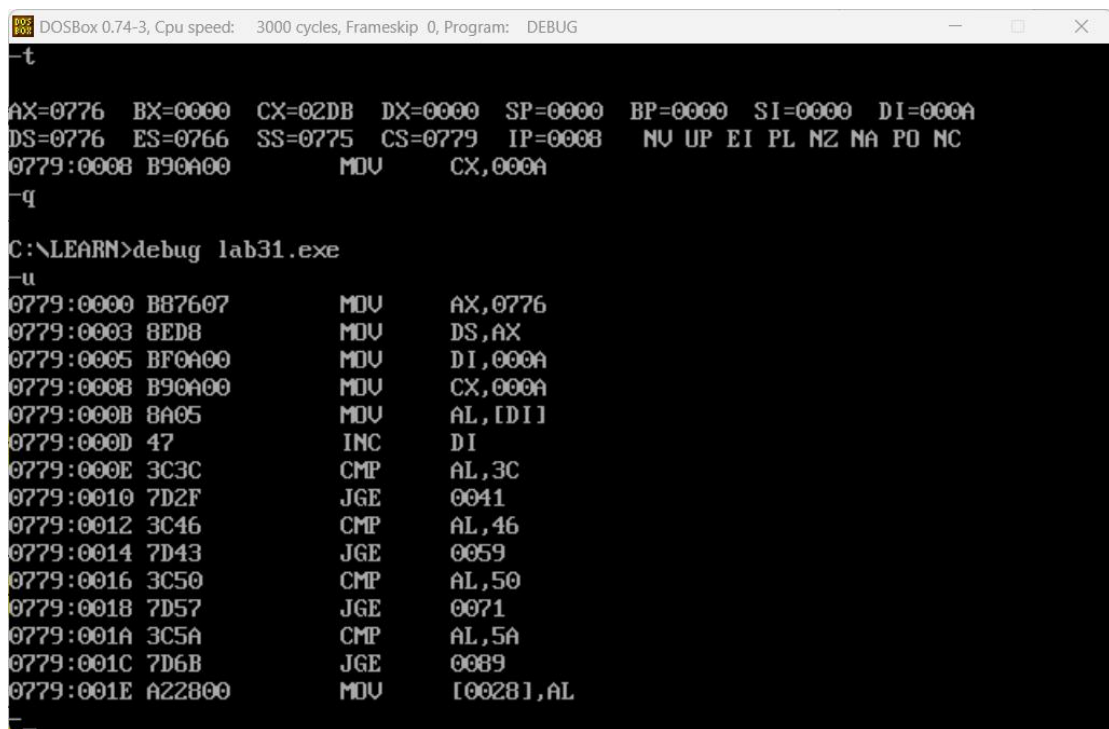


第 3 次上机(x86)

班级	学号	姓名
计试 2201	2223211946	王顺平

1、子程序设计

(1) 反汇编的截图



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip: 0, Program: DEBUG
-t
AX=0776 BX=0000 CX=02DB DX=0000 SP=0000 BP=0000 SI=0000 DI=000A
DS=0776 ES=0766 SS=0775 CS=0779 IP=0008  NU UP EI PL NZ NA PO NC
0779:0008 B90A00      MOV     CX,000A
-q
C:\LEARN>debug lab31.exe
-u
0779:0000 B87607      MOV     AX,0776
0779:0003 8ED8        MOV     DS,AX
0779:0005 BF0A00      MOV     DI,000A
0779:0008 B90A00      MOV     CX,000A
0779:000B 8A05        MOV     AL,[DI]
0779:000D 47          INC     DI
0779:000E 3C3C        CMP     AL,3C
0779:0010 7D2F        JGE     0041
0779:0012 3C46        CMP     AL,46
0779:0014 7D43        JGE     0059
0779:0016 3C50        CMP     AL,50
0779:0018 7D57        JGE     0071
0779:001A 3C5A        CMP     AL,5A
0779:001C 7D6B        JGE     0089
0779:001E A22800      MOV     [0028],AL
```

(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
0776:0010 63 3F 64 50 0A 00 00 00-00 0A 00 00 00 00 0A 00 c?dP.....
0776:0020 00 00 00 0A 00 00 00 00-0A .....
-q
C:\LEARN>debug lab31.exe
-t
AX=0776 BX=0000 CX=02DB DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0766 ES=0766 SS=0775 CS=0779 IP=0003  NU UP EI PL NZ NA PO NC
0779:0003 8ED8          MOV     DS,AX
-t
AX=0776 BX=0000 CX=02DB DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0776 ES=0766 SS=0775 CS=0779 IP=0005  NU UP EI PL NZ NA PO NC
0779:0005 BFA000      MOV     DI,000A
-t
AX=0776 BX=0000 CX=02DB DX=0000 SP=0000 BP=0000 SI=0000 DI=000A
DS=0776 ES=0766 SS=0775 CS=0779 IP=0008  NU UP EI PL NZ NA PO NC
0779:0008 B90A00      MOV     CX,000A
-d 0 28
0776:0000 32 32 32 33 32 31 31 39-34 36 4C 45 54 5A 49 58 2223211946LETZIX
0776:0010 63 3F 64 50 0A 00 00 00-00 0A 00 00 00 00 0A 00 c?dP.....
0776:0020 00 00 00 0A 00 00 00 00-0A .....
```

(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
-t
AX=0750 BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000 DI=0014
DS=0776 ES=0766 SS=0775 CS=0779 IP=002D  NU UP EI PL NZ NA PE NC
0779:002D C606220000      MOV     BYTE PTR [0022],00      DS:0022=03
-t
AX=0750 BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000 DI=0014
DS=0776 ES=0766 SS=0775 CS=0779 IP=0032  NU UP EI PL NZ NA PE NC
0779:0032 C606270000      MOV     BYTE PTR [0027],00      DS:0027=02
-t
AX=0750 BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000 DI=0014
DS=0776 ES=0766 SS=0775 CS=0779 IP=0037  NU UP EI PL NZ NA PE NC
0779:0037 C6062C0000      MOV     BYTE PTR [002C],00      DS:002C=00
-t
AX=0750 BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000 DI=0014
DS=0776 ES=0766 SS=0775 CS=0779 IP=003C  NU UP EI PL NZ NA PE NC
0779:003C B8004C          MOV     AX,4C00
-d 0 28
0776:0000 32 32 32 33 32 31 31 39-34 36 4C 45 54 5A 49 58 2223211946LETZIX
0776:0010 63 3F 64 50 45 3F 00 00-00 4C 49 00 00 00 54 58 c?dPE?...LI...TX
0776:0020 50 00 00 5A 63 00 00 00-64 P...Zc...d
```

(4) 源代码

1. data segment
2. ID db '2223211946'

```

3.      array db 76,69,84,90,73,88,99,63,100,80
4.      S6 db 10,0,0,0,0
5.      S7 db 10,0,0,0,0
6.      S8 db 10,0,0,0,0
7.      S9 db 10,0,0,0,0
8.      S10 db 10
9.      data ends
10.
11.     code segment
12.         assume cs:code,ds:data
13.
14.     start:  mov ax,data
15.             mov ds,ax
16.
17.             mov di,offset array
18.             mov cx,10
19.     loop1:  mov al,[di]
20.             inc di
21.             cmp al,60
22.             jge ss6
23.     j7:     cmp al,70
24.             jge ss7
25.     j8:     cmp al,80
26.             jge ss8
27.     j9:     cmp al,90
28.             jge ss9
29.     j10:    mov [S10],al
30.     retL:   loop loop1
31.
32.             mov [S6 + 4],0
33.             mov [S7 + 4],0
34.             mov [S8 + 4],0
35.             mov [S9 + 4],0
36.             mov [S10 + 4],0
37.
38.             mov ax, 4c00h
39.             int 21h
40.
41.     ss6:    cmp al,70
42.             jge j7
43.
44.             push di
45.             mov di,word ptr[S6 + 4]
46.             and di,00FFh

```

```
47.      mov [S6 + di],al
48.      inc [S6 + 4]
49.      pop di
50.
51.      jmp retL
52.
53. ss7:   cmp al,80
54.      jge j8
55.
56.      push di
57.      mov di,word ptr[S7 + 4]
58.      and di,00FFh
59.      mov [S7 + di],al
60.      inc [S7 + 4]
61.      pop di
62.
63.      jmp retL
64.
65. ss8:   cmp al,90
66.      jge j9
67.
68.      push di
69.      mov di,word ptr[S8 + 4]
70.      and di,00FFh
71.      mov [S8 + di],al
72.      inc [S8 + 4]
73.      pop di
74.
75.      jmp retL
76.
77. ss9:   cmp al,100
78.      jge j10
79.
80.      push di
81.      mov di,word ptr[S9 + 4]
82.      and di,00FFh
83.      mov [S9 + di],al
84.      inc [S9 + 4]
85.      pop di
86.
87.      jmp retL
88.
89.
90. code ends
```

91. end start

2、高级汇编语言技术

(1) 场景 1 的 .lst 文件的截图

```
1  Microsoft (R) Macro Assembler Version 5.00 4/21/24 14:42:52
2  Page 1-1
3
4
5      1          qq macro k
6      2
7      3          ife k - 1
8      4              rept 6
9      5                  add ax,ax
10     6              endm
11     7          endif
12     8
13     9          ife k
14    10              rept 5
15    11                  add ax,ax
16    12              endm
17    13          endif
18    14
19    15          endm
20
21    17
22    18 0000          data segment
23    19 0000 32 32 32 33 32 31 31 ID db '223211946'
24    20          39 34 36
25    21 000A 31 31 39 34 36 X db '11946'
26    22          :X db '223211946'
27    23 000F 4E Y db 'N'
28    24 0010          data ends
29    25
30    26 0000          code segment
31    27          assume cs:code,ds:data
32    28
33    29 0000 B8 ---- R start: mov ax,data
34    30 0003 8E DS      mov ds,ax
35    31
36    32 0005 EB 0005      mov bx,offset Y - offse
37    33          t X
38    34 0008 B9 0000      mov cx,0
39    35 000E 83 FB 05      cmp bx,5
40    36 0010 7E 03      jle jl
41    37 0013 B9 0001      mov cx,1
42    38          jl: mov ax,1
43    39
44    40 0016 83 F9 00      cmp cx,0
45    41 0019 74 0A      je qq0
46    42 001B 83 F9 01      cmp cx,1
47    43 001E 74 11      je qq1
48    44
49
50    45 0020 B8 4C00      rl: mov ax,4c00h
51    46 0023 CD 21      int 21h
52    47
53    48 0025          qq0:
54    49          qq 0
55    50 0025 03 C0          2 add ax,ax
56    51 0027 03 C0          2 add ax,ax
57    52 0029 03 C0          2 add ax,ax
58    53 002B 03 C0          2 add ax,ax
59
60  Microsoft (R) Macro Assembler Version 5.00 4/
61  Page
62
63    54 002D 03 C0          2 add ax,ax
64    55 002F EB EF          jmp rl
65    56 0031          qq1:
66    57          qq 1
67    58 0031 03 C0          2 add ax,ax
68    59 0033 03 C0          2 add ax,ax
69    60 0035 03 C0          2 add ax,ax
70    61 0037 03 C0          2 add ax,ax
71    62 0039 03 C0          2 add ax,ax
72    63 003B 03 C0          2 add ax,ax
73    64 003D EB E1          jmp rl
74    65
75    66 003F          code ends
76    67          end start
77
78  Microsoft (R) Macro Assembler Version 5.00 4/
79  Symb
80
81  Macros:
82
83      Name Lines
84
85  QQ . . . . . 10
86
87  Segments and Groups:
88
89      Name Length Align Combine Class
90
91  CODE . . . . . 003F PARA NONE
92  DATA . . . . . 0010 PARA NONE
93
94  Symbols:
95
96      Name Type Value Attr
97
98
99  ID . . . . . L BYTE 0000 DATA
100
101  J1 . . . . . L NEAR 0013 CODE
102
103  QQ0 . . . . . L NEAR 0025 CODE
104  QQ1 . . . . . L NEAR 0031 CODE
105
106  RL . . . . . L NEAR 0020 CODE
107
108  START . . . . . L NEAR 0000 CODE
109
110  X . . . . . L BYTE 000A DATA
111
112  Y . . . . . L BYTE 000F DATA
113
114  @FILENAME . . . . . TEXT lab32
115
116
117  55 Source Lines
118  96 Total Lines
119  17 Symbols
120
121  50386 + 465998 Bytes symbol space free
122
123  0 Warning Errors
124  0 Severe Errors
125
```

(2) 场景 1 的反汇编的截图

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

0 Warning Errors
0 Severe Errors

Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

LINK : warning L4021: no stack segment
Generate lab32.exe successfully.

C:\LEARN>debug lab32.exe
-u
077F:0000 B87E07      MOV     AX,077E
077F:0003 8ED8             MOV     DS,AX
077F:0005 BB0500      MOV     BX,0005
077F:0008 B90000      MOV     CX,0000
077F:000B 83FB05      CMP     BX,+05
077F:000E 7E03        JLE     0013
077F:0010 B90100      MOV     CX,0001
077F:0013 B80100      MOV     AX,0001
077F:0016 83F900      CMP     CX,+00
077F:0019 740A        JZ      0025
077F:001B 83F901      CMP     CX,+01
077F:001E 7411        JZ      0031
```

(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

077F:0019 740A        JZ      0025
077F:001B 83F901      CMP     CX,+01
077F:001E 7411        JZ      0031
-t
AX=077E BX=0000 CX=0153 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076E ES=076E SS=077D CS=077F IP=0003  NU UP EI PL NZ NA PO NC
077F:0003 8ED8             MOV     DS,AX
-t
AX=077E BX=0000 CX=0153 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=077E ES=076E SS=077D CS=077F IP=0005  NU UP EI PL NZ NA PO NC
077F:0005 BB0500      MOV     BX,0005
-t
AX=077E BX=0005 CX=0153 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=077E ES=076E SS=077D CS=077F IP=0008  NU UP EI PL NZ NA PO NC
077F:0008 B90000      MOV     CX,0000
-d 0 9
077E:0000 32 32 32 33 32 31 31 39-34 36                2223211946
-d a f
077E:0000                31 31 39 34 36 4E                11946N
-d a e
077E:0000                31 31 39 34 36                11946
```

(4) 场景 2 的 .lst 文件的截图

```
1 Microsoft (R) Macro Assembler Version 5.00 4/21/24 14:48:29
2 Page 1-1
3
4
5 qq macro k
6     1
7     2
8     3 ife k - 1
9     4 rept 6
10    5 add ax,ax
11    6 endn
12    7 endif
13    8
14    9 ife k
15    10 rept 5
16    11 add ax,ax
17    12 endn
18    13 endif
19    14
20    15 endm
21    16
22    18 0000 data segment
23    19 0000 32 32 32 32 32 31 31 ID db '2223211946'
24    20 39 34 36
25    21
26    22 000A 32 32 32 32 32 31 31 X db '2223211946'
27    23 39 34 36
28    24 0014 4E Y db 'N'
29    25 0015
30    26 data ends
31    27 0000 code segment
32    28 assume cs:code,ds:data
33    29
34    30 0000 B8 ---- R start: mov ax,data
35    31 0003 BE D8 mov ds,ax
36    32
37    33 0005 BB 000A mov bx,offset Y - offset
38    34 0008 B9 0000 t X
39    35 000B B3 F9 05 cmp bx,5
40    36 000E 7E 03 jle jl
41    37 0010 B9 0001 mov cx,1
42    38 0013 B8 0001 jl: mov ax,1
43    39
44    40
45    41 0016 B3 F9 00 cmp cx,0
46    42 0019 74 0A je qd0
47    43 001B B3 F9 01 cmp cx,1
48    44 001E 74 11 je qe1
49    45
50    46 0020 B8 4C00 r1: mov ax,4c00h
51    47 0023 CD 21 int 21h
52    48
53    49 0025 end
54
55 50 qq 0
56 51 0025 03 C0 2 add ax,ax
57 52 0027 03 C0 2 add ax,ax
58 53 0029 03 C0 2 add ax,ax
59
60 Microsoft (R) Macro Assembler Version 5.00 4/21/24 14:48:25
61 Page 1-2
62
63 54 002B 03 C0 2 add ax,ax
64 55 002D 03 C0 2 add ax,ax
65 56 002F EB EF jmp r1
66 57 0031
67 58 qq 1
68 59 0031 03 C0 2 add ax,ax
69 60 0033 03 C0 2 add ax,ax
70 61 0035 03 C0 2 add ax,ax
71 62 0037 03 C0 2 add ax,ax
72 63 0039 03 C0 2 add ax,ax
73 64 003B 03 C0 2 add ax,ax
74 65 003D EB E1 jmp r1
75 66
76 67 003F code ends
77 68 end start
78
79 Microsoft (R) Macro Assembler Version 5.00 4/21/24 14:48:25
80 Symbol=1
81
82 Macros:
83
84 Name Lines
85
86 Q0 . . . . . 10
87
88 Segments and Groups:
89
90 Name Length Align Combine Class
91
92 CODE . . . . . 003F PARA NONE
93 DATA . . . . . 0015 PARA NONE
94
95 Symbols:
96
97 Name Type Value Attr
98
99 ID . . . . . L BYTE 0000 DATA
100
101 J1 . . . . . L NEAR 0013 CODE
102
103 Q00 . . . . . L NEAR 0025 CODE
104 Q01 . . . . . L NEAR 0031 CODE
105
106 8 % 未找到相关问题
```

```
106
107 RL . . . . . L NEAR 0020 CODE
108
109 START . . . . . L NEAR 0000 CODE
110
111 X . . . . . L BYTE 000A DATA
112
113 Y . . . . . L BYTE 0014 DATA
114
115 @FILENAME . . . . . TEXT lab32
116
117
118 55 Source Lines
119 86 Total Lines
120 17 Symbols
121
122 50386 + 465998 Bytes symbol space free
123
124 0 Warning Errors
125 0 Severe Errors
126
```

(5) 场景 2 的反汇编的截图


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

Warning Errors
Severe Errors

Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

LINK : warning L4021: no stack segment
Generate lab32.exe successfully.

C:\LEARN>debug lab32.exe
-u
0780:0000 B87E07      MOV     AX,077E
0780:0003 8ED8        MOV     DS,AX
0780:0005 BB0A00      MOV     BX,000A
0780:0008 B90000      MOV     CX,0000
0780:000B 83FB05      CMP     BX,+05
0780:000E 7E03        JLE     0013
0780:0010 B90100      MOV     CX,0001
0780:0013 B80100      MOV     AX,0001
0780:0016 83F900      CMP     CX,+00
0780:0019 740A        JZ      0025
0780:001B 83F901      CMP     CX,+01
0780:001E 7411        JZ      0031
```

(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

0780:0003 8ED8        MOV     DS,AX
0780:0005 BB0A00      MOV     BX,000A
0780:0008 B90000      MOV     CX,0000
0780:000B 83FB05      CMP     BX,+05
0780:000E 7E03        JLE     0013
0780:0010 B90100      MOV     CX,0001
0780:0013 B80100      MOV     AX,0001
0780:0016 83F900      CMP     CX,+00
0780:0019 740A        JZ      0025
0780:001B 83F901      CMP     CX,+01
0780:001E 7411        JZ      0031
-t
AX=077E BX=0000 CX=0163 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=076E ES=076E SS=077D CS=0780 IP=0003  NU UP EI PL NZ NA PO NC
0780:0003 8ED8        MOV     DS,AX
-t
AX=077E BX=0000 CX=0163 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=077E ES=076E SS=077D CS=0780 IP=0005  NU UP EI PL NZ NA PO NC
0780:0005 BB0A00      MOV     BX,000A
-d a 13
077E:0000                                     32 32 32 33 32 31      222321
077E:0010 31 39 34 36                                     1946
```

(7) 源代码

1. qq macro k
- 2.
3. ife k - 1
4. rept 6


```

5.         add  ax,ax
6.         endm
7.     endif
8.
9.     ife k
10.        rept 5
11.        add  ax,ax
12.        endm
13.    endif
14.
15. endm
16.
17.
18. data segment
19.     ID  db  '2223211946'
20.     ;X  db  '11946'
21.     X   db  '2223211946'
22.     Y   db  'N'
23. data ends
24.
25. code segment
26.     assume cs:code,ds:data
27.
28. start:  mov  ax,data
29.        mov  ds,ax
30.
31.        mov  bx,offset Y - offset X
32.        mov  cx,0
33.        cmp  bx,5
34.        jle  j1
35.        mov  cx,1
36.    j1:  mov  ax,1
37.
38.
39.        cmp  cx,0
40.        je   qq0
41.        cmp  cx,1
42.        je   qq1
43.
44.    r1:  mov  ax,4c00h
45.        int  21h
46.
47.    qq0:
48.        qq  0

```

```
49.         jmp r1
50.     qq1:
51.         qq 1
52.         jmp r1
53.
54. code ends
55. end start
```