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 6-第六次上机实验			实验次序	6
实验地点	笃行南楼 A104	参与人员	杨跃浙	实验日期	5.15

一、实验目的

1、汇编语言编程

二、实验环境

Windows 2011，DOSBox

三、实验内容

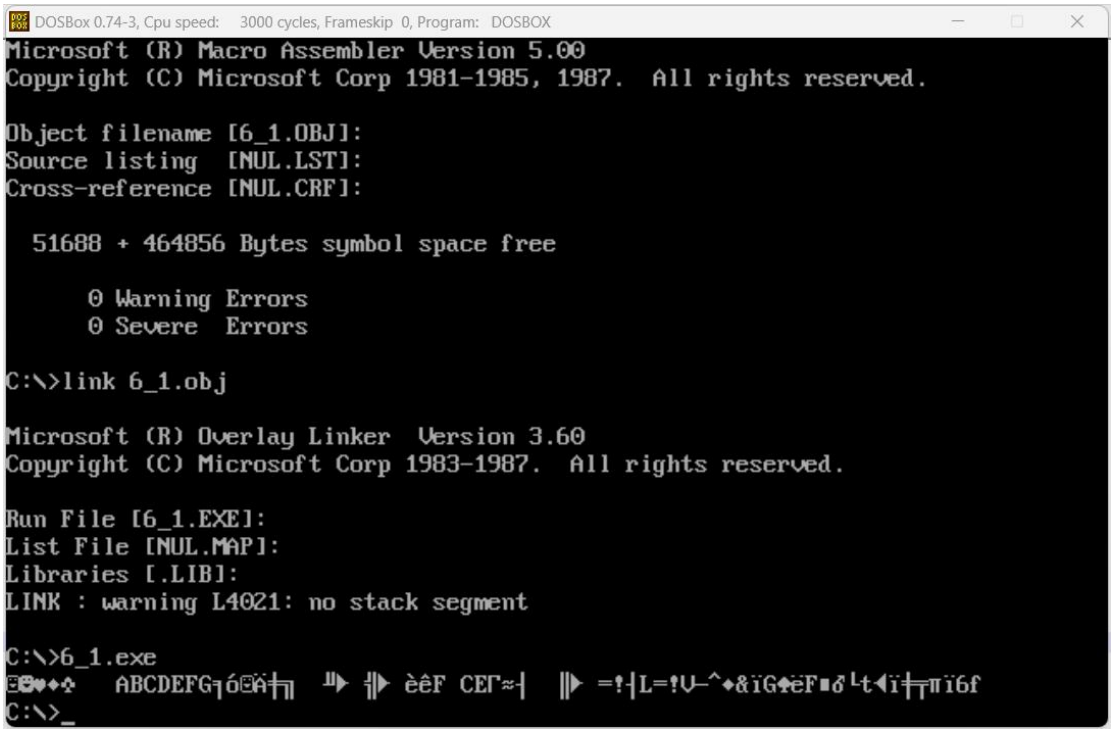
1. 将数组 **x 复制到数组 Y** 中。

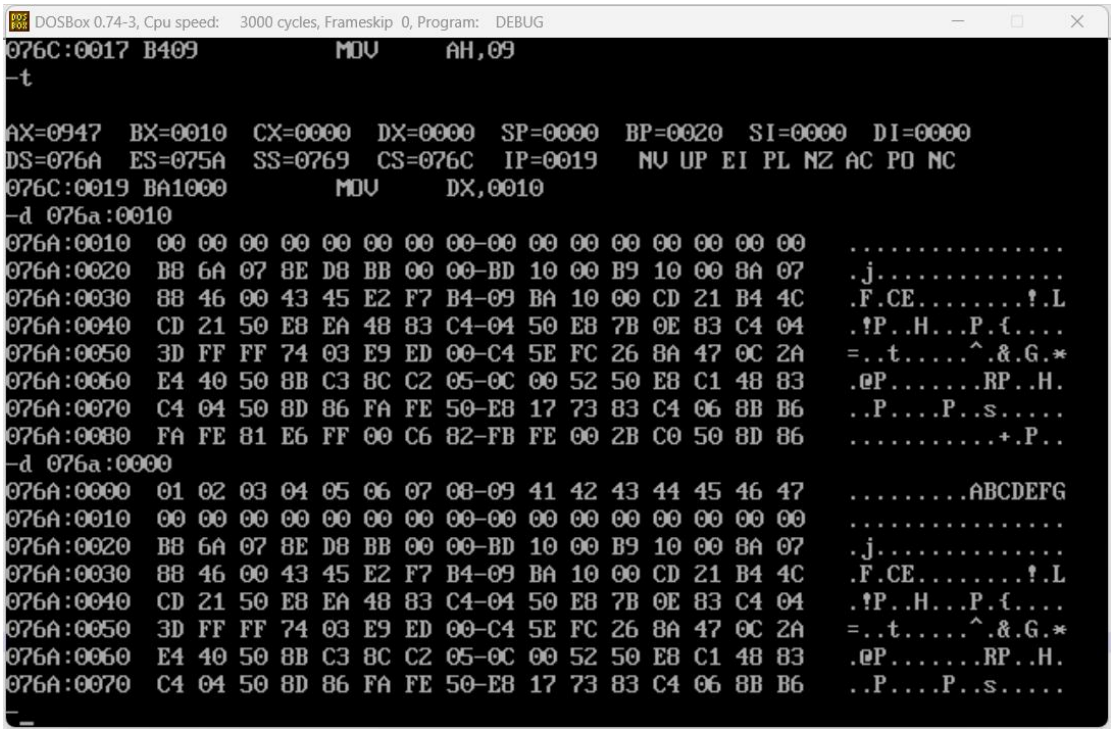
本题练习循环语句 loop。

```
data segment
  x   db   1,2,3,4,5,6,7,8,9,'ABCDEFG'
  Y   db   16 dup (0)
data ends
……
```

5_6.asm | 6_1.asm | 新文件 1

```
1  DATA SEGMENT
2  x db 1,2,3,4,5,6,7,8,9,'ABCDEFG'
3  Y db 16 dup (0)
4  DATA ENDS
5
6  CODE SEGMENT
7  assume cs:CODE, ds:DATA
8  Start:
9      mov ax, DATA
10     mov ds, ax
11
12     mov bx, offset x
13     mov bp, offset Y
14     mov cx, 10h
15
16 copy:
17     mov al, [bx]
18     mov [bp], al
19     inc bx
20     inc bp
21     loop copy
22     mov ah, 09h
23     mov dx, offset Y
24     int 21h
25     mov ah, 4ch
26     int 21h
27 CODE ENDS
28
29 END Start
30
```





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

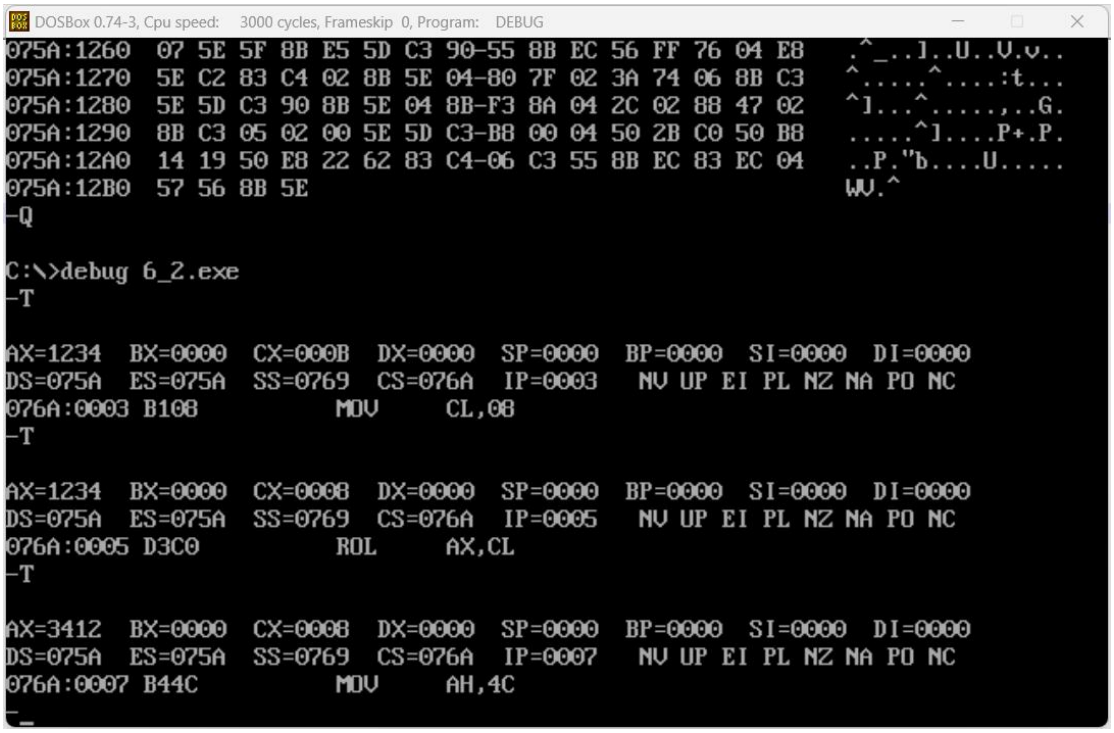
   注意：必须用移位指令，不能用 MOV

   1100 1001 0011

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)

5_6.asm | 6_1.asm | 6_2.asm

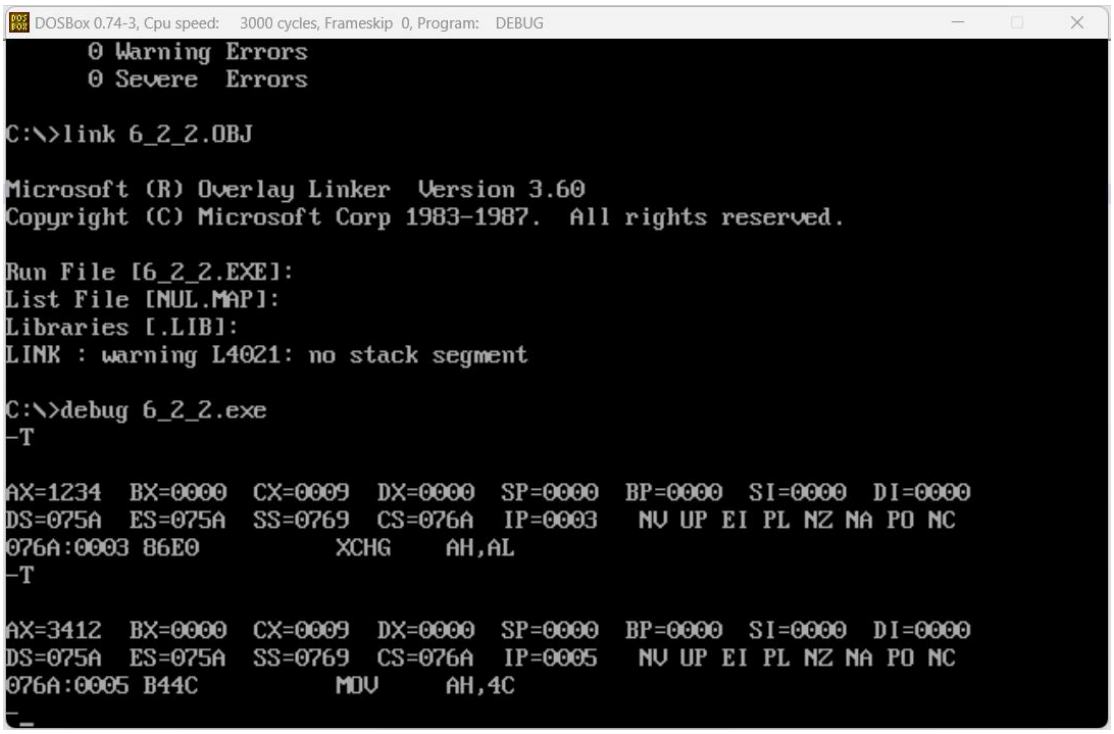
```
1  Code  segment
2  Assume  cs:code
3   Mov  AX, 1234h
4   Mov  CL, 8
5   ROL   ax, CL
6   MOV  ah, 4ch
7   Int  21h
8  Code  ends
9   End
```



做法 2 ： 使用 xchg 交换指令

5_6.asm | 6_1.asm | 6_2.asm | 6_2_2.asm

```
1  Code  segment
2  Assume  cs:code
3   Mov  AX, 1234h
4   Xchg  ah, al
5  MOV  ah, 4ch
6   Int  21h
7  Code  ends
8    end
```



3. 字符串 STR 中保存着 100 个字节的 ASCII 码，试编写一个程序统计该字符串中空格的个数(空格的 ASCII 码为 20H)。将统计结果保存在 count 变量中。

```
count   db   0
str   db      20h,20h, 'abc def',20h, 90 dup (0)
```

本例练习 cmp， jnz, loop 等语句

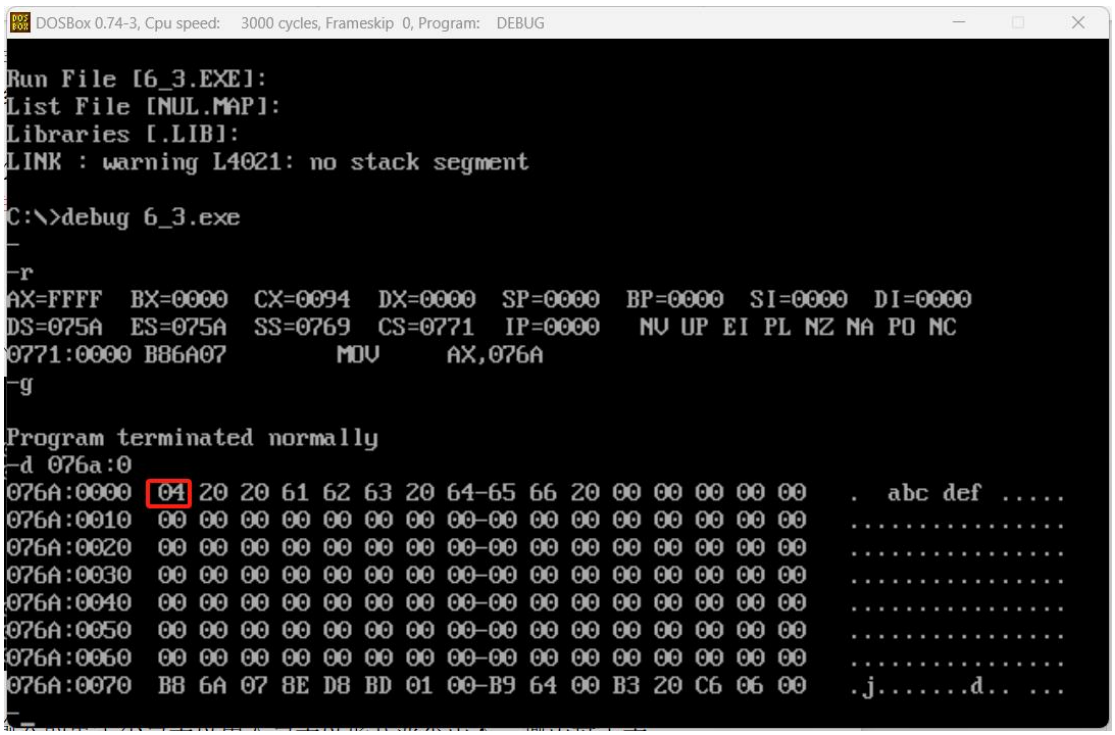
参考代码如下：

5_6.asm | 6_1.asm | 6_2.asm | 6_2_2.asm | 6_3.ASM

```
DATA SEGMENT
count db 0
str db 20h,20h,'abc def',20h,90 dup (0)
DATA ENDS

CODE SEGMENT
assume cs:CODE, ds:DATA
Start:
    mov ax, DATA
    mov ds, ax

    mov bp, offset str
    mov cx, 100
    mov bl, 20h
    mov count,0

count_loop:
    mov al, [bp]
    cmp al, bl
    jnz not_space
    inc count
not_space:
    inc bp
    loop count_loop

    MOV  ah, 4ch
    Int  21h
CODE ENDS

END Start
```

按上面的测试数据，str 中有 4 个空格符，所以测试结果如下。

```
X:\>debug y3.exe
-r
AX=FFFF  BX=0000  CX=0090  DX=0000  SP=0000  BP=0000  SI=0000  DI=0000
DS=075A  ES=075A  SS=0769  CS=0771  IP=0000   NV UP EI PL NZ NA PO NC
0771:0000 B86A07        MOV     AX,076A
-g

Program terminated normally
-d 076a:0
076A:0000  04 20 20 61 62 63 20 64-65 66 20 00 00 00 00 00   .  abc def
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   ................
```



4. 试编写一个程序实现从键盘输入的单个小写字母用大写字母形式显示出来，输出每个字母后都要求换行，最后输入!退出。

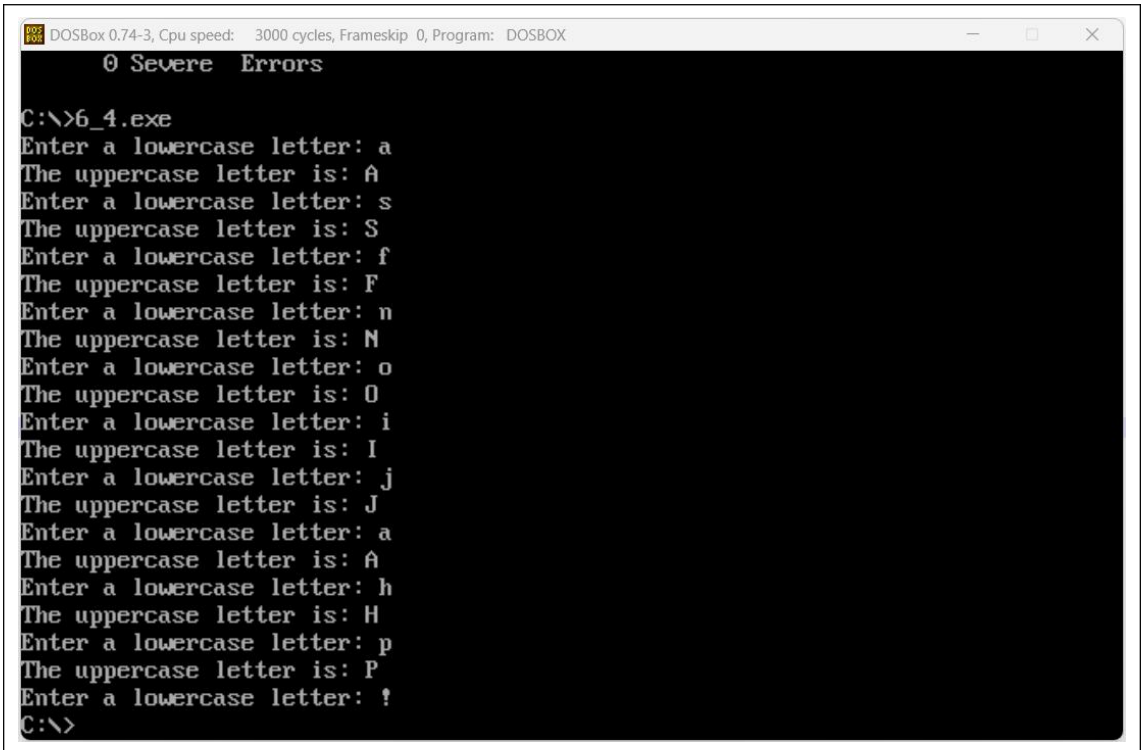
提示: 输入单个字母用 01 号子功能，输出用 02 号子功能。连续输出回车换行符 0DH, 0AH 即可实现换行。

```
MOV   AH, 01     ;调用 1 号子功能，输入单个字符
INT   21H        ; 输入字符放在 AL 中

MOV   DL, 'A'    ; 待输出字符放在 DL 中
MOV   AH, 02     ; 调用 2 号子功能，输出 DL 中的字符
INT   21H
```

参考代码如下:

5_6.asm | 6_1.asm | 6_2.asm | 6_2_2.asm | 6_3.ASM | 6_4.asm

```
DATA SEGMENT
input db 'Enter a lowercase letter: $'
output db 0DH, 0AH, 'The uppercase letter is: $'
DATA ENDS

CODE SEGMENT
assume cs:CODE, ds:DATA
Start:
    mov ax, DATA
    mov ds, ax

input_loop:

    mov ah, 09h
    mov dx, offset input
    int 21h


    mov ah, 01h
    int 21h

    cmp al, '!'
    je exit_program

    cmp al, 'a'
    jl input_loop
    cmp al, 'z'
    jg input_loop
    sub al, 20h


    mov ah, 09h
    mov dx, offset output
    int 21h

    mov dl,al
    mov ah, 02h
    int 21h

    mov ah, 02h
    mov dl, 0DH
    int 21h
    mov dl, 0AH
    int 21h

    jmp input_loop

exit_program:
    mov ah, 4ch
    int 21h
CODE ENDS

END Start

```


DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
0 Severe Errors
C:\>6_4.exe
Enter a lowercase letter: a
The uppercase letter is: A
Enter a lowercase letter: s
The uppercase letter is: S
Enter a lowercase letter: f
The uppercase letter is: F
Enter a lowercase letter: n
The uppercase letter is: N
Enter a lowercase letter: o
The uppercase letter is: O
Enter a lowercase letter: i
The uppercase letter is: I
Enter a lowercase letter: j
The uppercase letter is: J
Enter a lowercase letter: a
The uppercase letter is: A
Enter a lowercase letter: h
The uppercase letter is: H
Enter a lowercase letter: p
The uppercase letter is: P
Enter a lowercase letter: !
C:\>