

南京林业大学



汇编语言上机实验

任务书

实验五 编写、调试具有多个段的程序

一. 实验目的

编写、调试具有多个段的程序

二. 实验内容

1. 将下面的程序编译、链接，用 Debug 加载、跟踪，然后回答问题。

```
assume cs:code, ds:data, ss:stack

data segment
    dw 0123h, 0456h, 0789h, 0defh, 0fedh, 0cbah, 0987h
data ends

stack segment
    dw 0, 0, 0, 0, 0, 0, 0, 0
stack ends

code segment
start:  mov ax, stack
        mov ss, ax
        mov sp, 16
        mov ax, data
        mov ds, ax
        push ds:[0]
        push ds:[2]
        pop ds:[2]
```

```

pop ds:[0]

mov ax, 4c00h

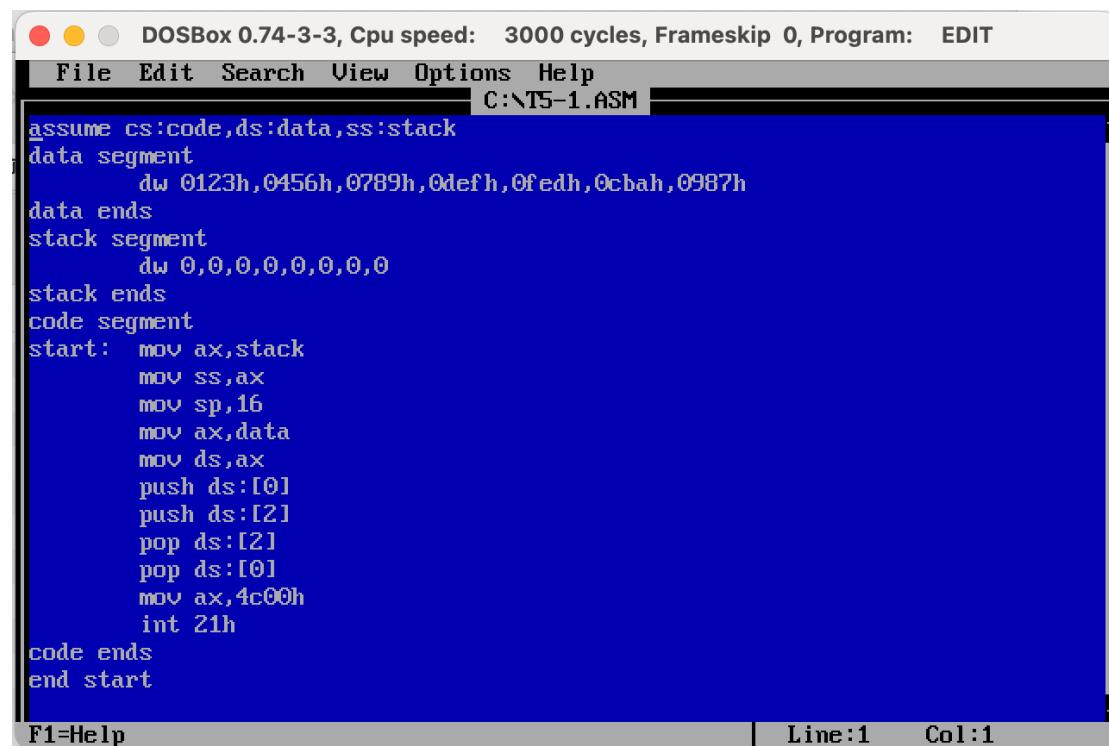
int 21h

code ends

end start

```

输入程序：



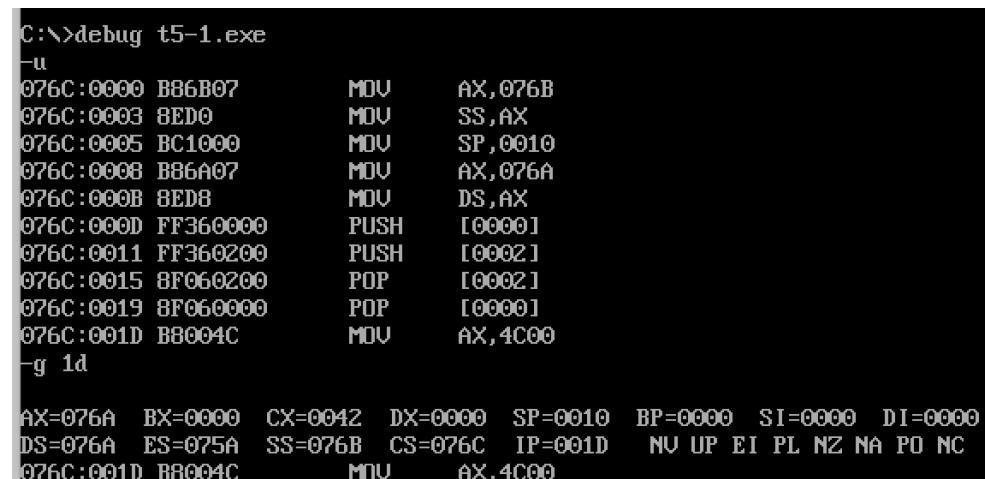
```

DOSBox 0.74-3-3, Cpu speed: 3000 cycles, Frameskip 0, Program: EDIT
File Edit Search View Options Help
C:\NT5-1.ASM

assume cs:code,ds:data,ss:stack
data segment
    dw 0123h,0456h,0789h,0defh,0fedh,0cbah,0987h
data ends
stack segment
    dw 0,0,0,0,0,0,0,0
stack ends
code segment
start: mov ax,stack
        mov ss,ax
        mov sp,16
        mov ax,data
        mov ds,ax
        push ds:[0]
        push ds:[2]
        pop ds:[2]
        pop ds:[0]
        mov ax,4c00h
        int 21h
code ends
end start

```

执行程序：



```

C:\>debug t5-1.exe
-u
076C:0000 B86B07      MOV     AX,076B
076C:0003 8ED0      MOV     SS,AX
076C:0005 BC1000      MOV     SP,0010
076C:0008 B86A07      MOV     AX,076A
076C:000B 8ED8      MOV     DS,AX
076C:000D FF360000    PUSH    [0000]
076C:0011 FF360200    PUSH    [0002]
076C:0015 8F060200    POP     [0002]
076C:0019 8F060000    POP     [0000]
076C:001D B8004C      MOV     AX,4C00
-g 1d

AX=076A  BX=0000  CX=0042  DX=0000  SP=0010  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=076B  CS=076C  IP=001D  NV UP EI PL NZ NA PU NC
076C:001D B8004C      MOV     AX,4C00

```

```

-d 076a:0000
076A:0000 23 01 56 04 89 07 EF 0D-ED OF BA 0C 87 09 00 00 #.U.....
076A:0010 00 00 00 00 00 00 00-00 00 1D 00 6C 07 A3 01 .....1...
076A:0020 B8 6B 07 8E D0 BC 10 00-B8 6A 07 8E D8 FF 36 00 .k.....j...6.
076A:0030 00 FF 36 02 00 8F 06 02-00 8F 06 00 00 B8 00 4C ..6.....L
076A:0040 CD 21 00 00 00 00 00-00 00 00 00 00 00 00 00 00 ..!.....
076A:0050 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00 .....
076A:0060 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00 .....
076A:0070 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 00 .....

```

1) CPU 执行程序, 程序返回前, data 段中的数据为多少?

Data 中的数据是: 23 01 56 04 89 07 EF 0D ED OF BA 0C 87 09

2) CPU 执行程序, 程序返回前, cs=0042, ss=076B, ds=076A。

3) 设程序加载后, code 段段地址为 X, 则 data 段段地址为: X-2, stack 段段地址为 X-1。

2. 将下面的程序编译、链接, 用 Debug 加载、跟踪, 然后回答问题。

```
assume cs:code, ds:data, ss:stack
```

```
data segment
```

```
    dw 0123h, 0456h
```

```
data ends
```

```
stack segment
```

```
    dw 0, 0
```

```
stack ends
```

```
code segment
```

```
start: mov ax, stack
```

```
    mov ss, ax
```

```
    mov sp, 16
```

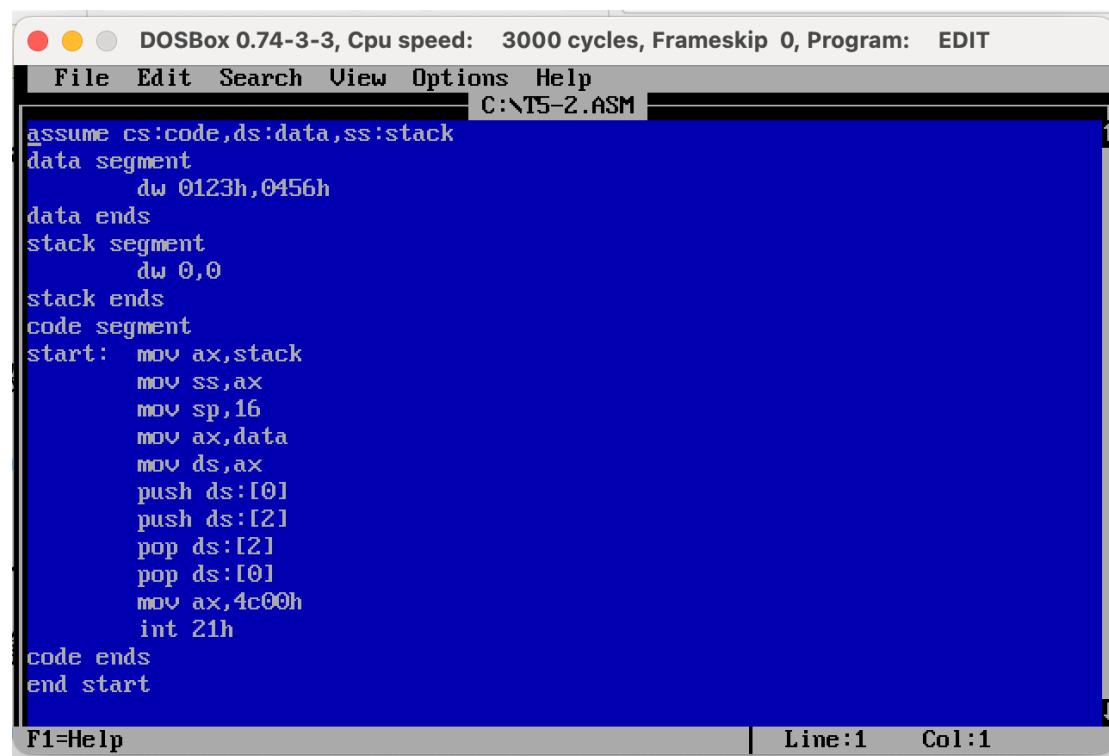
```
    mov ax, data
```

```
    mov ds, ax
```

```
    push ds:[0]
```

```
push ds:[2]
pop ds:[2]
pop ds:[0]
mov ax, 4c00h
int 21h
code ends
end start
```

输入程序：



The screenshot shows the DOSBox interface with the title bar "DOSBox 0.74-3-3, Cpu speed: 3000 cycles, Frameskip 0, Program: EDIT" and the file path "C:\NT5-2.ASM". The main window displays the following assembly code:

```
assume cs:code,ds:data,ss:stack
data segment
    dw 0123h,0456h
data ends
stack segment
    dw 0,0
stack ends
code segment
start: mov ax,stack
    mov ss,ax
    mov sp,16
    mov ax,data
    mov ds,ax
    push ds:[0]
    push ds:[2]
    pop ds:[2]
    pop ds:[0]
    mov ax,4c00h
    int 21h
code ends
end start
```

The status bar at the bottom shows "F1=Help" and "Line:1 Col:1".

执行程序：

```
C:\>debug t5-2.exe
-u
076C:0000 B86B07      MOV     AX,076B
076C:0003 8ED0      MOV     SS,AX
076C:0005 BC1000      MOV     SP,0010
076C:0008 B86A07      MOV     AX,076A
076C:000B 8ED8      MOV     DS,AX
076C:000D FF360000      PUSH    [0000]
076C:0011 FF360200      PUSH    [0002]
076C:0015 8F060200      POP     [0002]
076C:0019 8F060000      POP     [0000]
076C:001D B8004C      MOV     AX,4C00
-g 1d

AX=076A  BX=0000  CX=0042  DX=0000  SP=0010  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=076B  CS=076C  IP=001D  NU UP EI PL NZ NA PO NC
076C:001D B8004C      MOV     AX,4C00
```

1) CPU 执行程序, 程序返回前, data 段中的数据是多少?

```
076A:0000 23 01 56 04 00 00 00 00-00 00 00 00 00 00 00 00 #.U.....
-076A:0010 00 00 00 00 00 00 00 00-00 00 1D 00 6C 07 A3 01 .....1...
076A:0020 B8 6B 07 8E D0 BC 10 00-B8 6A 07 8E D8 FF 36 00 .k.....j...6.
076A:0030 00 FF 36 02 00 8F 06 02-00 8F 06 00 00 B8 00 4C ..6.....L
076A:0040 CD 21 F0 0B F0 74 03 E9-B8 00 83 3E CE 49 01 75 .!...t....>.I.u
076A:0050 50 A1 2C 4B 33 D2 E8 01-07 8B F0 89 56 FE 0B D0 P..K3.....U...
076A:0060 75 06 B8 1F 00 E9 9A 00-BB 1D 2A E4 8A 47 04 50 u.....*..G.P
076A:0070 8A 45 10 03 45 02 50 8A-45 11 2A E4 03 45 04 50 .E..E.P.E.*..E.P
```

为 23 01 56 04

2) CPU 执行程序, 程序返回前, cs=076C, ss=076B, ds=076A

3) 设程序加载后, code 段段地址是 X, 则 data 段的段地址为 X-2, stack 段段地址为 X-1。

3. 将下面的程序编译、链接, 用 Debug 加载、跟踪, 然后回答问题。

```
assume cs:code, ds:data, ss:stack

code segment

start:  mov ax, stack

        mov ss, ax

        mov sp, 16

        mov ax, data

        mov ds, ax
```

```
push ds:[0]
push ds:[2]
pop ds:[2]
pop ds:[0]
mov ax, 4c00h
int 21h

code ends

data segment
dw 0123h, 0456h

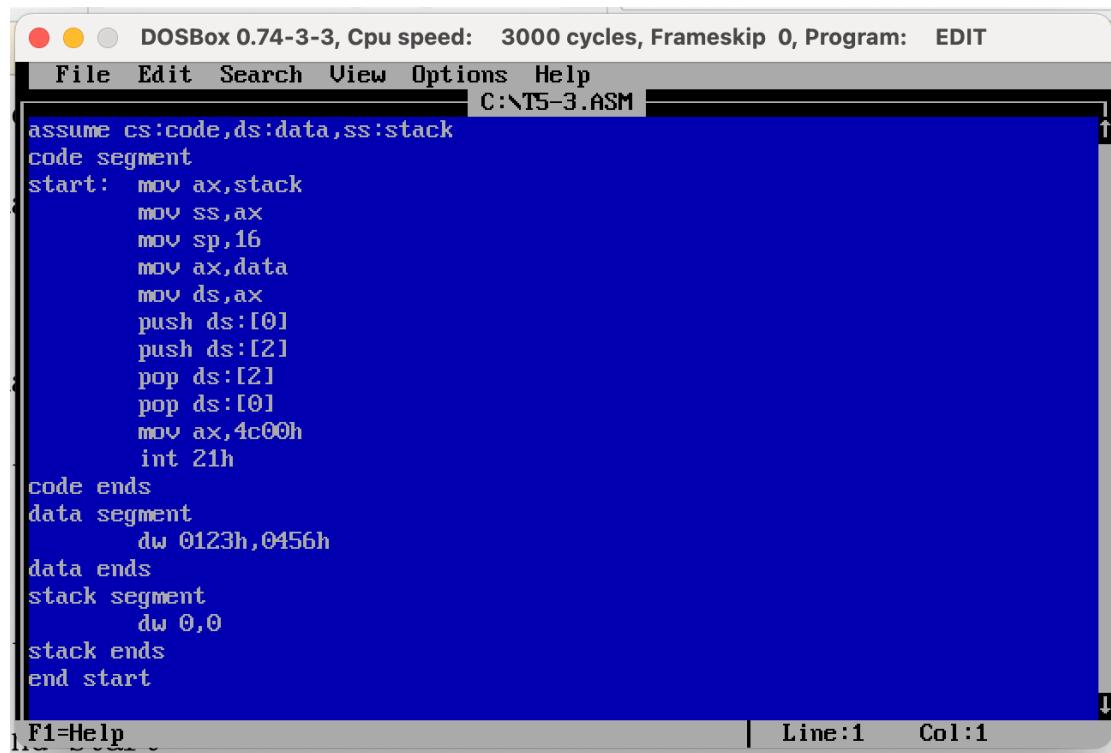
data ends

stack segment
dw 0, 0

stack ends

end start
```

输入程序：

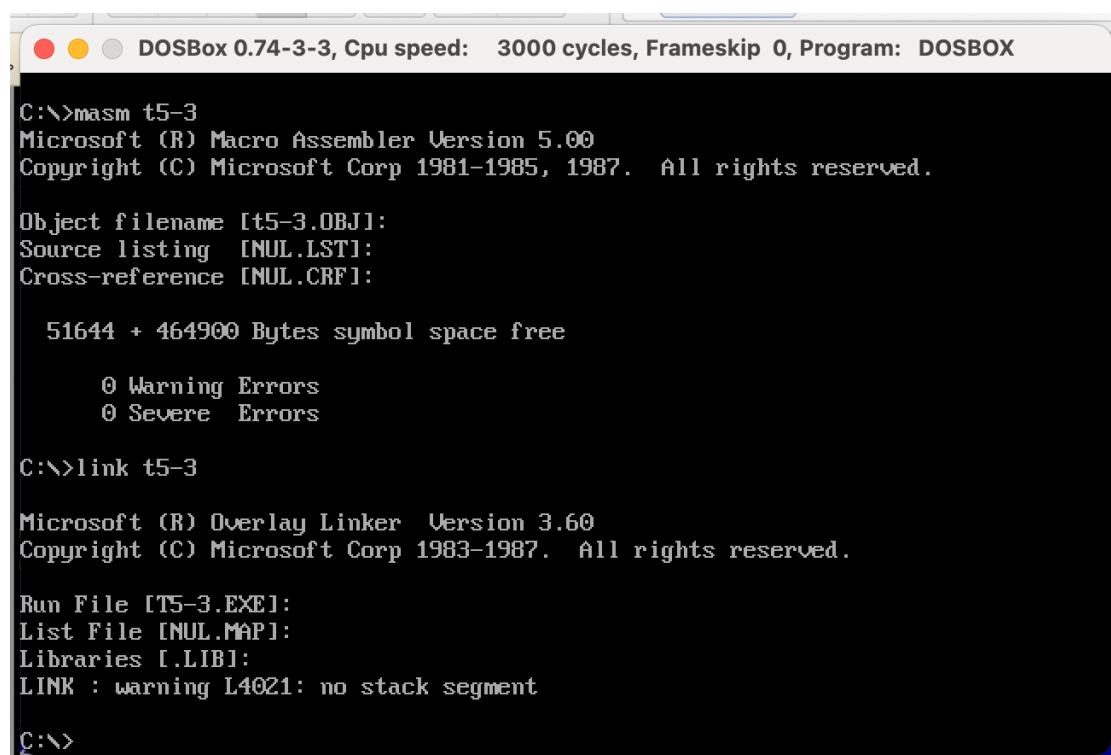


DOSBox 0.74-3-3, Cpu speed: 3000 cycles, Frameskip 0, Program: EDIT
File Edit Search View Options Help C:\NT5-3.ASM

```
assume cs:code,ds:data,ss:stack
code segment
start: mov ax,stack
       mov ss,ax
       mov sp,16
       mov ax,data
       mov ds,ax
       push ds:[0]
       push ds:[2]
       pop ds:[2]
       pop ds:[0]
       mov ax,4c00h
       int 21h
code ends
data segment
       dw 0123h,0456h
data ends
stack segment
       dw 0,0
stack ends
end start
```

F1=Help | Line:1 Col:1

编译并链接程序：



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

```
C:\>masm t5-3
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

Object filename [t5-3.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

51644 + 464900 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\>link t5-3

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

Run File [T5-3.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>
```

用 debug 执行程序：

```
C:\>debug t5-3.exe
F-u
I 076A:0000 B86E07      MOV      AX,076E
I 076A:0003 8ED0      MOV      SS,AX
I 076A:0005 BC1000      MOV      SP,0010
I 076A:0008 B86D07      MOV      AX,076D
I 076A:000B 8ED8      MOV      DS,AX
I 076A:000D FF360000      PUSH    [0000]
I 076A:0011 FF360200      PUSH    [0002]
I 076A:0015 8F060200      POP     [0002]
I 076A:0019 8F060000      POP     [0000]
I 076A:001D B8004C      MOV      AX,4C00
-g 1d
AX=076D  BX=0000  CX=0044  DX=0000  SP=0010  BP=0000  SI=0000  DI=0000
DS=076D  ES=075A  SS=076E  CS=076A  IP=001D  NU UP EI PL NZ NA PO NC
076A:001D B8004C      MOV      AX,4C00
```

1) CPU 执行程序, 程序返回前, data 段中的数据为多少?

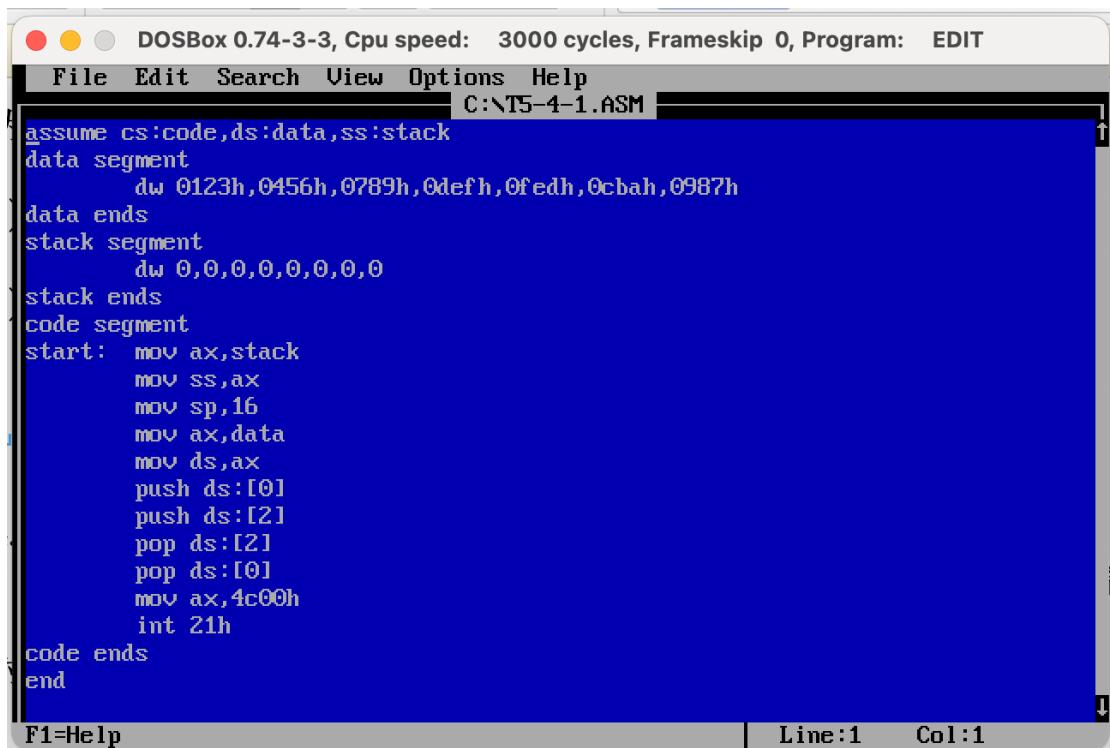
为: 23 01 56 04

2) CPU 执行程序, 程序返回前, $cs=076A$, $ss=076E$, $ds=076D$

3) 设程序加载后, code 段的段地址是 X, 则 data 段的段地址为 X+3, stack 段段地址为 X+4。

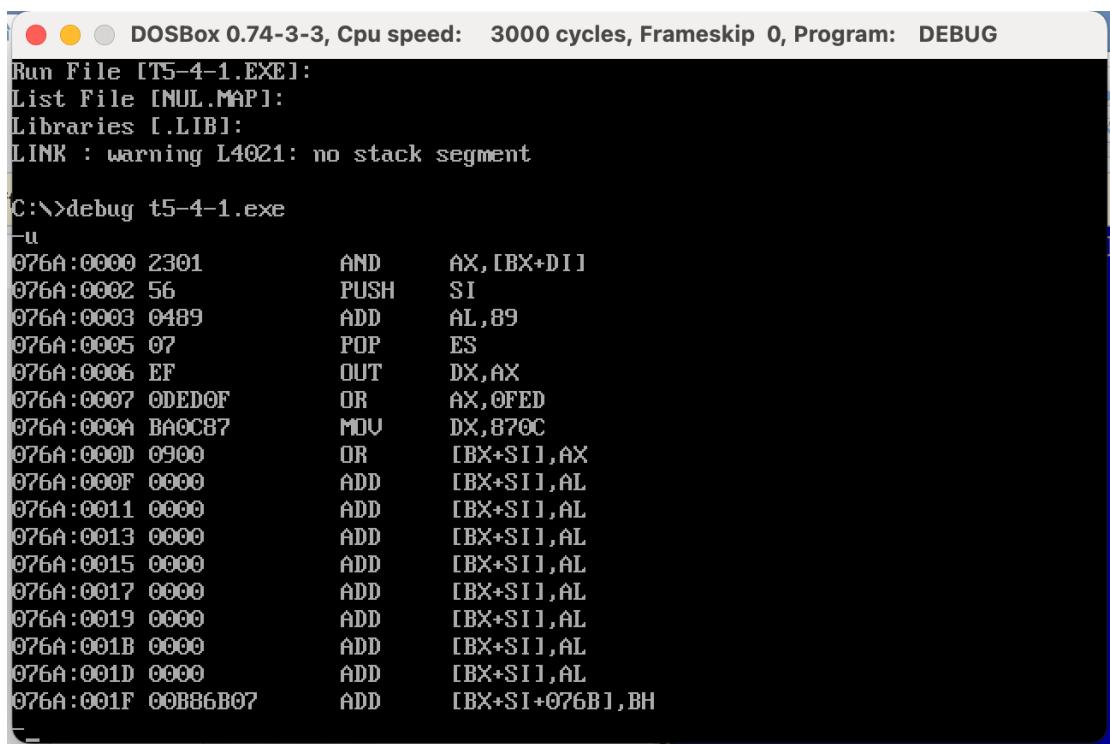
4. 如果将(1)(2)(3)题中的最后一条伪指令“end start”改为“end”(也就是说,不指明程序的入口),则哪个程序仍然可以正确执行?请说明原因。

对于 1 题，编写 T5-4-1.asm：



```
assume cs:code,ds:data,ss:stack
data segment
    dw 0123h,0456h,0789h,0defh,0fedh,0cbah,0987h
data ends
stack segment
    dw 0,0,0,0,0,0,0,0
stack ends
code segment
start: mov ax,stack
        mov ss,ax
        mov sp,16
        mov ax,data
        mov ds,ax
        push ds:[0]
        push ds:[2]
        pop ds:[2]
        pop ds:[0]
        mov ax,4c00h
        int 21h
code ends
end
```

编译、链接、用 debug 查看代码：



```
Run File [T5-4-1.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>debug t5-4-1.exe
-u
076A:0000 2301      AND    AX,[BX+DI]
076A:0002 56        PUSH   SI
076A:0003 0489      ADD    AL,89
076A:0005 07        POP    ES
076A:0006 EF        OUT    DX,AX
076A:0007 0DED0F    OR     AX,0FED
076A:000A BA0C87    MOV    DX,870C
076A:000D 0900      OR     [BX+SI],AX
076A:000F 0000      ADD    [BX+SI],AL
076A:0011 0000      ADD    [BX+SI],AL
076A:0013 0000      ADD    [BX+SI],AL
076A:0015 0000      ADD    [BX+SI],AL
076A:0017 0000      ADD    [BX+SI],AL
076A:0019 0000      ADD    [BX+SI],AL
076A:001B 0000      ADD    [BX+SI],AL
076A:001D 0000      ADD    [BX+SI],AL
076A:001F 00B86B07    ADD    [BX+SI+076B],BH
```

执行：

```
-g 1f  
AX=2FFF BX=0000 CX=0042 DX=870C SP=0000 BP=0000 SI=0000 DI=0000  
DS=075A ES=0000 SS=0769 CS=076A IP=001F NV UP EI NG NZ AC PO CY  
076A:001F 00B86B07 ADD [BX+SI+076B],BH DS:076B=58
```

程序没有正常执行。

对于 2 题，编写 T5-4-2.asm：



The screenshot shows the DOSBox 0.74-3-3 interface with the assembly code for T5-4-2.asm. The code initializes segments, sets up the stack, and performs a series of pushes and pops to demonstrate segment switching. The assembly code is as follows:

```
assume cs:code,ds:data,ss:stack  
data segment  
    dw 0123h,0456h  
data ends  
stack segment  
    dw 0,0  
stack ends  
code segment  
start:  mov ax,stack  
        mov ss,ax  
        mov sp,16  
        mov ax,data  
        mov ds,ax  
        push ds:[0]  
        push ds:[2]  
        pop ds:[2]  
        pop ds:[0]  
        mov ax,4c00h  
        int 21h  
code ends  
end
```

编译、链接、用 debug 查看代码：

```
dosbox 0.74-3-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
Run File [T5-4-2.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

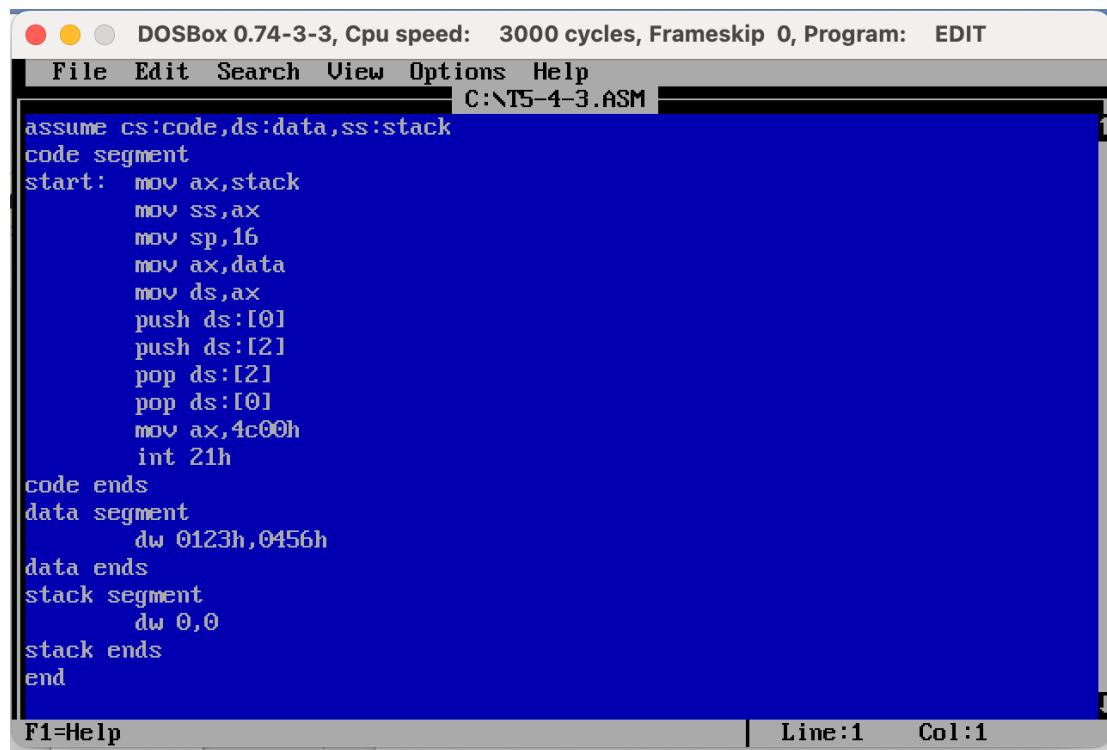
C:\>debug t5-4-2.exe
-u
076A:0000 2301      AND     AX,[BX+DI]
076A:0002 56        PUSH    SI
076A:0003 0400      ADD     AL,00
076A:0005 0000      ADD     [BX+SI],AL
076A:0007 0000      ADD     [BX+SI],AL
076A:0009 0000      ADD     [BX+SI],AL
076A:000B 0000      ADD     [BX+SI],AL
076A:000D 0000      ADD     [BX+SI],AL
076A:000F 0000      ADD     [BX+SI],AL
076A:0011 0000      ADD     [BX+SI],AL
076A:0013 0000      ADD     [BX+SI],AL
076A:0015 0000      ADD     [BX+SI],AL
076A:0017 0000      ADD     [BX+SI],AL
076A:0019 0000      ADD     [BX+SI],AL
076A:001B 0000      ADD     [BX+SI],AL
076A:001D 0000      ADD     [BX+SI],AL
076A:001F 00B86B07  ADD     [BX+SI+076B],BH
```

执行：

```
-g 1f
AX=20CD  BX=0000  CX=0042  DX=0000  SP=FFFE  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=001F  NV UP EI PL NZ AC PE CY
076A:001F 00B86B07  ADD     [BX+SI+076B],BH          DS:076B=58
```

程序没有正常执行。

对于 3 题，编写 T5-4-3.asm：

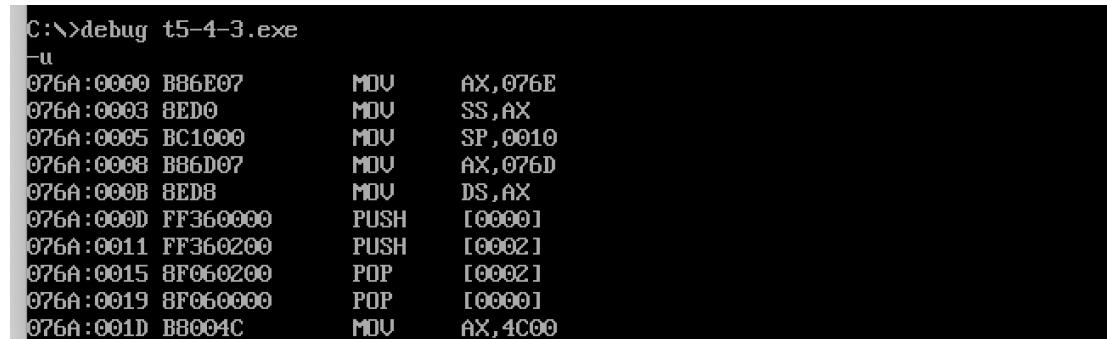


DOSBox 0.74-3-3, Cpu speed: 3000 cycles, Frameskip 0, Program: EDIT
File Edit Search Options Help C:\T5-4-3.ASM

```
assume cs:code,ds:data,ss:stack
code segment
start:  mov ax,stack
        mov ss,ax
        mov sp,16
        mov ax,data
        mov ds,ax
        push ds:[0]
        push ds:[2]
        pop ds:[2]
        pop ds:[0]
        mov ax,4c00h
        int 21h
code ends
data segment
        dw 0123h,0456h
data ends
stack segment
        dw 0,0
stack ends
end
```

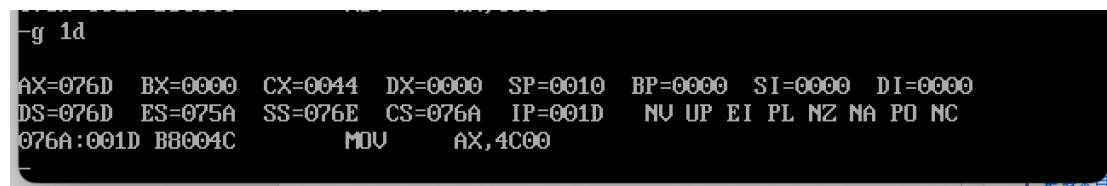
F1=Help | Line:1 Col:1

编译、链接、用 debug 查看代码：



```
C:\>debug t5-4-3.exe
-u
076A:0000 B86E07      MOV     AX,076E
076A:0003 8ED0      MOV     SS,AX
076A:0005 BC1000      MOV     SP,0010
076A:0008 B86D07      MOV     AX,076D
076A:000B 8ED8      MOV     DS,AX
076A:000D FF360000      PUSH    [0000]
076A:0011 FF360200      PUSH    [0002]
076A:0015 8F060200      POP     [0002]
076A:0019 8F060000      POP     [0000]
076A:001D B8004C      MOV     AX,4C00
```

执行：



```
-g 1d
AX=076D  BX=0000  CX=0044  DX=0000  SP=0010  BP=0000  SI=0000  DI=0000
DS=076D  ES=075A  SS=076E  CS=076A  IP=001D  NV UP EI PL NZ NA PO NC
076A:001D B8004C      MOV     AX,4C00
```

程序正确执行了。

结论：只有 3 可以正确执行，1 2 不可以正确执行。

5. 程序如下，编写 code 段段代码，将 a 段和 b 段段数据一次想家，并将结果存在 c 段中。

```
assume cs:code

a segment
db 1, 2, 3, 4, 5, 6, 7, 8
a ends

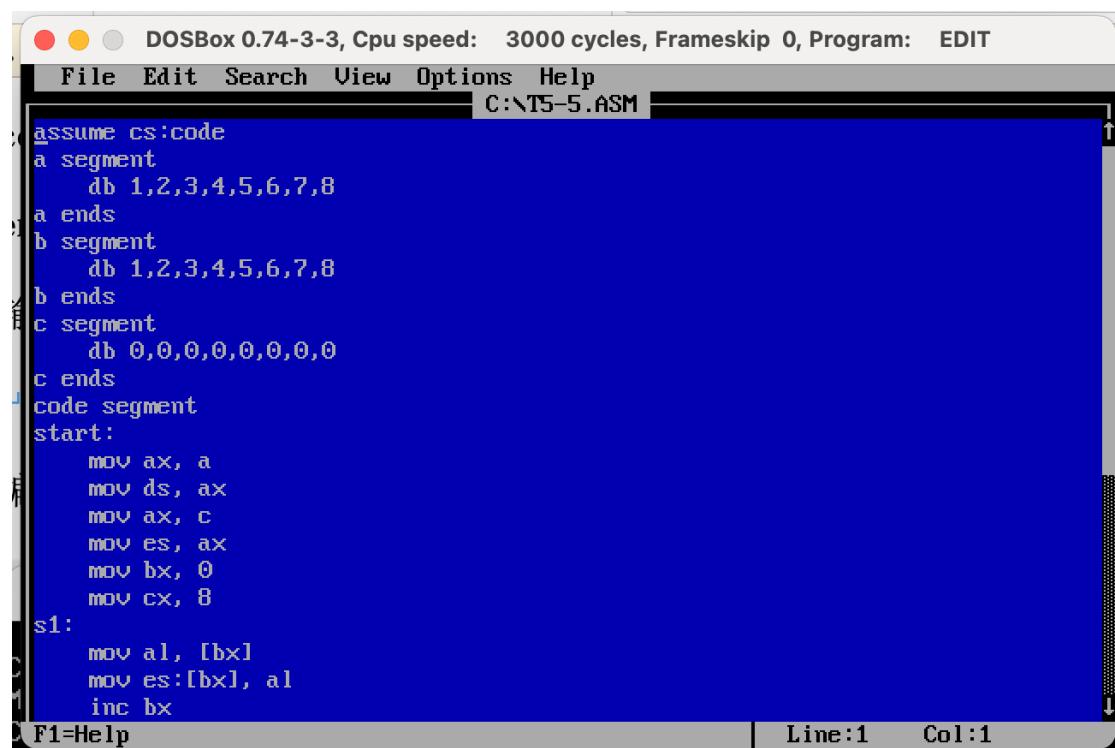
b segment
db 1, 2, 3, 4, 5, 6, 7, 8
b ends

c segment
db 0, 0, 0, 0, 0, 0, 0, 0
c ends

code segment
start:    mov ax, A
          mov ds, ax
          mov ax, B
          mov es, ax
          mov ax, C
          mov ss, ax
          mov sp, 8
          mov cx, 8
          mov si, 0
s:    mov al, [si]
          add al, es:[si]
          mov ss:[si], al
```

```
inc si
loop s
mov ah, 4ch
int 21h
code ends
end start
```

输入程序：



The screenshot shows the DOSBox 0.74-3-3 interface with the following details:

- Window title: DOSBox 0.74-3-3, Cpu speed: 3000 cycles, Frameskip 0, Program: EDIT
- File menu: File, Edit, Search, View, Options, Help
- File path: C:\NT5-5.ASM
- Code area (scrollable):

```
assume cs:code
a segment
    db 1,2,3,4,5,6,7,8
a ends
b segment
    db 1,2,3,4,5,6,7,8
b ends
c segment
    db 0,0,0,0,0,0,0,0
c ends
code segment
start:
    mov ax, a
    mov ds, ax
    mov ax, c
    mov es, ax
    mov bx, 0
    mov cx, 8
s1:
    mov al, [bx]
    mov es:[bx], al
    inc bx
```
- Bottom status bar: F1=Help | Line:1 Col:1

编译并链接程序：

```
C:\>
C:\>link t5-5.asm
Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

Run File [T5-5.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
T5-5.ASM : fatal error L1101: invalid object module
pos: 1 Record type: 61
(
C:\>link t5-5
(
Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

Run File [T5-5.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment
(
C:\>
```

运行程序：

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

Run File [T5-5.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>debug t5-5.exe
-u
076D:0000 B86A07      MOV     AX,076A
076D:0003 8ED8      MOV     DS,AX
076D:0005 B86C07      MOV     AX,076C
076D:0008 8EC0      MOV     ES,AX
076D:000A BB0000      MOV     BX,0000
076D:000D B90800      MOV     CX,0008
076D:0010 8A07      MOV     AL,[BX]
076D:0012 26      ES:
076D:0013 8807      MOV     [BX],AL
076D:0015 43      INC     BX
076D:0016 E2F8      LOOP    0010
076D:0018 B86B07      MOV     AX,076B
076D:001B 8ED8      MOV     DS,AX
076D:001D BB0000      MOV     BX,0000
```

```
-g
Program terminated normally
```

查看数据段 a、b 处内存的内容：

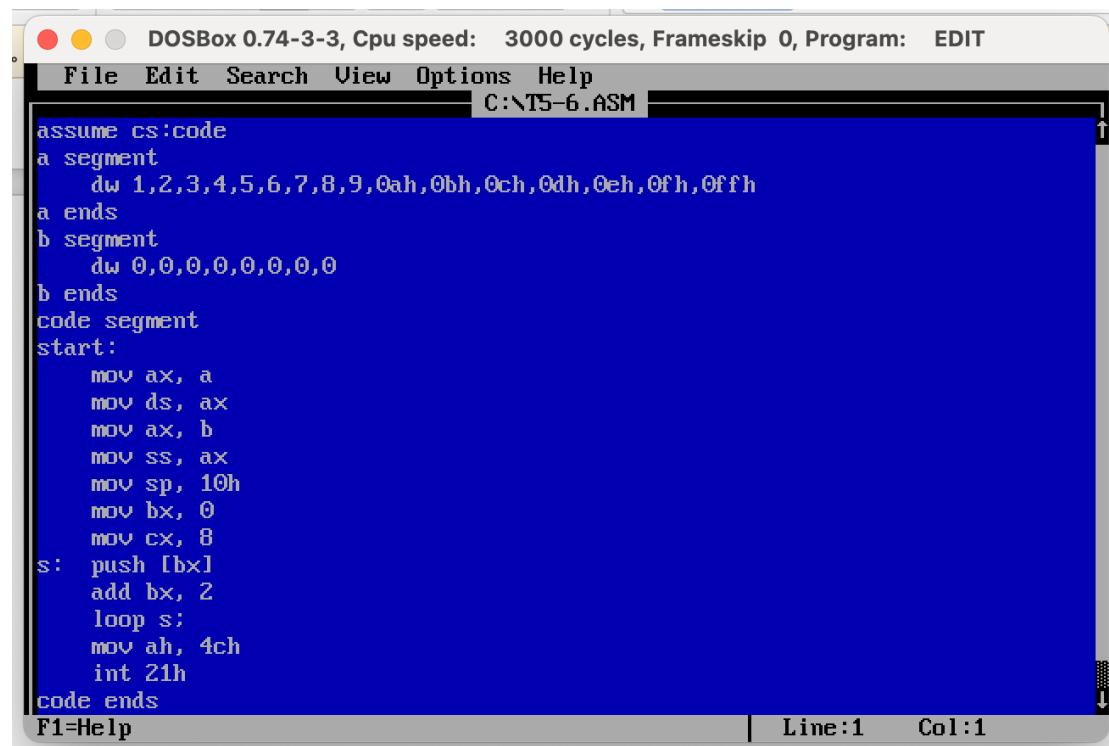
查看数据段 c 中的内容：

6. 程序如下, 编写 code 段中的代码, 用 push 指令将 a 段中的前 8 个字型数据, 逆序存储到 b 段中。

```
assume cs:code
a segment
dw 1,2,3,4,5,6,7,8,9,0ah,0bh,0ch,0dh,0eh,0fh,0ffh
a ends
b segment
dw 0,0,0,0,0,0,0,0,0
b ends
code segment
start:
    mov ax, a
    mov ds, ax
    mov ax, b
    mov ss, ax
    mov sp, 10h
    mov bx, 0
    mov cx, 8
s:   push [bx]
    add bx, 2
    loop s;
    mov ah, 4ch
```

```
int 21h
code ends
end start
```

输入代码:

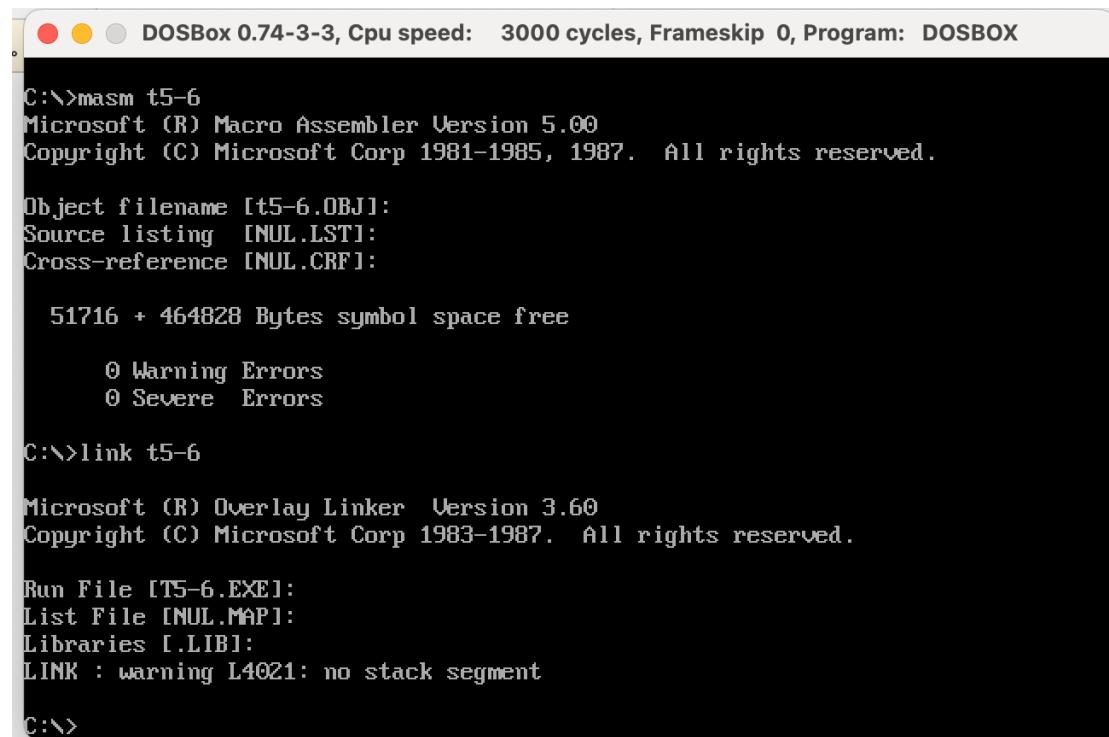


DOSBox 0.74-3-3, Cpu speed: 3000 cycles, Frameskip 0, Program: EDIT
File Edit Search View Options Help C:\T5-6.ASM

```
assume cs:code
a segment
    dw 1,2,3,4,5,6,7,8,9,0ah,0bh,0ch,0dh,0eh,0fh,0ffh
a ends
b segment
    dw 0,0,0,0,0,0,0,0
b ends
code segment
start:
    mov ax, a
    mov ds, ax
    mov ax, b
    mov ss, ax
    mov sp, 10h
    mov bx, 0
    mov cx, 8
s:   push [bx]
    add bx, 2
    loop s:
    mov ah, 4ch
    int 21h
code ends
F1=Help
```

Line:1 Col:1

编译并链接代码:



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

```
C:\>masm t5-6
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

Object filename [t5-6.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

51716 + 464828 Bytes symbol space free

    0 Warning Errors
    0 Severe Errors

C:\>link t5-6

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

Run File [T5-6.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>
```

运行代码:

```
Run File [T5-6.EXE]:  
List File [NUL.MAP]:  
Libraries [.LIB]:  
LINK : warning L4021: no stack segment  
  
C:\>debug t5-6.exe  
-u  
076D:0000 B86A07      MOV      AX,076A  
076D:0003 8ED8      MOV      DS,AX  
076D:0005 B86C07      MOV      AX,076C  
076D:0008 8ED0      MOV      SS,AX  
076D:000A BC1000      MOV      SP,0010  
076D:000D BB0000      MOV      BX,0000  
076D:0010 B90800      MOV      CX,0008  
076D:0013 FF37      PUSH     [BX]  
076D:0015 83C302      ADD      BX,+02  
076D:0018 E2F9      LOOP    0013  
076D:001A B44C      MOV      AH,4C  
076D:001C CD21      INT      21  
076D:001E C404      LES      AX,[SI]  
-g  
  
Program terminated normally  
-
```

查看代码段 a

```
Run File [T5-6.EXE]:  
List File [NUL.MAP]:  
Libraries [.LIB]:  
LINK : warning L4021: no stack segment  
  
076D:0003 8ED8      MOV      DS,AX  
076D:0005 B86C07      MOV      AX,076C  
076D:0008 8ED0      MOV      SS,AX  
076D:000A BC1000      MOV      SP,0010  
076D:000D BB0000      MOV      BX,0000  
076D:0010 B90800      MOV      CX,0008  
076D:0013 FF37      PUSH     [BX]  
076D:0015 83C302      ADD      BX,+02  
076D:0018 E2F9      LOOP    0013  
076D:001A B44C      MOV      AH,4C  
076D:001C CD21      INT      21  
076D:001E C404      LES      AX,[SI]  
-g  
  
Program terminated normally  
-d 076a:0000  
076A:0000 01 00 02 00 03 00 04 00-05 00 06 00 07 00 08 00  ....  
076A:0010 09 00 0A 00 0B 00 0C 00-0D 00 0E 00 0F 00 FF 00  ....  
076A:0020 08 00 07 00 06 00 05 00-04 00 03 00 02 00 01 00  ....  
076A:0030 B8 6A 07 8E D8 B8 6C 07-8E D0 BC 10 00 BB 00 00  .j...1....  
076A:0040 B9 08 00 FF 37 83 C3 02-E2 F9 B4 4C CD 21 C4 04  ...7...L.?  
076A:0050 3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A  =.t....^.&.G.*  
076A:0060 E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83  .@P.....RP..H.  
076A:0070 C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6  ..P....P..s....
```

查看代码段 b

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

```
076D:001A B44C      MDV    AH,4C
076D:001C CD21      INT    21
076D:001E C404      LES    AX,[SI]
-g
Program terminated normally
-d 076a:0000
076A:0000 01 00 02 00 03 00 04 00-05 00 06 00 07 00 08 00  ...
076A:0010 09 00 0A 00 0B 00 0C 00-0D 00 0E 00 0F 00 FF 00  ...
076A:0020 08 00 07 00 06 00 05 00-04 00 03 00 02 00 01 00  ...
076A:0030 B8 6A 07 8E D8 B8 6C 07-8E D0 BC 10 00 BB 00 00  .j..1...
076A:0040 B9 08 00 FF 37 83 C3 02-E2 F9 B4 4C CD 21 C4 04  ...?...L.?
076A:0050 3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A  =.t....^.&G.#
076A:0060 E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83  .@P.....RP..H.
076A:0070 C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6  ..P....P..s...
-d 076c:0000
076C:0000 08 00 07 00 06 00 05 00-04 00 03 00 02 00 01 00  ...
076C:0010 B8 6A 07 8E D8 B8 6C 07-8E D0 BC 10 00 BB 00 00  .j..1...
076C:0020 B9 08 00 FF 37 83 C3 02-E2 F9 B4 4C CD 21 C4 04  ...?...L.?
076C:0030 3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A  =.t....^.&G.#
076C:0040 E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83  .@P.....RP..H.
076C:0050 C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6  ..P....P..s...
076C:0060 FA FE 81 E6 FF 00 C6 82-FB FE 00 2B C0 50 8D 86  .....+P...
076C:0070 FB FE 50 E8 08 6A 83 C4-04 0B C0 75 03 E9 A5 00  ..P..j....u...
```