0015  CD 21                            int 21h
 0017                               main endp

 0017                           code ends
                                    end main
 Microsoft (R) Macro Assembler Version 5.00                 4/21/24 23:38:09
conditional Assembly and repitition Assembly               Symbols-1


Segments and Groups:

                N a m e                 Length   Align  Combine Class

CODE . . . . . . . . . . . . . .        0017    PARA    NONE
DATA . . . . . . . . . . . . . .        000F    PARA    NONE

Symbols:

                N a m e                 Type     Value   Attr

ID . . . . . . . . . . . . . . .        L BYTE  0000    DATA

MAIN . . . . . . . . . . . . . .        F PROC  0000    CODE    Length = 0017

STRBEGIN . . . . . . . . . . . .        L BYTE  000A    DATA
STREND . . . . . . . . . . . . .        L BYTE  000F    DATA
```

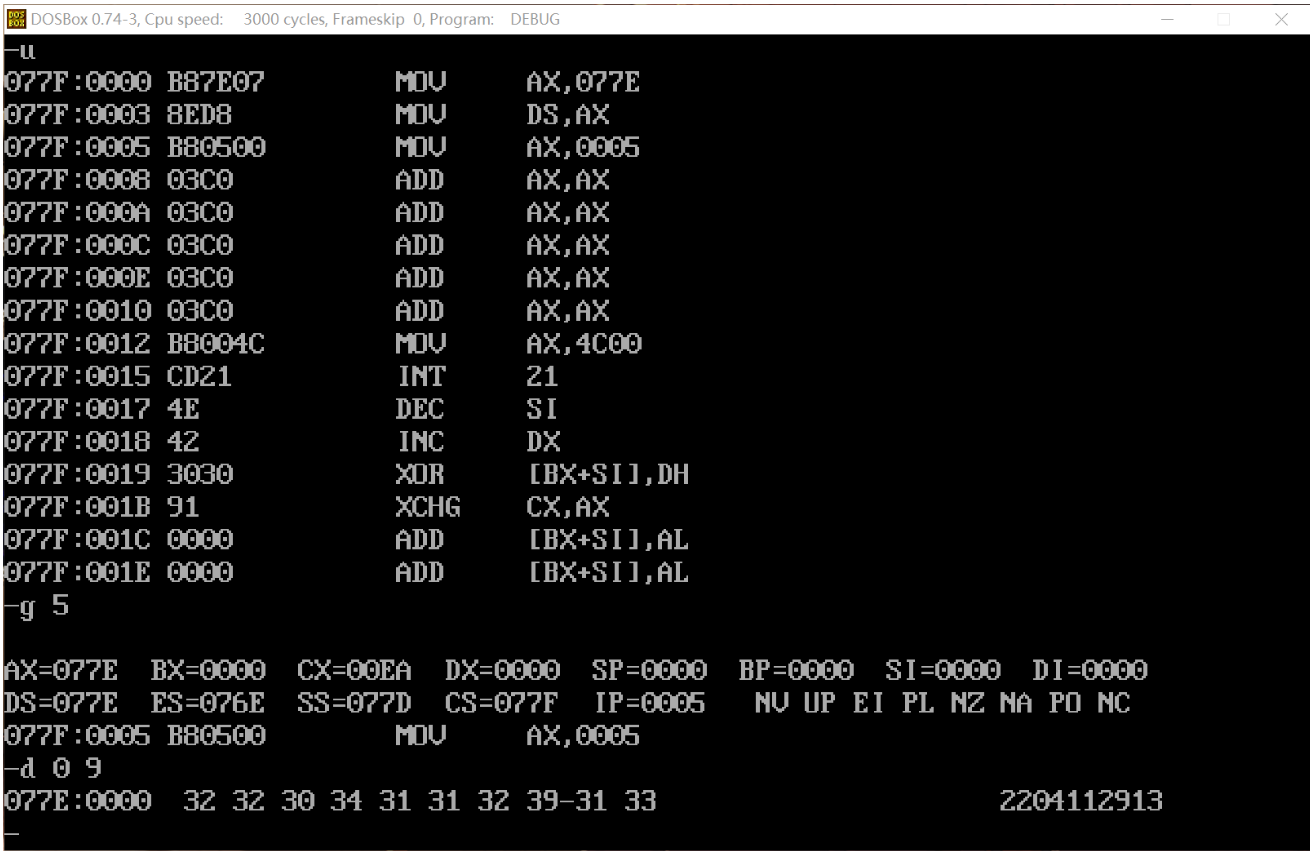
```
X . . . . . . . . . . . . . . .          L BYTE   000A     DATA

@FILENAME . . . . . . . . . . .          TEXT  reptASM


     35 Source  Lines
     40 Total   Lines
      9 Symbols

  50960 + 465424 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors
```

2.2 场景1的反汇编的截图


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
     Generate reptasm.exe successfully.


==================================================================
     debug reptasm.exe ...
==================================================================

-u
077F:0000 B87E07        MOV     AX,077E
077F:0003 8ED8          MOV     DS,AX
077F:0005 B80500        MOV     AX,0005
077F:0008 03C0          ADD     AX,AX
077F:000A 03C0          ADD     AX,AX
077F:000C 03C0          ADD     AX,AX
077F:000E 03C0          ADD     AX,AX
077F:0010 03C0          ADD     AX,AX
077F:0012 B8004C        MOV     AX,4C00
077F:0015 CD21          INT     21
077F:0017 4E            DEC     SI
077F:0018 42            INC     DX
077F:0019 3030          XOR     [BX+SI],DH
077F:001B 91            XCHG    CX,AX
077F:001C 0000          ADD     [BX+SI],AL
077F:001E 0000          ADD     [BX+SI],AL
-
```

2.3 场景1的显示X的内存值的截图（多显示、少显示均扣分）



DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
-u
077F:0000 B87E07        MOV     AX,077E
077F:0003 8ED8          MOV     DS,AX
077F:0005 B80500        MOV     AX,0005
077F:0008 03C0          ADD     AX,AX
077F:000A 03C0          ADD     AX,AX
077F:000C 03C0          ADD     AX,AX
077F:000E 03C0          ADD     AX,AX
077F:0010 03C0          ADD     AX,AX
077F:0012 B8004C        MOV     AX,4C00
077F:0015 CD21          INT     21
077F:0017 4E            DEC     SI
077F:0018 42            INC     DX
077F:0019 3030          XOR     [BX+SI],DH
077F:001B 91            XCHG    CX,AX
077F:001C 0000          ADD     [BX+SI],AL
077F:001E 0000          ADD     [BX+SI],AL
-g 5

AX=077E  BX=0000  CX=00EA  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=077E  ES=076E  SS=077D  CS=077F  IP=0005   NV UP EI PL NZ NA PO NC
077F:0005 B80500        MOV     AX,0005
-d 0 9
077E:0000  32 32 30 34 31 31 32 39-31 33                     2204112913
-
```

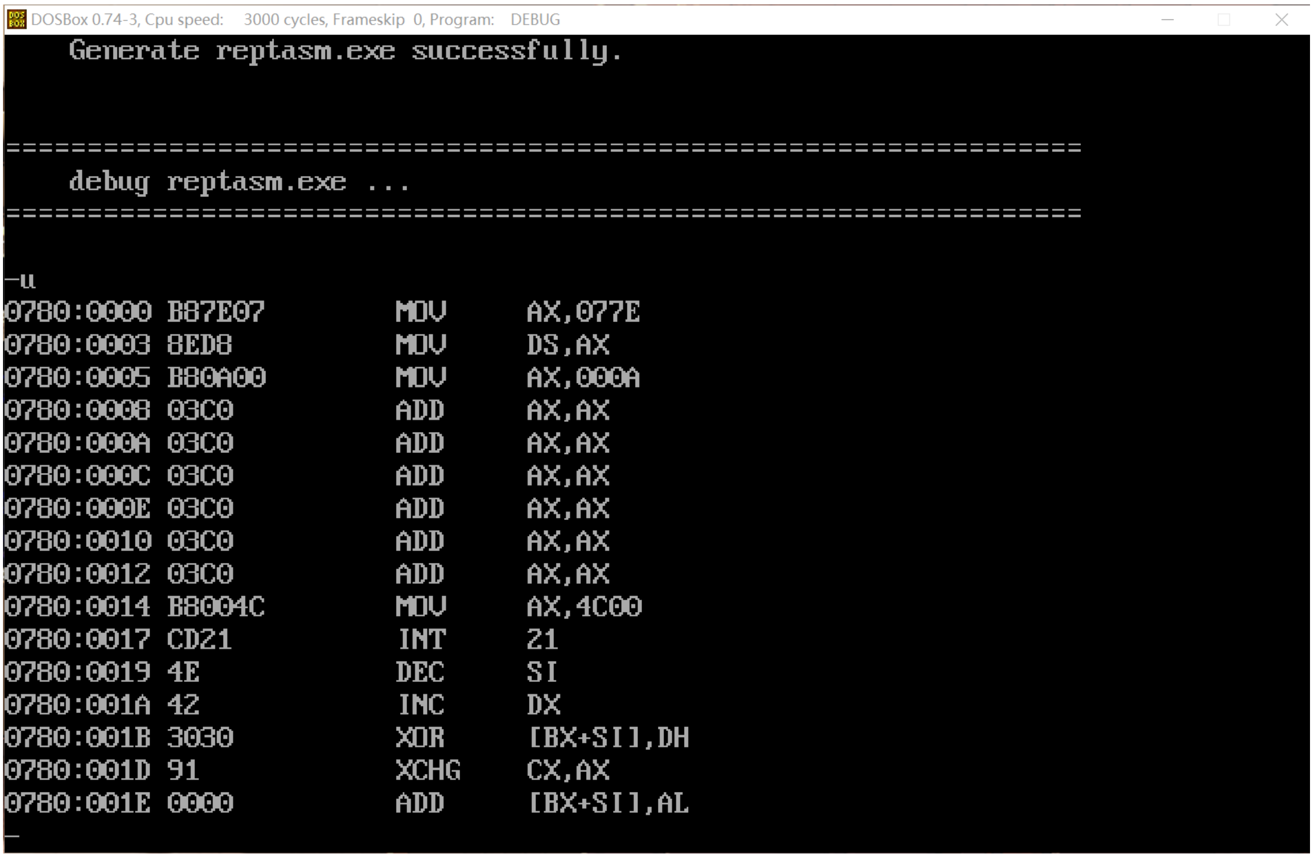
2.4 场景2的.lst文件的截图

ASM3\REPTASM.LST

```
 Microsoft (R) Macro Assembler Version 5.00                 4/21/24 23:40:26
conditional Assembly and repitition Assembly                Page     1-1


                                name REPTASM
                                title conditional Assembly and repitition Assem
                                bly

 0000                           data segment
 0000  32 32 30 34 31 31 32         ID db '2204112913'
       39 31 33

 000A                               strBegin label byte
 000A  32 32 30 34 31 31 32         X db '2204112913'
       39 31 33
 0014                               strEnd label byte

 0014                           data ends

 0000                           code segment
                                    assume cs:code, ds:data
 0000                               main proc far
 0000  B8 ---- R                        mov ax, seg data
 0003  8E D8                            mov ds, ax
 0005  B8 000A                          mov ax, strEnd-strBegin

                                    ELSE
                                        REPT 6
                                            add ax, ax
                                        ENDM
 0008  03 C0                 1              add ax, ax
 000A  03 C0                 1              add ax, ax
 000C  03 C0                 1              add ax, ax
 000E  03 C0                 1              add ax, ax
 0010  03 C0                 1              add ax, ax
 0012  03 C0                 1              add ax, ax
                                    ENDIF

 0014  B8 4C00                          mov ax, 4c00h
 0017  CD 21                            int 21h
 0019                               main endp

 0019                           code ends
                                    end main
 Microsoft (R) Macro Assembler Version 5.00                 4/21/24 23:40:26
conditional Assembly and repitition Assembly                Symbols-1


Segments and Groups:

                N a m e                 Length   Align  Combine Class

CODE . . . . . . . . . . . . . .        0019    PARA    NONE
DATA . . . . . . . . . . . . . .        0014    PARA    NONE

Symbols:

                N a m e                 Type     Value   Attr

ID . . . . . . . . . . . . . . .        L BYTE  0000    DATA

MAIN . . . . . . . . . . . . . .        F PROC  0000    CODE    Length = 0019

STRBEGIN . . . . . . . . . . . .        L BYTE  000A    DATA
STREND . . . . . . . . . . . . .        L BYTE  0014    DATA
```

```
X . . . . . . . . . . . . . . .        L BYTE  000A    DATA

@FILENAME . . . . . . . . . . .        TEXT  reptASM


     35 Source  Lines
     41 Total   Lines
      9 Symbols

  50960 + 465424 Bytes symbol space free

      0 Warning Errors
      0 Severe  Errors
```

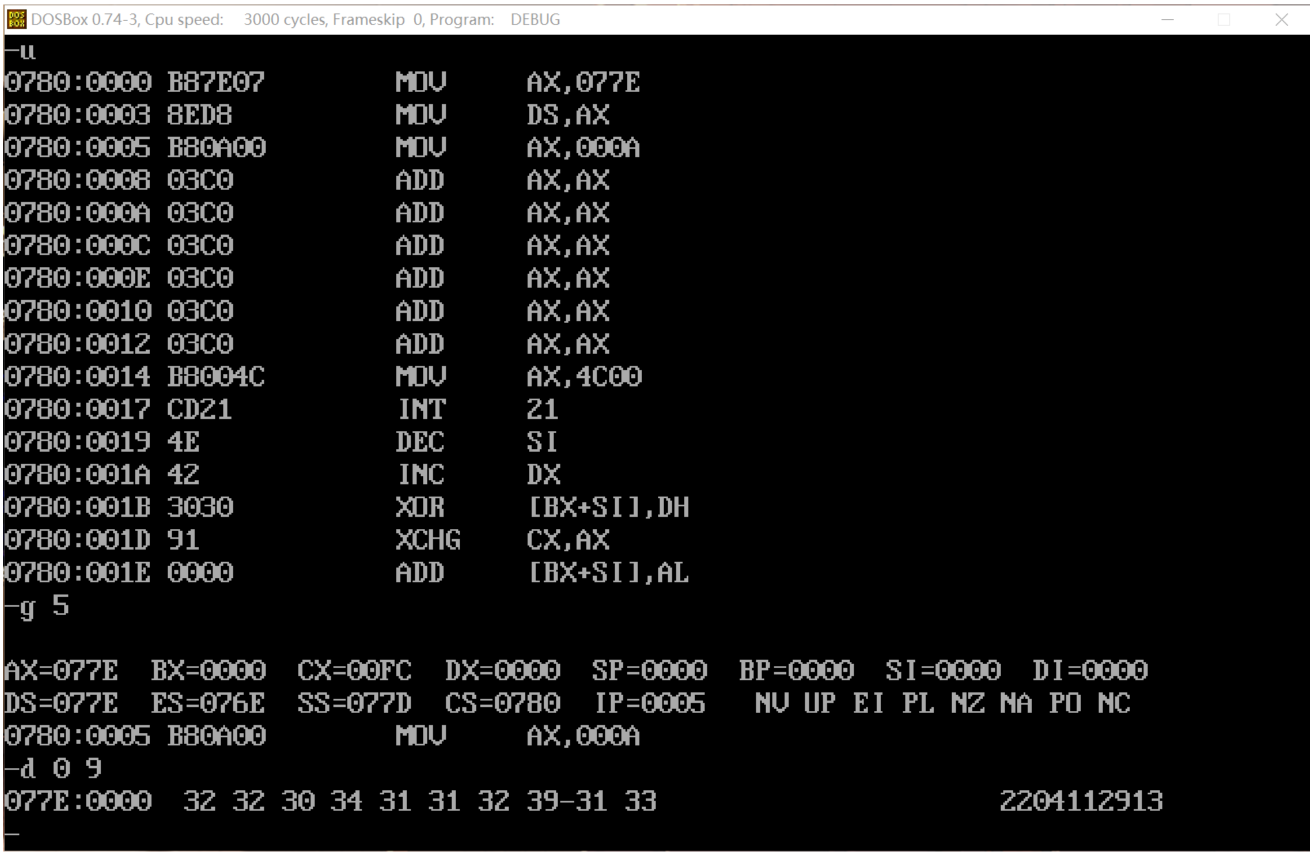
2.5 场景2的反汇编的截图



```
    Generate reptasm.exe successfully.


==================================================================
    debug reptasm.exe ...
==================================================================

-u
0780:0000 B87E07        MOV     AX,077E
0780:0003 8ED8          MOV     DS,AX
0780:0005 B80A00        MOV     AX,000A
0780:0008 03C0          ADD     AX,AX
0780:000A 03C0          ADD     AX,AX
0780:000C 03C0          ADD     AX,AX
0780:000E 03C0          ADD     AX,AX
0780:0010 03C0          ADD     AX,AX
0780:0012 03C0          ADD     AX,AX
0780:0014 B8004C        MOV     AX,4C00
0780:0017 CD21          INT     21
0780:0019 4E            DEC     SI
0780:001A 42            INC     DX
0780:001B 3030          XOR     [BX+SI],DH
0780:001D 91            XCHG    CX,AX
0780:001E 0000          ADD     [BX+SI],AL
-
```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

2.6 场景2的显示X的内存值的截图（多显示、少显示均扣分）


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
-u
0780:0000 B87E07        MOV     AX,077E
0780:0003 8ED8          MOV     DS,AX
0780:0005 B80A00        MOV     AX,000A
0780:0008 03C0          ADD     AX,AX
0780:000A 03C0          ADD     AX,AX
0780:000C 03C0          ADD     AX,AX
0780:000E 03C0          ADD     AX,AX
0780:0010 03C0          ADD     AX,AX
0780:0012 03C0          ADD     AX,AX
0780:0014 B8004C        MOV     AX,4C00
0780:0017 CD21          INT     21
0780:0019 4E            DEC     SI
0780:001A 42            INC     DX
0780:001B 3030          XOR     [BX+SI],DH
0780:001D 91            XCHG    CX,AX
0780:001E 0000          ADD     [BX+SI],AL
-g 5

AX=077E  BX=0000  CX=00FC  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=077E  ES=076E  SS=077D  CS=0780  IP=0005   NV UP EI PL NZ NA PO NC
0780:0005 B80A00        MOV     AX,000A
-d 0 9
077E:0000  32 32 30 34 31 31 32 39-31 33                     2204112913
-
```

2.7 源代码

```
name REPTASM
title conditional Assembly and repitition Assembly

data segment
    ID db '2204112913'

    strBegin label byte
    X db '2204112913'
    strEnd label byte

data ends

code segment
    assume cs:code, ds:data
    main proc far
        mov ax, seg data
        mov ds, ax
        mov ax, strEnd-strBegin

    IF strEnd-strBegin LE 5 ; if length of string is less than or equal to 5
        REPT strEnd-strBegin
            add ax, ax
        ENDM

    ELSE
```

```
        REPT 6
            add ax, ax
        ENDM
    ENDIF

        mov ax, 4c00h
        int 21h
    main endp

code ends
    end main
```