安徽大学人工智能学院

实验报告

课程名称：《计算机组成原理与汇编语言》

专　　业：人工智能

学　　号：WA2214014

姓　　名：杨跃浙

指导老师：杜库

实验项目	实验 5-第五次上机实验			**实验次序**	5
实验地点	笃行南楼 A104	**参与人员**	杨跃浙	**实验日期**	5.08

一、实验目的

完成课本第 8 章部分习题

二、实验环境

Windows 2011， DOSBox

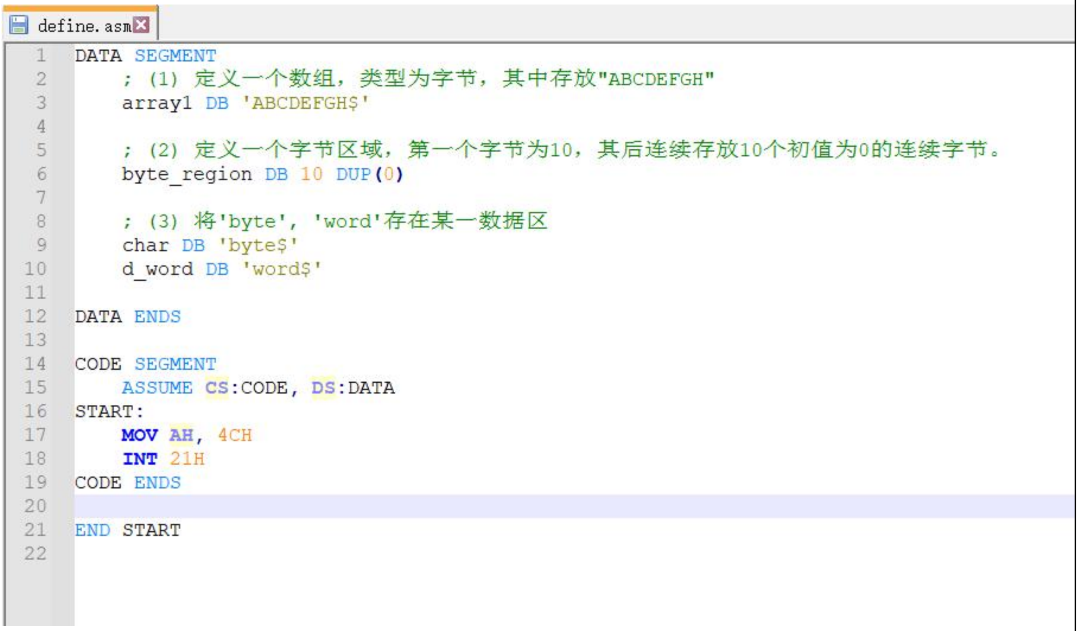
三、实验内容

8.1 按要求写出相应的数据定义语句

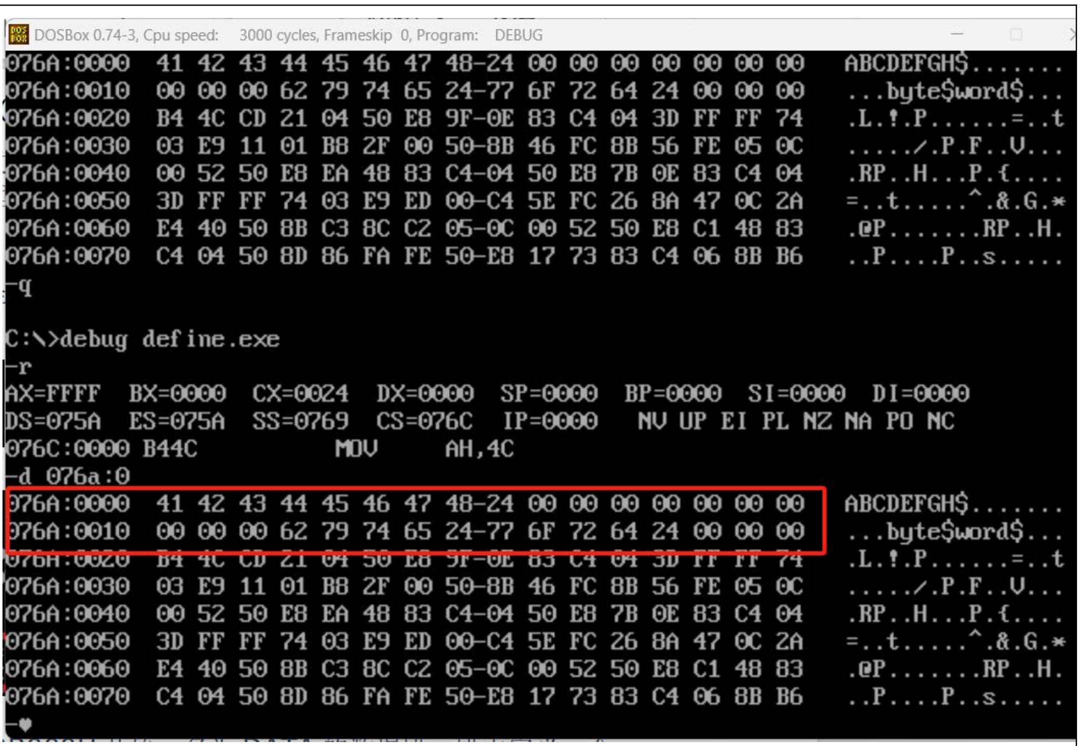
(1)定义一个数组，类型为字节，其中存放"ABCDEFGH"

(2)定义一个字节区域，第一个字节为 10， 其后连续存放 10 个初值为 0 的连续字节。

(3)将'byte', 'word'存在某一数据区

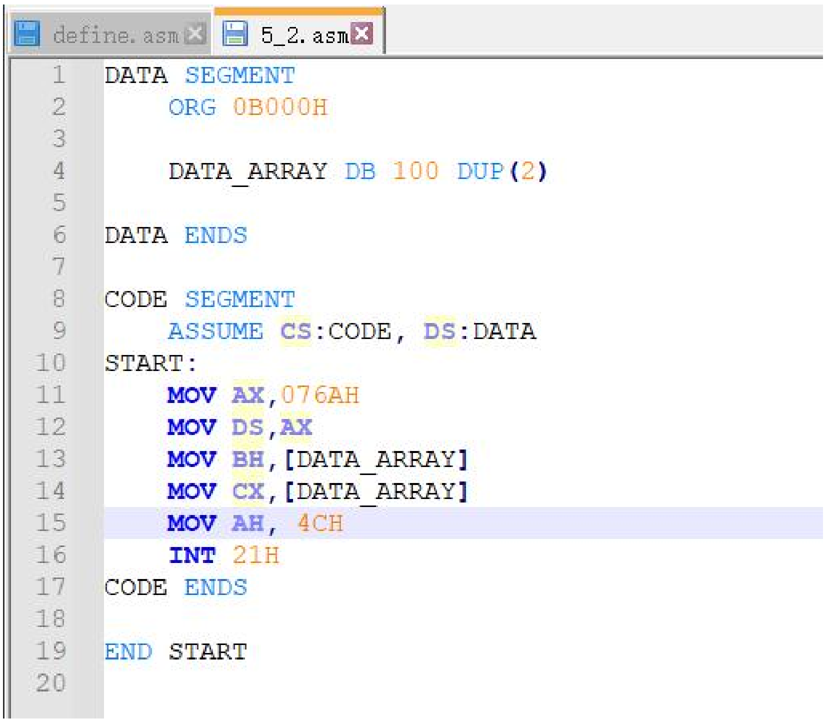


```
DATA SEGMENT
    ; (1) 定义一个数组，类型为字节，其中存放"ABCDEFGH"
    array1 DB 'ABCDEFGH$'

    ; (2) 定义一个字节区域，第一个字节为10，其后连续存放10个初值为0的连续字节。
    byte_region DB 10 DUP(0)

    ; (3) 将'byte', 'word'存在某一数据区
    char DB 'byte$'
    d_word DB 'word$'

DATA ENDS

CODE SEGMENT
    ASSUME CS:CODE, DS:DATA
START:
    MOV AH, 4CH
    INT 21H
CODE ENDS

END START
```

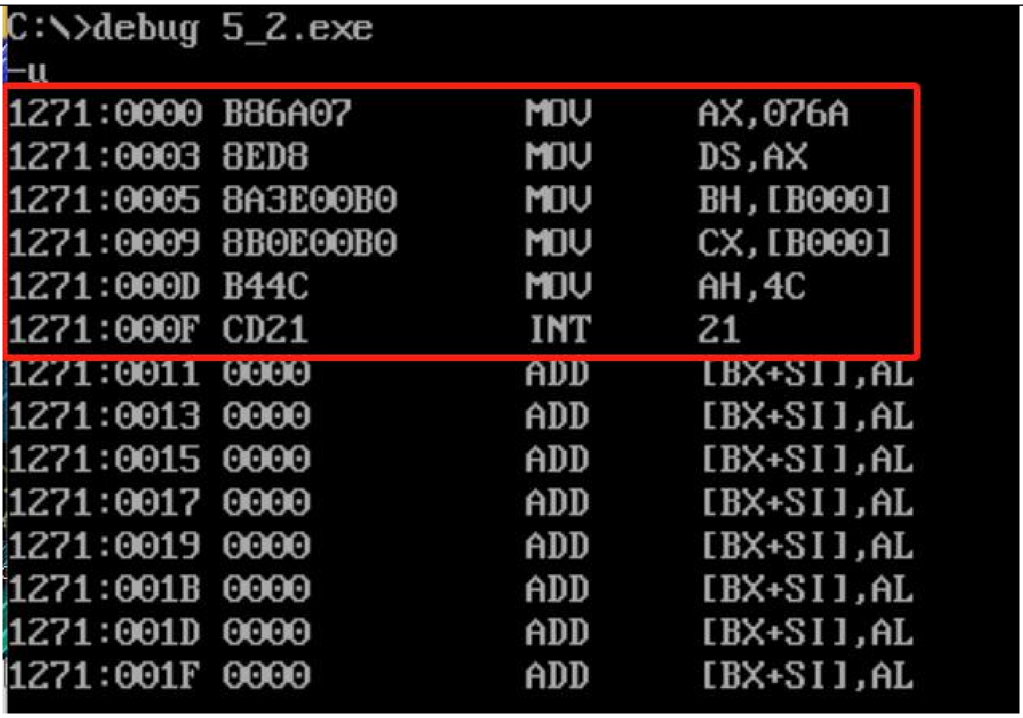

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
076A:0000  41 42 43 44 45 46 47 48-24 00 00 00 00 00 00 00   ABCDEFGH$.......
076A:0010  00 00 00 62 79 74 65 24-77 6F 72 64 24 00 00 00   ...byte$word$...
076A:0020  B4 4C CD 21 04 50 E8 9F-0E 83 C4 04 3D FF FF 74   .L.!.P......=..t
076A:0030  03 E9 11 01 B8 2F 00 50-8B 46 FC 8B 56 FE 05 0C   ...../.P.F..V...
076A:0040  00 52 50 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04   .RP..H...P.{....
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.....
-q

C:\>debug define.exe
-r
AX=FFFF  BX=0000  CX=0024  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076C  IP=0000   NV UP EI PL NZ NA PO NC
076C:0000 B44C          MOV     AH,4C
-d 076a:0
076A:0000  41 42 43 44 45 46 47 48-24 00 00 00 00 00 00 00   ABCDEFGH$.......
076A:0010  00 00 00 62 79 74 65 24-77 6F 72 64 24 00 00 00   ...byte$word$...
076A:0020  B4 4C CD 21 04 50 E8 9F-0E 83 C4 04 3D FF FF 74   .L.!.P......=..t
076A:0030  03 E9 11 01 B8 2F 00 50-8B 46 FC 8B 56 FE 05 0C   ...../.P.F..V...
076A:0040  00 52 50 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04   .RP..H...P.{....
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.....
-
```

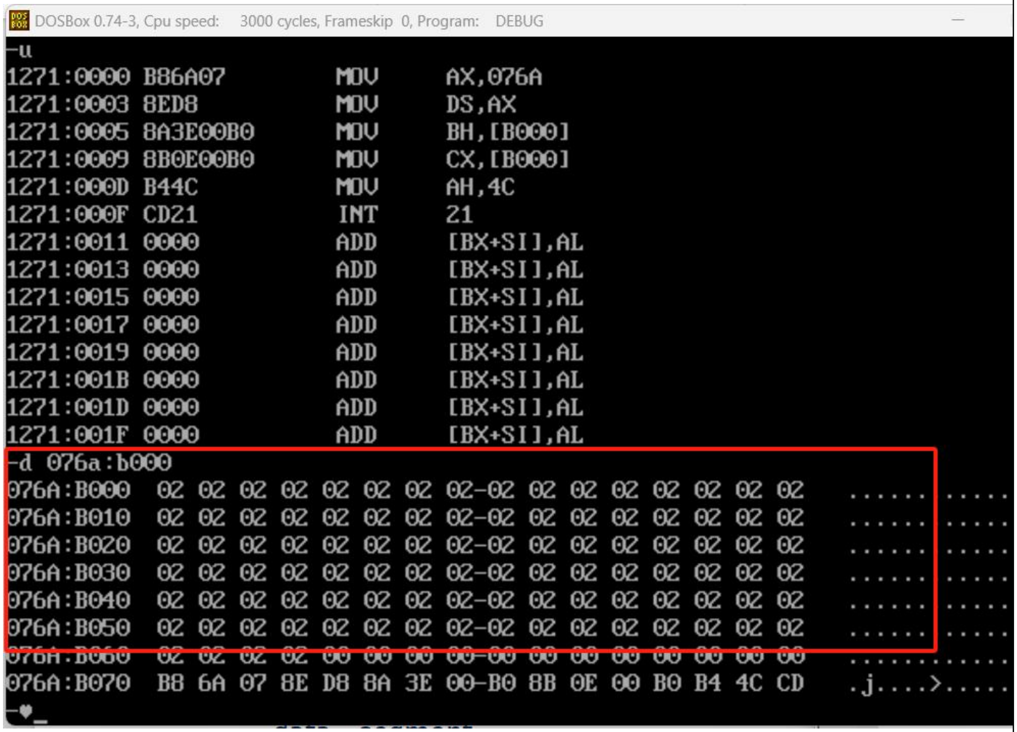
8.2 设置一个位置从 0B000H 开始，名为 DATA 的数据段，段中定义一个具有 100 字节的数组，其类型属性即是字又是字节。



```
DATA SEGMENT
    ORG 0B000H

    DATA_ARRAY DB 100 DUP(2)

DATA ENDS

CODE SEGMENT
    ASSUME CS:CODE, DS:DATA
START:
    MOV AX,076AH
    MOV DS,AX
    MOV BH,[DATA_ARRAY]
    MOV CX,[DATA_ARRAY]
    MOV AH, 4CH
    INT 21H
CODE ENDS

END START

```



```
C:\>debug 5_2.exe
-u
1271:0000 B86A07        MOV     AX,076A
1271:0003 8ED8          MOV     DS,AX
1271:0005 8A3E00B0      MOV     BH,[B000]
1271:0009 8B0E00B0      MOV     CX,[B000]
1271:000D B44C          MOV     AH,4C
1271:000F CD21          INT     21
1271:0011 0000          ADD     [BX+SI],AL
1271:0013 0000          ADD     [BX+SI],AL
1271:0015 0000          ADD     [BX+SI],AL
1271:0017 0000          ADD     [BX+SI],AL
1271:0019 0000          ADD     [BX+SI],AL
1271:001B 0000          ADD     [BX+SI],AL
1271:001D 0000          ADD     [BX+SI],AL
1271:001F 0000          ADD     [BX+SI],AL
```



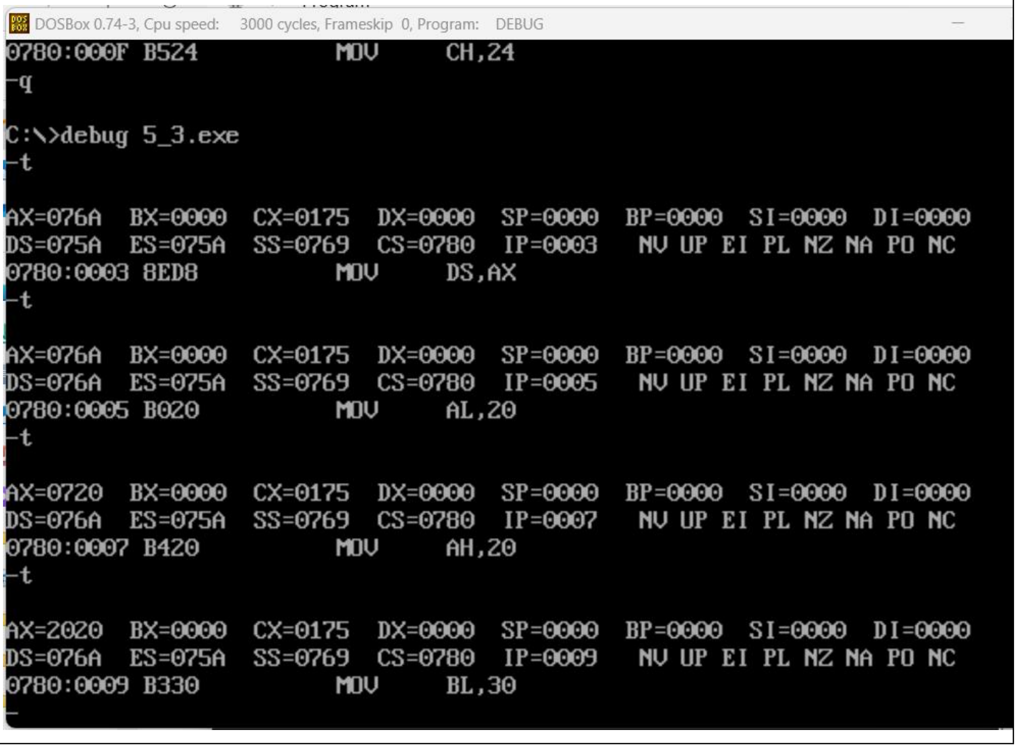
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
-u
1271:0000 B86A07        MOV     AX,076A
1271:0003 8ED8          MOV     DS,AX
1271:0005 8A3E00B0      MOV     BH,[B000]
1271:0009 8B0E00B0      MOV     CX,[B000]
1271:000D B44C          MOV     AH,4C
1271:000F CD21          INT     21
1271:0011 0000          ADD     [BX+SI],AL
1271:0013 0000          ADD     [BX+SI],AL
1271:0015 0000          ADD     [BX+SI],AL
1271:0017 0000          ADD     [BX+SI],AL
1271:0019 0000          ADD     [BX+SI],AL
1271:001B 0000          ADD     [BX+SI],AL
1271:001D 0000          ADD     [BX+SI],AL
1271:001F 0000          ADD     [BX+SI],AL
-d 076a:b000
076A:B000  02 02 02 02 02 02 02 02-02 02 02 02 02 02 02 02   ................
076A:B010  02 02 02 02 02 02 02 02-02 02 02 02 02 02 02 02   ................
076A:B020  02 02 02 02 02 02 02 02-02 02 02 02 02 02 02 02   ................
076A:B030  02 02 02 02 02 02 02 02-02 02 02 02 02 02 02 02   ................
076A:B040  02 02 02 02 02 02 02 02-02 02 02 02 02 02 02 02   ................
076A:B050  02 02 02 02 02 02 02 02-02 02 02 02 02 02 02 02   ................
076A:B060  02 02 02 02 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:B070  B8 6A 07 8E D8 8A 3E 00-B0 8B 0E 00 B0 B4 4C CD   .j....>.....
-
```

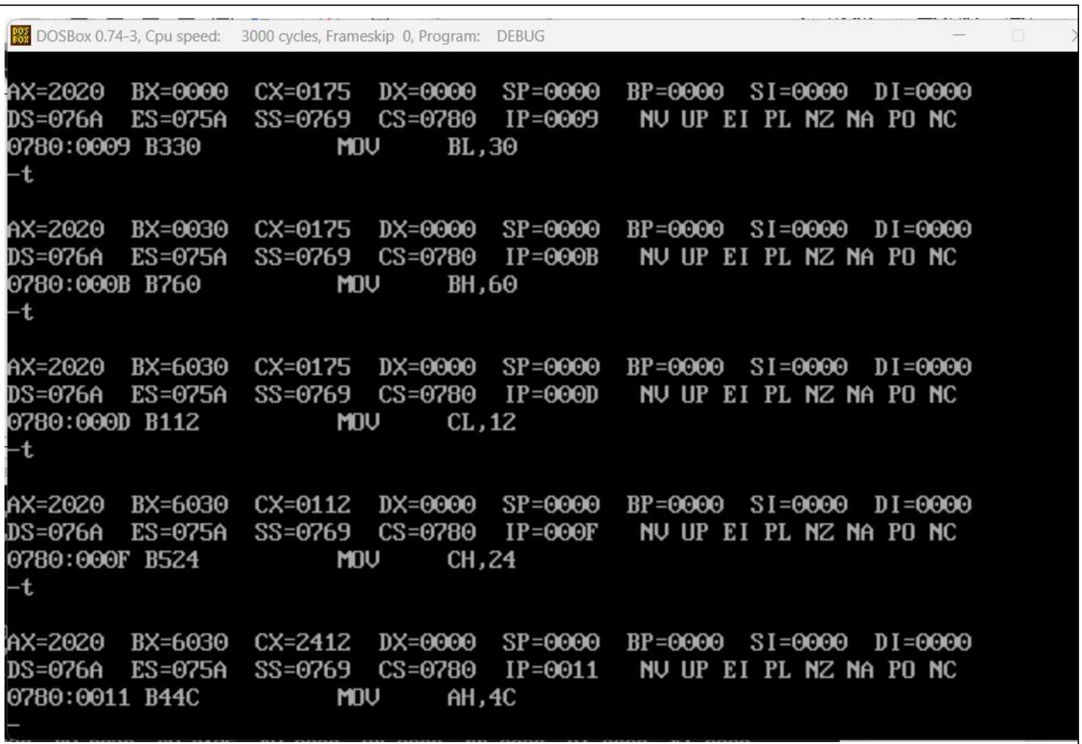
8.3 下述指令序列执行后，AX, BX, CX 寄存器的内容分别是多少？

```
data  segment
    org  20h         ; 规定起始位置
    var1  db  20h  dup(0)
```

```
   var2   dw   30h   dup(0)
   var3   dw   12h   dup(4 dup(2), 30h)
data   ends

code   segment
    assume   cs:code,   ds:data
start:     mov   ax, data
        mov   ds, ax
        mov    AL,    LENGTH   VAR1   ;   H
        MOV    AH,   SIZE        VAR1   ; H
        MOV    BL,    LENGTH   VAR2   ;   H
        MOV    BH,   SIZE       VAR2    ;   H
        MOV    CL,    LENGTH   VAR3   ; H
        MOV    CH,    SIZE   VAR3       ; H
        mov    ah, 4ch
        int   21h
 code   ends
    end   start
```


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

```
0780:000F B524          MOV     CH,24
-q

C:\>debug 5_3.exe
-t

AX=076A  BX=0000  CX=0175  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=0780  IP=0003   NV UP EI PL NZ NA PO NC
0780:0003 8ED8          MOV     DS,AX
-t

AX=076A  BX=0000  CX=0175  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=0780  IP=0005   NV UP EI PL NZ NA PO NC
0780:0005 B020          MOV     AL,20
-t

AX=0720  BX=0000  CX=0175  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=0780  IP=0007   NV UP EI PL NZ NA PO NC
0780:0007 B420          MOV     AH,20
-t

AX=2020  BX=0000  CX=0175  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=0780  IP=0009   NV UP EI PL NZ NA PO NC
0780:0009 B330          MOV     BL,30
-
```


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

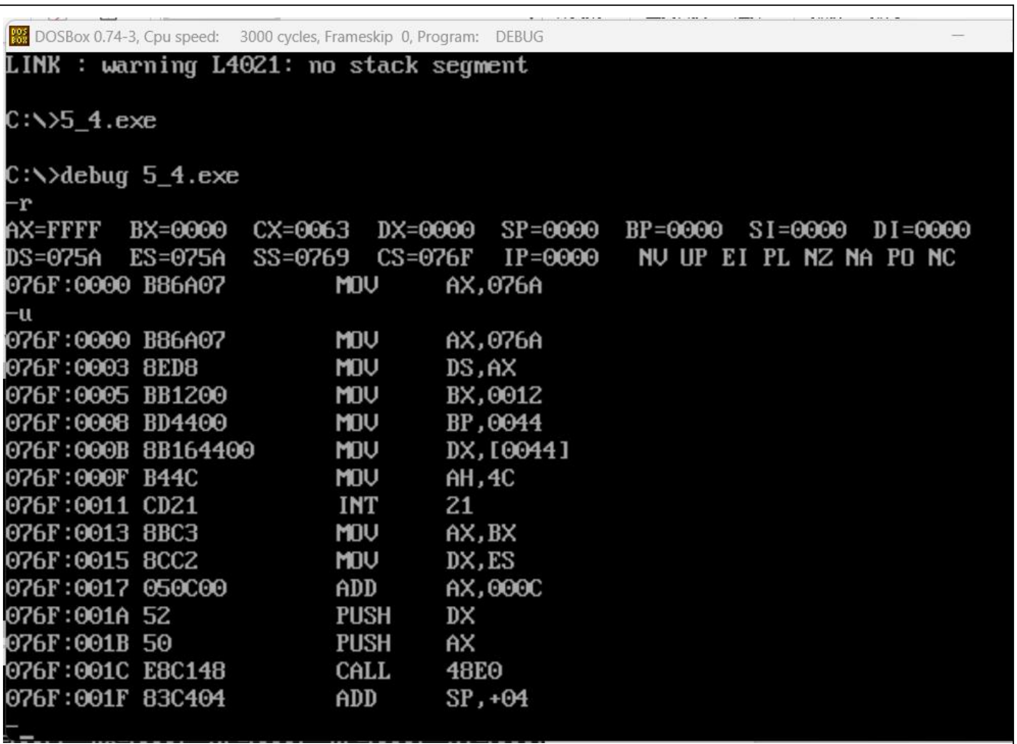
```
AX=2020  BX=0000  CX=0175  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=0780  IP=0009   NV UP EI PL NZ NA PO NC
0780:0009 B330          MOV     BL,30
-t

AX=2020  BX=0030  CX=0175  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=0780  IP=000B   NV UP EI PL NZ NA PO NC
0780:000B B760          MOV     BH,60
-t

AX=2020  BX=6030  CX=0175  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=0780  IP=000D   NV UP EI PL NZ NA PO NC
0780:000D B112          MOV     CL,12
-t

AX=2020  BX=6030  CX=0112  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=0780  IP=000F   NV UP EI PL NZ NA PO NC
0780:000F B524          MOV     CH,24
-t

AX=2020  BX=6030  CX=2412  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=076A  ES=075A  SS=0769  CS=0780  IP=0011   NV UP EI PL NZ NA PO NC
0780:0011 B44C          MOV     AH,4C
-
```

AX:2020H BX:6030H CX:2412H

8.4 根据下面的程序回答问题。

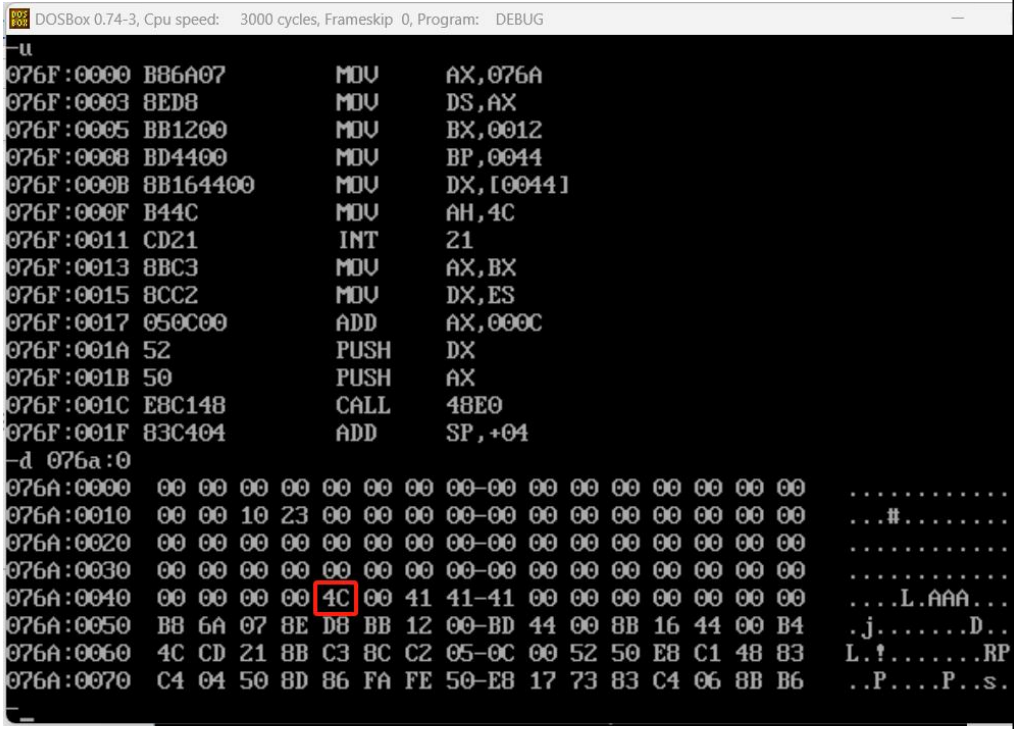
```
DATA   SEGMENT
 org 12h
 db1   db 10h, 23h
 org $+30h
 var1   dw   $+8
 x   db 'AAA'
DATA   ENDS

Code   segment
     assume     cs:code,ds:data
Start:   mov     ax,   data
         mov     ds, ax
         mov   bx, offset   db1
         mov   bp, offset   var1
         mov   dx, var1
         mov   ah, 4ch
         int     21h
CODE       ENDS
 END    START
```

DB1 :偏移量 12H,   VAR1:偏移量 44H, 其内容为 4CH



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

```
LINK : warning L4021: no stack segment

C:\>5_4.exe

C:\>debug 5_4.exe
-r
AX=FFFF  BX=0000  CX=0063  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076F  IP=0000   NV UP EI PL NZ NA PO NC
076F:0000 B86A07        MOV     AX,076A
-u
076F:0000 B86A07        MOV     AX,076A
076F:0003 8ED8          MOV     DS,AX
076F:0005 BB1200        MOV     BX,0012
076F:0008 BD4400        MOV     BP,0044
076F:000B 8B164400      MOV     DX,[0044]
076F:000F B44C          MOV     AH,4C
076F:0011 CD21          INT     21
076F:0013 8BC3          MOV     AX,BX
076F:0015 8CC2          MOV     DX,ES
076F:0017 050C00        ADD     AX,000C
076F:001A 52            PUSH    DX
076F:001B 50            PUSH    AX
076F:001C E8C148        CALL    48E0
076F:001F 83C404        ADD     SP,+04
-
```



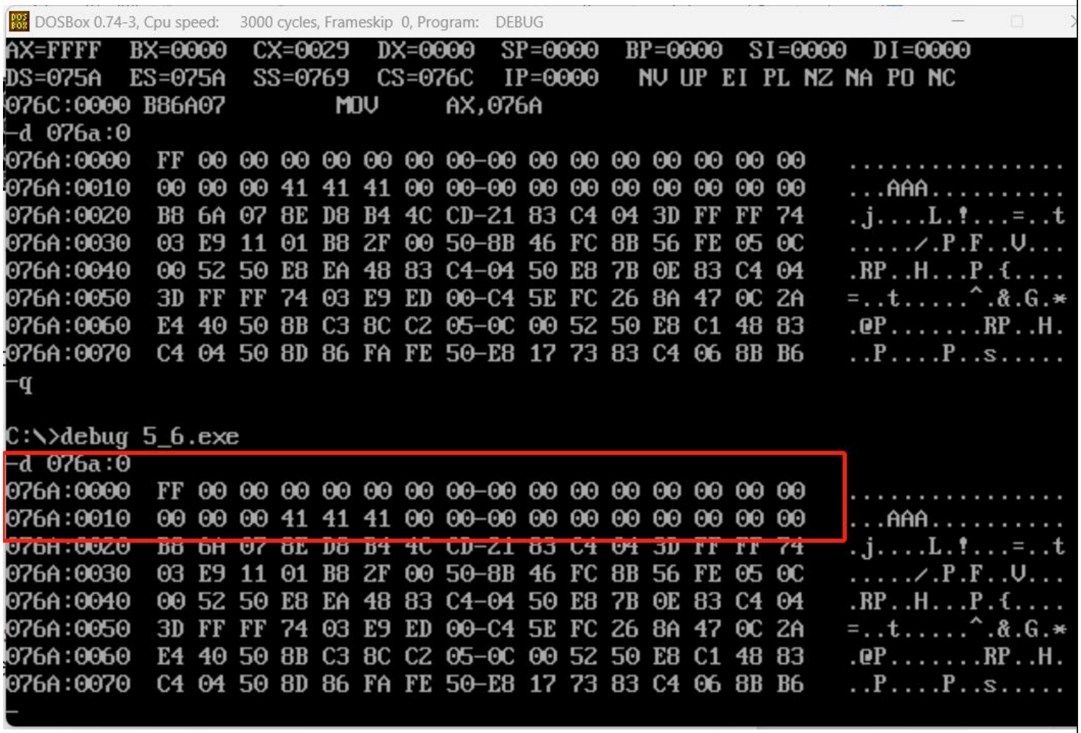
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
-u
076F:0000 B86A07        MOV     AX,076A
076F:0003 8ED8          MOV     DS,AX
076F:0005 BB1200        MOV     BX,0012
076F:0008 BD4400        MOV     BP,0044
076F:000B 8B164400      MOV     DX,[0044]
076F:000F B44C          MOV     AH,4C
076F:0011 CD21          INT     21
076F:0013 8BC3          MOV     AX,BX
076F:0015 8CC2          MOV     DX,ES
076F:0017 050C00        ADD     AX,000C
076F:001A 52            PUSH    DX
076F:001B 50            PUSH    AX
076F:001C E8C148        CALL    48E0
076F:001F 83C404        ADD     SP,+04
-d 076a:0
076A:0000  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0010  00 00 10 23 00 00 00 00-00 00 00 00 00 00 00 00   ...#............
076A:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0040  00 00 00 00 4C 00 41 41-41 00 00 00 00 00 00 00   ....L.AAA.......
076A:0050  B8 6A 07 8E D8 BB 12 00-BD 44 00 8B 16 44 00 B4   .j.......D...D..
076A:0060  4C CD 21 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83   L.!.......RP..H.
076A:0070  C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6   ..P....P..s.....
-
```

8.6 写出以下数据段中各符号对应的值。

```
DATA   SEGMENT
  num1   equ 10h
  num2   equ   num1 mod   10h
  num3   db   (12 or 6 and 2)    le 0eh   ;   FFH
  num4   db   num1 dup(?)                 ;   10H 个 0
```

```
    num5   dw   num3            ;  取 num3 的偏移量，值为 0000H

    x   db 'AAA'
DATA   ENDS

Code   segment
       assume    cs:code,ds:data
Start:   mov     ax,   data
         mov     ds, ax
         mov   ah, 4ch
         int     21h
CODE       ENDS
 END    START
```



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

```
AX=FFFF  BX=0000  CX=0029  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076C  IP=0000   NV UP EI PL NZ NA PO NC
076C:0000 B86A07        MOV     AX,076A
-d 076a:0
076A:0000  FF 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0010  00 00 00 41 41 41 00 00-00 00 00 00 00 00 00 00   ...AAA..........
076A:0020  B8 6A 07 8E D8 B4 4C CD-21 83 C4 04 3D FF FF 74   .j....L.!...=..t
076A:0030  03 E9 11 01 B8 2F 00 50-8B 46 FC 8B 56 FE 05 0C   ...../.P.F..V...
076A:0040  00 52 50 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04   .RP..H...P.{....
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.....
-q

C:\>debug 5_6.exe
-d 076a:0
076A:0000  FF 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076A:0010  00 00 00 41 41 41 00 00-00 00 00 00 00 00 00 00   ...AAA..........
076A:0020  B8 6A 07 8E D8 B4 4C CD-21 83 C4 04 3D FF FF 74   .j....L.!...=..t
076A:0030  03 E9 11 01 B8 2F 00 50-8B 46 FC 8B 56 FE 05 0C   ...../.P.F..V...
076A:0040  00 52 50 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04   .RP..H...P.{....
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.....
-
```

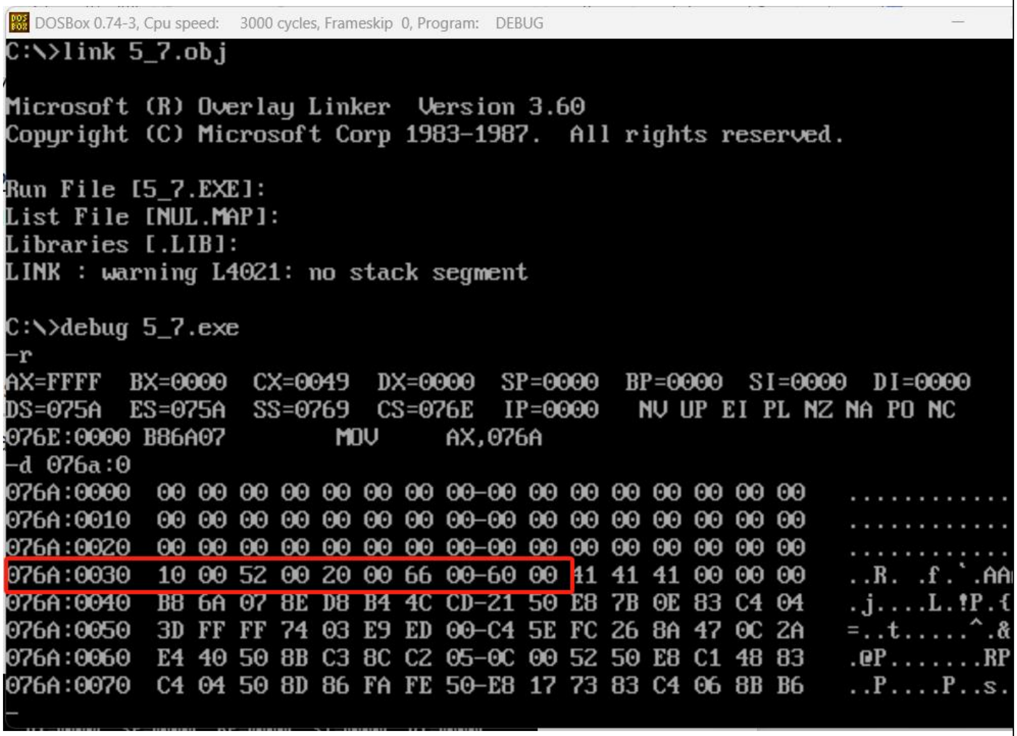
8.7 下面语句汇编后，两处$值各为多少？为使 DA2 字存储单元中数据为 60H，空格处应为何值？

32H

36H

```
    DATA  SEGMENT
      org 30h
      num=20h
      da1   dw 10h, $+20h, 20h, $+30h
      da2   dw   da1+num+10h
      x   db 'AAA'
    DATA  ENDS
```

```
Code   segment
     assume    cs:code,ds:data
Start:   mov     ax,   data
          mov     ds, ax
          mov   ah, 4ch
          int     21h
CODE       ENDS
 END    START
```

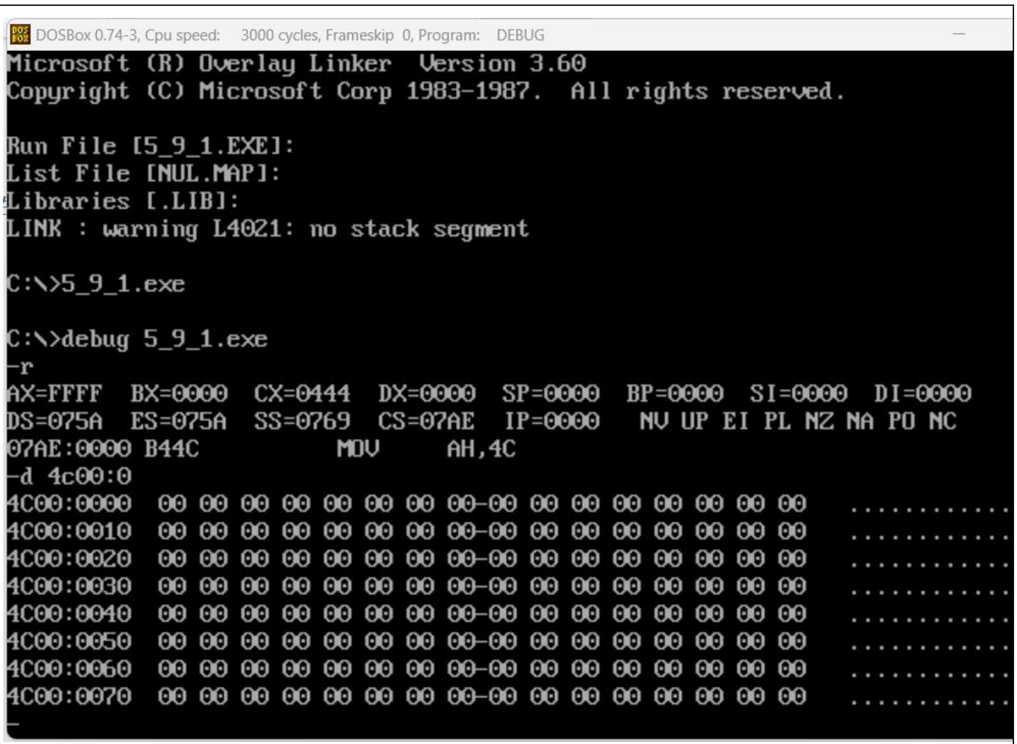

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
C:\>link 5_7.obj

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

Run File [5_7.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>debug 5_7.exe
-r
AX=FFFF  BX=0000  CX=0049  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076E  IP=0000   NV UP EI PL NZ NA PO NC
076E:0000 B86A07        MOV     AX,076A
-d 076a:0
076A:0000  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ............
076A:0010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ............
076A:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ............
076A:0030  10 00 52 00 20 00 66 00-60 00 41 41 41 00 00 00   ..R. .f.`.AA
076A:0040  B8 6A 07 8E D8 B4 4C CD-21 50 E8 7B 0E 83 C4 04   .j....L.!P.{
076A:0050  3D FF FF 74 03 E9 ED 00-C4 5E FC 26 8A 47 0C 2A   =..t.....^.&
076A:0060  E4 40 50 8B C3 8C C2 05-0C 00 52 50 E8 C1 48 83   .@P.......RP
076A:0070  C4 04 50 8D 86 FA FE 50-E8 17 73 83 C4 06 8B B6   ..P....P..s.
-
```

8.9 按要求写出程序框架:

（1）数据段的位置从 0400H 开始定义一个 50 字节的数组，其类型属性即是字节又可以是字

```
DATA  SEGMENT
   org 0400h
   arword equ this word
   arbyte   db   50 dup(0)
DATA  ENDS
```


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

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

Run File [5_9_1.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>5_9_1.exe

C:\>debug 5_9_1.exe
-r
AX=FFFF  BX=0000  CX=0444  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=07AE  IP=0000   NV UP EI PL NZ NA PO NC
07AE:0000 B44C          MOV     AH,4C
-d 4c00:0
4C00:0000  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ............
4C00:0010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ............
4C00:0020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ............
4C00:0030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ............
4C00:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ............
4C00:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ............
4C00:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ............
4C00:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ............
-
```

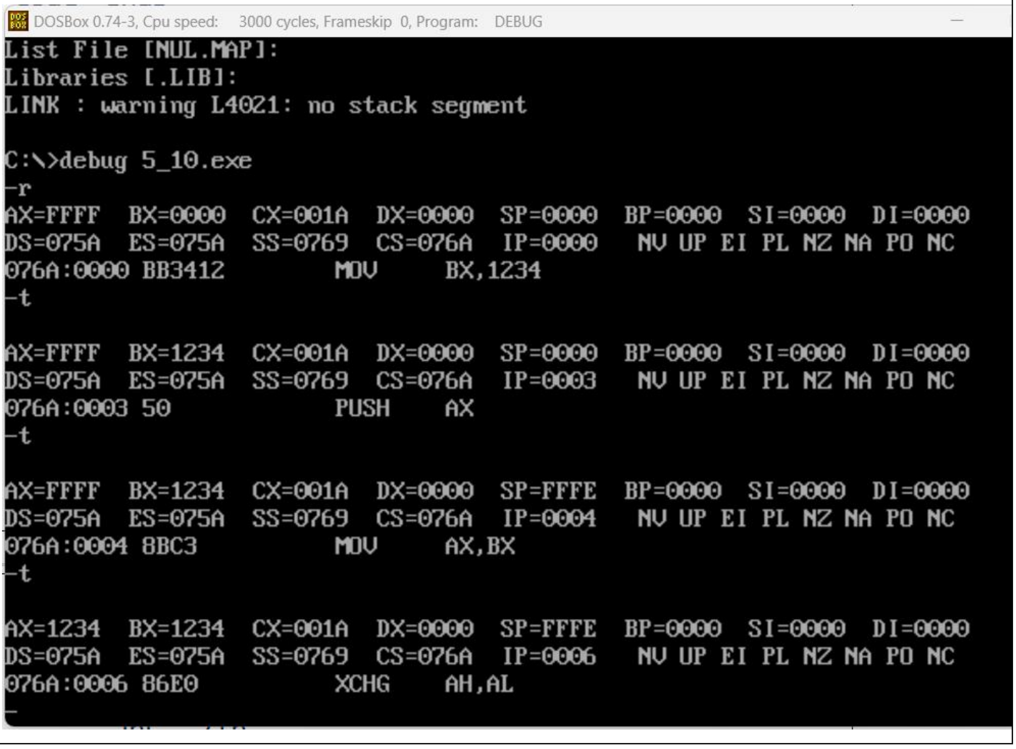
（2）堆栈从节边界开始，指定 50H 字节单元为堆栈，并定义栈顶指针

```
mystack   SEGMENT    para   stack
          db   50 dup(0)
          top   label word   ;  标记栈底位置
mystack   ENDS
CODE      SEGMENT
       assume    cs:code,    ss:mystack
          MOV    AX , mySTACK          ;设置 SS
          MOV    SS , AX
          MOV    SP , OFFSET    TOP   ;设置 SP
code   ends
 end
```

8.10 定义一个宏，实现将某一寄存器内容高 8 位与低 8 位互换

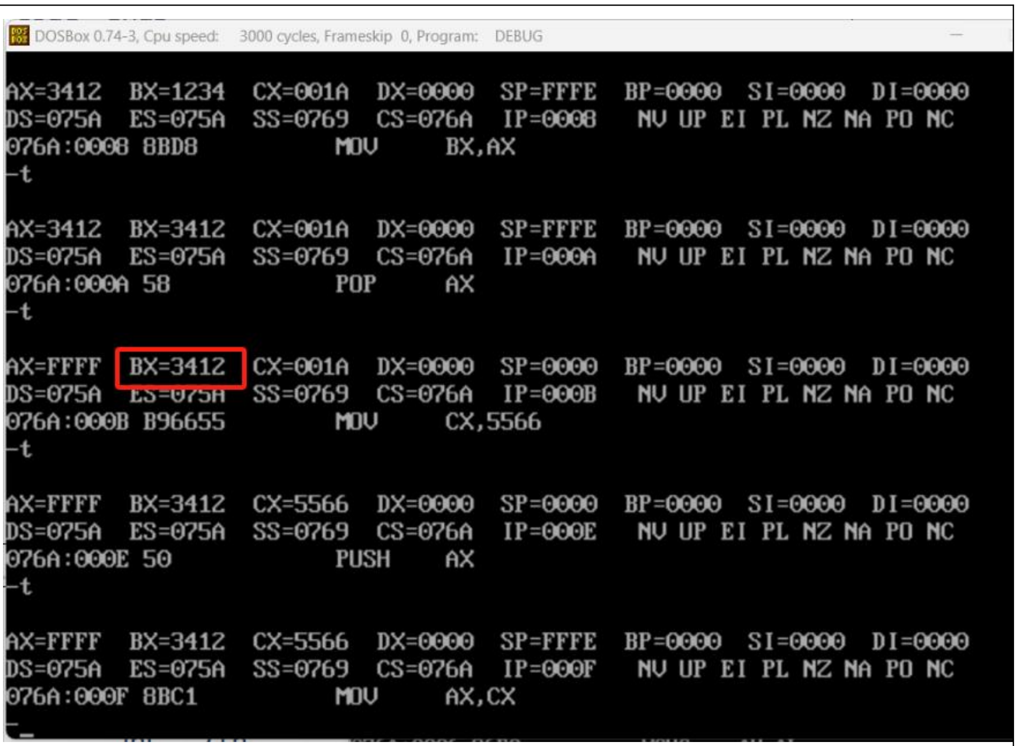
```
CODE      SEGMENT
      assume    cs:code
 exreg    macro   op
      push   ax            ; ax 入栈保存
      mov   ax,    op
```

```
        xchg   ah, al        ; 高 8、低 8 位互换

        mov   op, ax         ; 重新赋值

        pop   ax             ; 恢复 ax

    endm

START:   MOV   BX, 1234H
         EXREG   BX      ; 测试，执行后（bx)=3412h
         MOV     CX, 5566H
         EXREG   CX      ; 测试，执行后（CX)=6655h

         mov   ah, 4ch
          int    21h
code   ends
 end   START
```

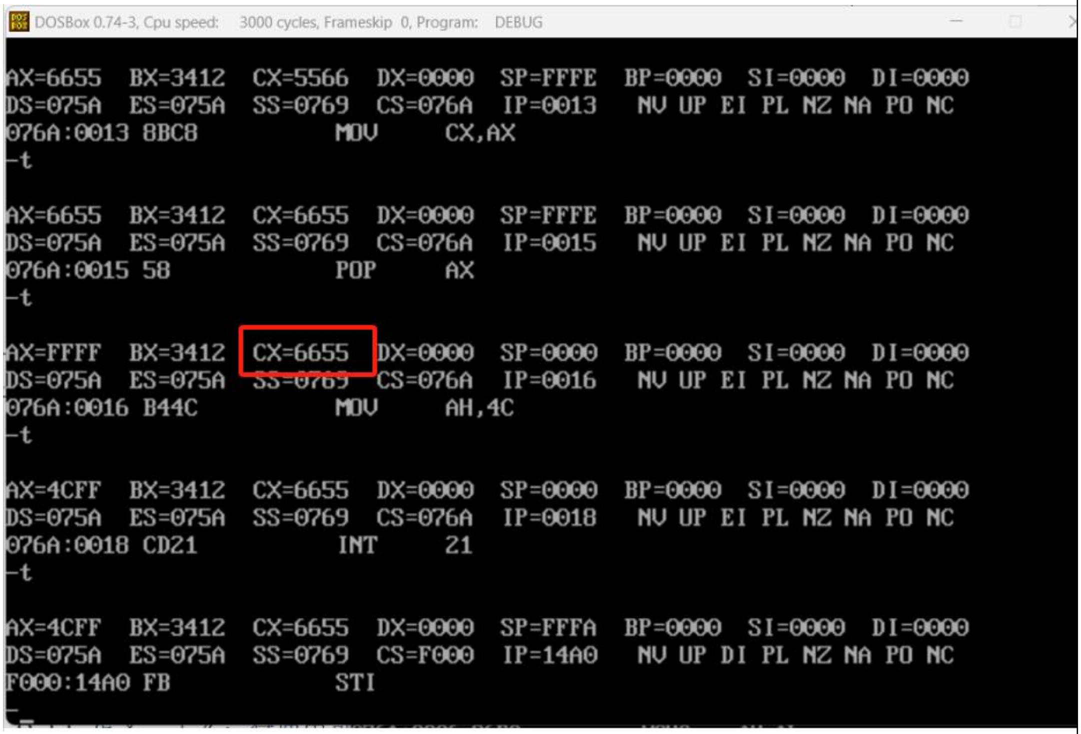


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>debug 5_10.exe
-r
AX=FFFF  BX=0000  CX=001A  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0000   NV UP EI PL NZ NA PO NC
076A:0000 BB3412        MOV     BX,1234
-t

AX=FFFF  BX=1234  CX=001A  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0003   NV UP EI PL NZ NA PO NC
076A:0003 50            PUSH    AX
-t

AX=FFFF  BX=1234  CX=001A  DX=0000  SP=FFFE  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0004   NV UP EI PL NZ NA PO NC
076A:0004 8BC3          MOV     AX,BX
-t

AX=1234  BX=1234  CX=001A  DX=0000  SP=FFFE  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0006   NV UP EI PL NZ NA PO NC
076A:0006 86E0          XCHG    AH,AL
-
```


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

```
AX=3412  BX=1234  CX=001A  DX=0000  SP=FFFE  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0008   NV UP EI PL NZ NA PO NC
076A:0008 8BD8          MOV     BX,AX
-t

AX=3412  BX=3412  CX=001A  DX=0000  SP=FFFE  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=000A   NV UP EI PL NZ NA PO NC
076A:000A 58            POP     AX
-t

AX=FFFF  BX=3412  CX=001A  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=000B   NV UP EI PL NZ NA PO NC
076A:000B B96655        MOV     CX,5566
-t

AX=FFFF  BX=3412  CX=5566  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=000E   NV UP EI PL NZ NA PO NC
076A:000E 50            PUSH    AX
-t

AX=FFFF  BX=3412  CX=5566  DX=0000  SP=FFFE  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=000F   NV UP EI PL NZ NA PO NC
076A:000F 8BC1          MOV     AX,CX
-
```


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

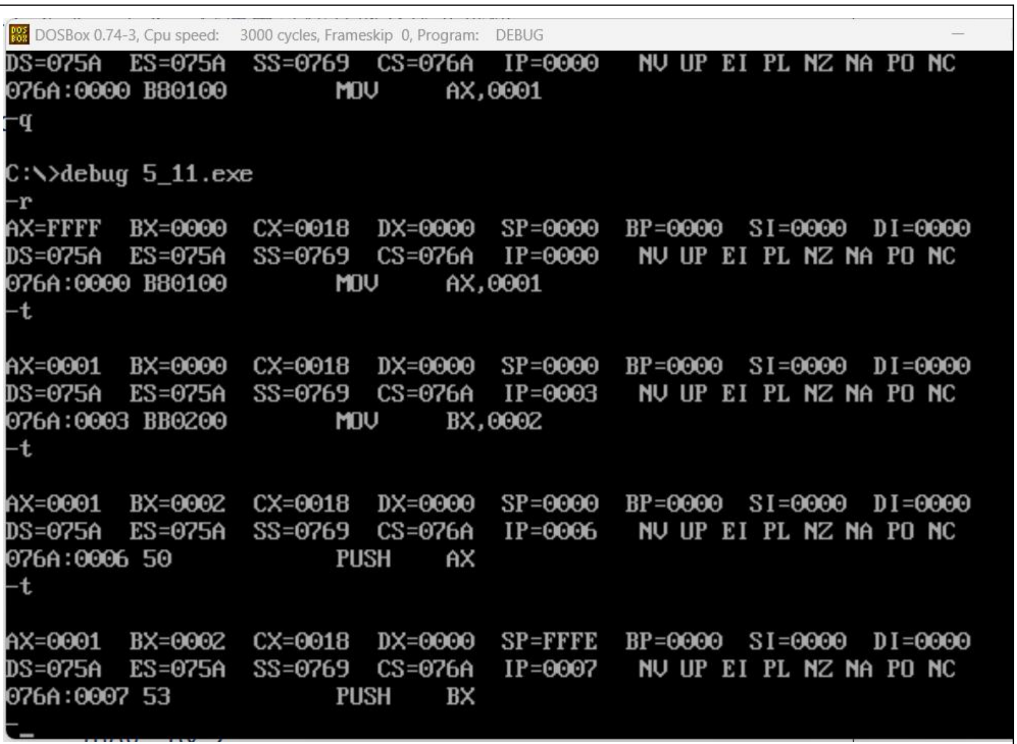
```
AX=6655  BX=3412  CX=5566  DX=0000  SP=FFFE  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0013   NV UP EI PL NZ NA PO NC
076A:0013 8BC8          MOV     CX,AX
-t

AX=6655  BX=3412  CX=6655  DX=0000  SP=FFFE  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0015   NV UP EI PL NZ NA PO NC
076A:0015 58            POP     AX
-t

AX=FFFF  BX=3412  CX=6655  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0016   NV UP EI PL NZ NA PO NC
076A:0016 B44C          MOV     AH,4C
-t

AX=4CFF  BX=3412  CX=6655  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0018   NV UP EI PL NZ NA PO NC
076A:0018 CD21          INT     21
-t

AX=4CFF  BX=3412  CX=6655  DX=0000  SP=FFFA  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=F000  IP=14A0   NV UP DI PL NZ NA PO NC
F000:14A0 FB            STI
-
```

8.11 定义一个宏，将通用寄存器内容压入堆栈。

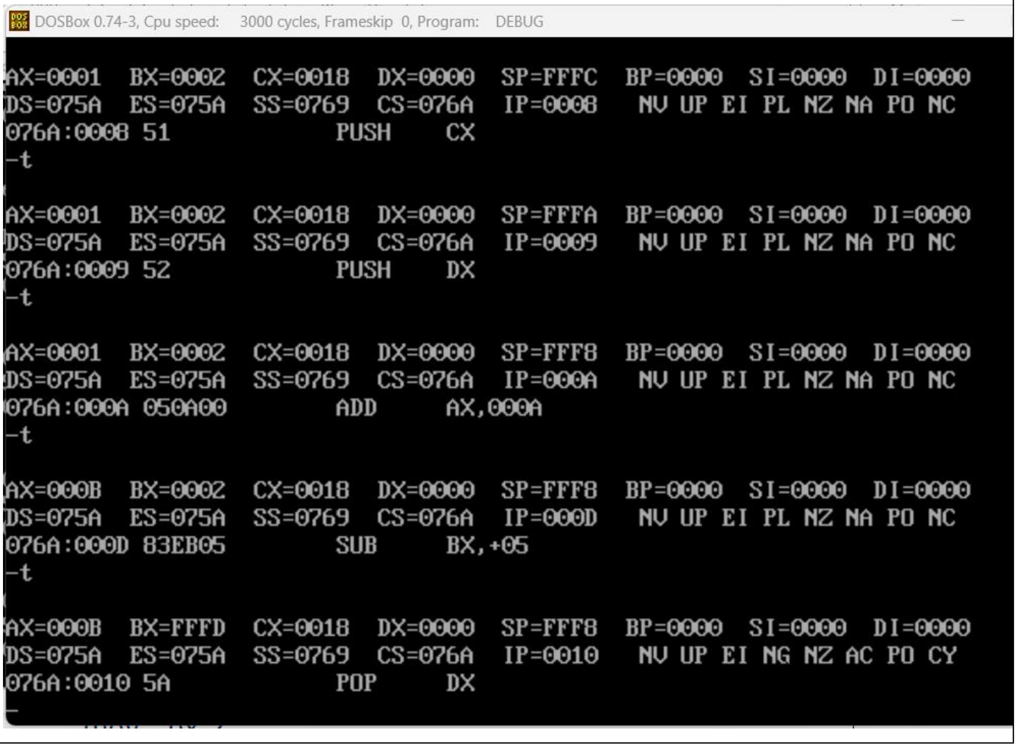
注：通用寄存器指 ax, bx, cx, dx

```
Code   segment
     assume    cs:code
```

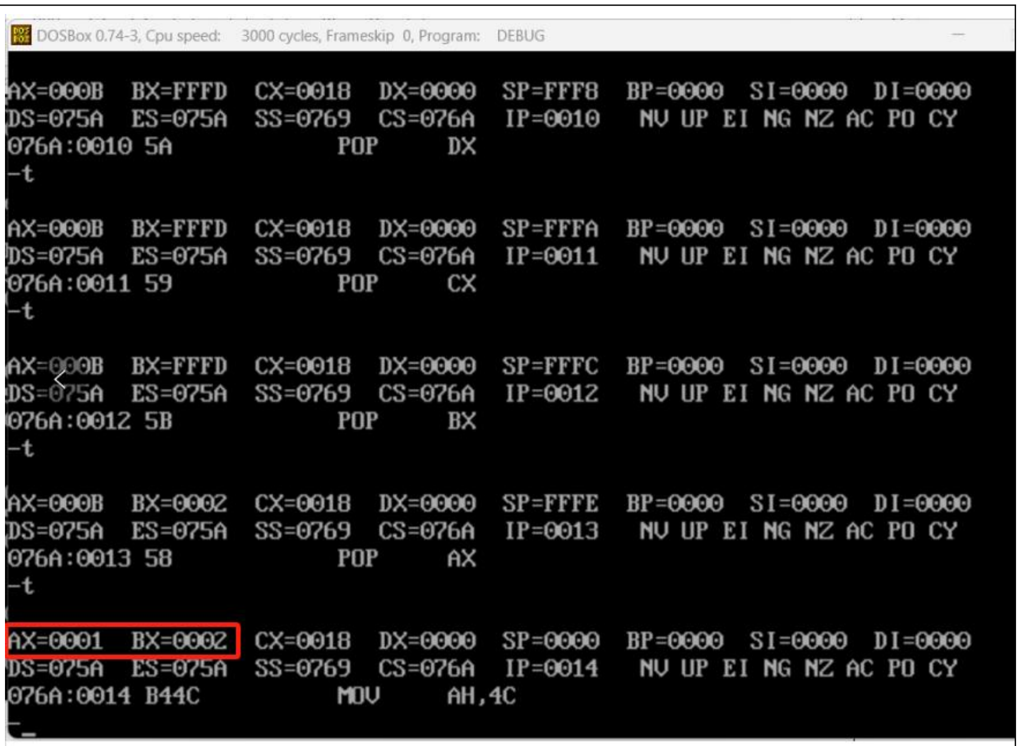
```
    savereg   macro   ； 入栈保存

        push  ax
        push  bx
        push  cx
        push  dx
      endm


    restorereg   macro   ;出栈恢复

        pop  dx
        pop  cx
        pop  bx
        pop  ax
      endm

start:   mov  ax,1
         mov  bx,2
        savereg   ； 保存寄存器值

        add  ax, 10  ； 变动 ax, bx 的值

        sub  bx, 5

        restorereg  ； 恢复寄存器。应该可见(ax)=1,(bx)=2 恢复初值

        mov  ah, 4ch
        int   21h
CODE      ENDS
 END   START
```


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

```
DS=075A  ES=075A  SS=0769  CS=076A  IP=0000   NV UP EI PL NZ NA PO NC
076A:0000 B80100        MOV     AX,0001
-q

C:\>debug 5_11.exe
-r
AX=FFFF  BX=0000  CX=0018  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0000   NV UP EI PL NZ NA PO NC
076A:0000 B80100        MOV     AX,0001
-t

AX=0001  BX=0000  CX=0018  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0003   NV UP EI PL NZ NA PO NC
076A:0003 BB0200        MOV     BX,0002
-t

AX=0001  BX=0002  CX=0018  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0006   NV UP EI PL NZ NA PO NC
076A:0006 50            PUSH    AX
-t

AX=0001  BX=0002  CX=0018  DX=0000  SP=FFFE  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0007   NV UP EI PL NZ NA PO NC
076A:0007 53            PUSH    BX
-
```


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

```

AX=0001  BX=0002  CX=0018  DX=0000  SP=FFFC  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0008   NV UP EI PL NZ NA PO NC
076A:0008 51            PUSH    CX
-t

AX=0001  BX=0002  CX=0018  DX=0000  SP=FFFA  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0009   NV UP EI PL NZ NA PO NC
076A:0009 52            PUSH    DX
-t

AX=0001  BX=0002  CX=0018  DX=0000  SP=FFF8  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=000A   NV UP EI PL NZ NA PO NC
076A:000A 050A00        ADD     AX,000A
-t

AX=000B  BX=0002  CX=0018  DX=0000  SP=FFF8  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=000D   NV UP EI PL NZ NA PO NC
076A:000D 83EB05        SUB     BX,+05
-t

AX=000B  BX=FFFD  CX=0018  DX=0000  SP=FFF8  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0010   NV UP EI NG NZ AC PO CY
076A:0010 5A            POP     DX
-
```

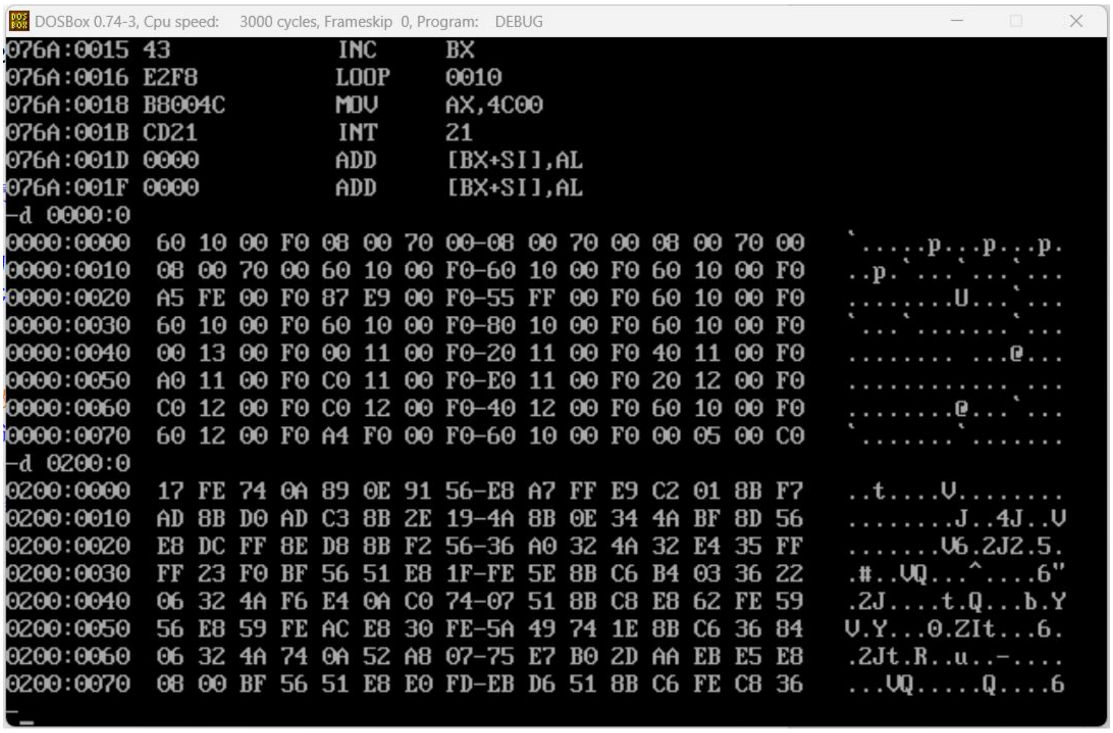

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
AX=000B  BX=FFFD  CX=0018  DX=0000  SP=FFF8  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0010   NV UP EI NG NZ AC PO CY
076A:0010 5A            POP     DX
-t

AX=000B  BX=FFFD  CX=0018  DX=0000  SP=FFFA  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0011   NV UP EI NG NZ AC PO CY
076A:0011 59            POP     CX
-t

AX=000B  BX=FFFD  CX=0018  DX=0000  SP=FFFC  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0012   NV UP EI NG NZ AC PO CY
076A:0012 5B            POP     BX
-t

AX=000B  BX=0002  CX=0018  DX=0000  SP=FFFE  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0013   NV UP EI NG NZ AC PO CY
076A:0013 58            POP     AX
-t

AX=0001  BX=0002  CX=0018  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0014   NV UP EI NG NZ AC PO CY
076A:0014 B44C          MOV     AH,4C
-
```

补充练习

1. 将内存 **0000:0000H** 开始的 128 个字节复制到 **0200:0000H** 处开始的内存中。

```
Code   segment
   Assume   cs:code
 Mov    ax,0
 Mov    ds,ax            ;设置 ds 段寄存器
 Mov    ax, 0200h
 Mov    es, ax          ;  设置 es 附加段寄存器
 Mov    bx,0             ;初始的偏移地址
 Mov    cx,128           ;循环次数
S: mov    al, [bx]
   Mov    es:[bx], al     ;  使用 es  段前缀
   Inc    bx              ;  偏移地址递增 1
   Loop    s              ;  循环
   ;执行完循环后，在 debug 中观察上述两个数据段内容是否一致
   Mov ax,    4c00h
   Int    21h
 Code    ends

end
```



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

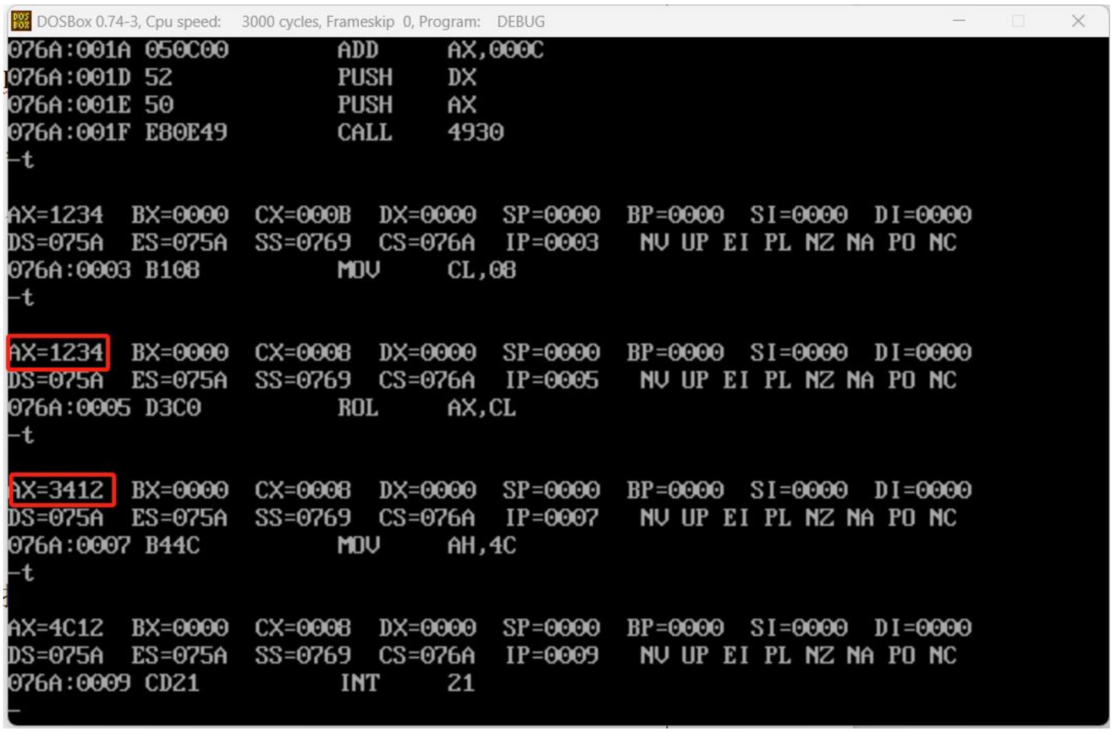
```
076A:0015 43            INC     BX
076A:0016 E2F8          LOOP    0010
076A:0018 B8004C        MOV     AX,4C00
076A:001B CD21          INT     21
076A:001D 0000          ADD     [BX+SI],AL
076A:001F 0000          ADD     [BX+SI],AL
-d 0000:0
0000:0000  60 10 00 F0 08 00 70 00-08 00 70 00 08 00 70 00   `.....p...p...p.
0000:0010  08 00 70 00 60 10 00 F0-60 10 00 F0 60 10 00 F0   ..p.`...`...`...
0000:0020  A5 FE 00 F0 87 E9 00 F0-55 FF 00 F0 60 10 00 F0   ........U...`...
0000:0030  60 10 00 F0 60 10 00 F0-80 10 00 F0 60 10 00 F0   `...`.......`...
0000:0040  00 13 00 F0 00 11 00 F0-20 11 00 F0 40 11 00 F0   ........ ...@...
0000:0050  A0 11 00 F0 C0 11 00 F0-E0 11 00 F0 20 12 00 F0   ............ ...
0000:0060  C0 12 00 F0 C0 12 00 F0-40 12 00 F0 60 10 00 F0   ........@...`...
0000:0070  60 12 00 F0 A4 F0 00 F0-60 10 00 F0 00 05 00 C0   `.......`.......
-d 0200:0
0200:0000  17 FE 74 0A 89 0E 91 56-E8 A7 FF E9 C2 01 8B F7   ..t....V........
0200:0010  AD 8B D0 AD C3 8B 2E 19-4A 8B 0E 34 4A BF 8D 56   ........J..4J..V
0200:0020  E8 DC FF 8E D8 8B F2 56-36 A0 32 4A 32 E4 35 FF   .......V6.2J2.5.
0200:0030  FF 23 F0 BF 56 51 E8 1F-FE 5E 8B C6 B4 03 36 22   .#..VQ...^....6"
0200:0040  06 32 4A F6 E4 0A C0 74-07 51 8B C8 E8 62 FE 59   .2J....t.Q...b.Y
0200:0050  56 E8 59 FE AC E8 30 FE-5A 49 74 1E 8B C6 36 84   V.Y...0.ZIt...6.
0200:0060  06 32 4A 74 0A 52 A8 07-75 E7 B0 2D AA EB E5 E8   .2Jt.R..u..-....
0200:0070  08 00 BF 56 51 E8 E0 FD-EB D6 51 8B C6 FE C8 36   ...VQ.....Q....6
-_
```

2.用循环移位指令或 xchg 指令将 AX 的高 8 位和低 8 位交换。例如(AX)=1234H，交换后为 (AX)=3412H

注意：必须用移位指令，不能用 MOV

做法 1:使用循环移位指令 ROL (即高位被移掉的位又循环补充到低位上)

```
Code   segment
       Assume   cs:code
       Mov    AX, 1234h
       Mov    CL, 8
       ROL      ax, CL          ;循环移位 8 次即可。掌握 ROL 指令
       MOV    ah, 4ch
       Int    21h
Code   ends
   End
```

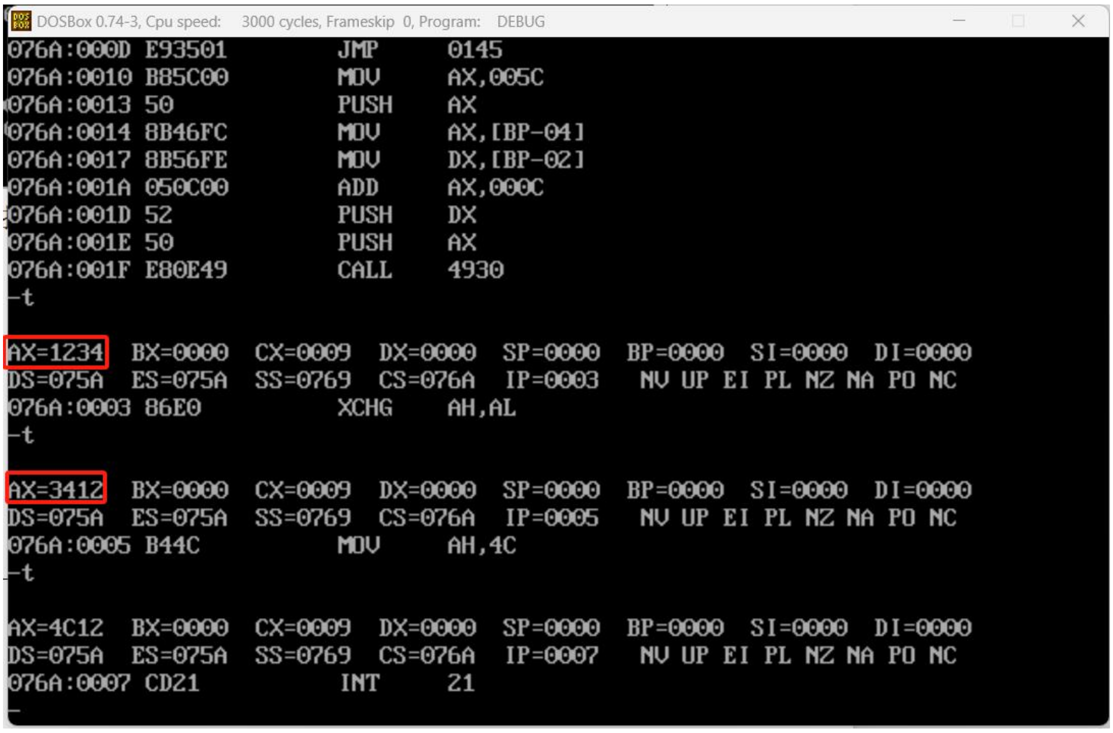


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG

```
076A:001A 050C00        ADD     AX,000C
076A:001D 52            PUSH    DX
076A:001E 50            PUSH    AX
076A:001F E80E49        CALL    4930
-t

AX=1234  BX=0000  CX=000B  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0003   NV UP EI PL NZ NA PO NC
076A:0003 B108          MOV     CL,08
-t

AX=1234  BX=0000  CX=0008  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0005   NV UP EI PL NZ NA PO NC
076A:0005 D3C0          ROL     AX,CL
-t

AX=3412  BX=0000  CX=0008  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0007   NV UP EI PL NZ NA PO NC
076A:0007 B44C          MOV     AH,4C
-t

AX=4C12  BX=0000  CX=0008  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0009   NV UP EI PL NZ NA PO NC
076A:0009 CD21          INT     21
-
```

做法 2 ： 使用 xchg 交換指令

```
Code   segment
       Assume   cs:code
       Mov    AX, 1234h
       Xchg    ah, al
       MOV    ah, 4ch
       Int    21h
Code   ends
   end
```



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

```
076A:000D E93501        JMP     0145
076A:0010 B85C00        MOV     AX,005C
076A:0013 50            PUSH    AX
076A:0014 8B46FC        MOV     AX,[BP-04]
076A:0017 8B56FE        MOV     DX,[BP-02]
076A:001A 050C00        ADD     AX,000C
076A:001D 52            PUSH    DX
076A:001E 50            PUSH    AX
076A:001F E80E49        CALL    4930
-t

AX=1234  BX=0000  CX=0009  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0003   NV UP EI PL NZ NA PO NC
076A:0003 86E0          XCHG    AH,AL
-t

AX=3412  BX=0000  CX=0009  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0005   NV UP EI PL NZ NA PO NC
076A:0005 B44C          MOV     AH,4C
-t

AX=4C12  BX=0000  CX=0009  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=076A  IP=0007   NV UP EI PL NZ NA PO NC
076A:0007 CD21          INT     21
-
```