

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
<p>一、实验目的</p> <p>1、汇编语言编程</p>					
<p>二、实验环境</p> <p>Windows 2011, DOSBox</p>					

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**。

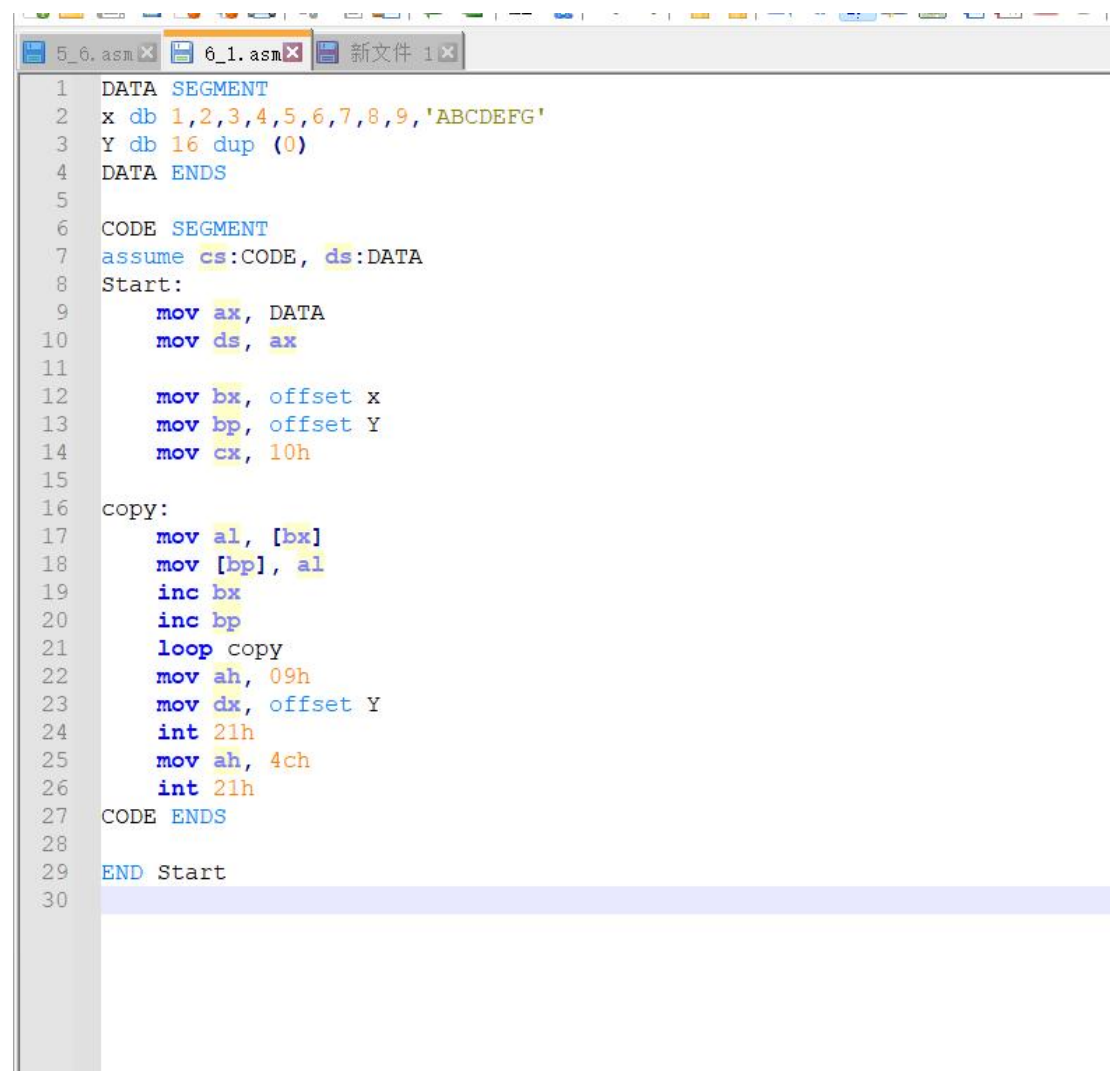
data segment

x db 1,2,3,4,5,6,7,8,9,'ABCDEFG'

Y db 16 dup (0)

data ends

.....



The screenshot shows an assembly code editor with three tabs: '5_6.asm', '6_1.asm', and '新文件 1'. The '6_1.asm' tab is active and displays the following assembly code:

```
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
```

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987. All rights reserved.

Object filename [6_1.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

51688 + 464856 Bytes symbol space free

0 Warning Errors
0 Severe Errors

C:\>link 6_1.obj

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

Run File [6_1.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>6_1.exe
E0v+ ABCDEFG7óEÄt  J  J  èêF CEF≈-  || =!-L=!U-^♦&iG+ëF■δLt◀i-πi6f
C:\>_

```

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
076C:0017 B409      MOV     AH,09
-t
AX=0947 BX=0010 CX=0000 DX=0000 SP=0000 BP=0020 SI=0000 DI=0000
DS=076A ES=075A SS=0769 CS=076C IP=0019  NU UP EI PL NZ AC PO NC
076C:0019 BA1000     MOV     DX,0010
-d 076a:0010
076A:0010  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
076A:0020  B8 6A 07 8E D8 BB 00 00-BD 10 00 B9 10 00 8A 07   .j.....
076A:0030  88 46 00 43 45 E2 F7 B4-09 BA 10 00 CD 21 B4 4C   .F.CE.....!.L
076A:0040  CD 21 50 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04   .!P..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....
076A:0080  FA FE 81 E6 FF 00 C6 82-FB FE 00 2B C0 50 8D 86   .....+.P..
-d 076a:0000
076A:0000  01 02 03 04 05 06 07 08-09 41 42 43 44 45 46 47   .....ABCDEFGH
076A:0010  00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
076A:0020  B8 6A 07 8E D8 BB 00 00-BD 10 00 B9 10 00 8A 07   .j.....
076A:0030  88 46 00 43 45 E2 F7 B4-09 BA 10 00 CD 21 B4 4C   .F.CE.....!.L
076A:0040  CD 21 50 E8 EA 48 83 C4-04 50 E8 7B 0E 83 C4 04   .!P..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....
-

```

- 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.asm6_1.asm6_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
```

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

```
075A:1260 07 5E 5F 8B E5 5D C3 90-55 8B EC 56 FF 76 04 E8 ^_...1..U..U.v..
075A:1270 5E C2 83 C4 02 8B 5E 04-80 7F 02 3A 74 06 8B C3 ^.....^.....:t...
075A:1280 5E 5D C3 90 8B 5E 04 8B-F3 8A 04 2C 02 8B 47 02 ^1...^.....G.
075A:1290 8B C3 05 02 00 5E 5D C3-B8 00 04 50 2B C0 50 B8 .....^1....P+.P.
075A:12A0 14 19 50 E8 22 62 83 C4-06 C3 55 8B EC 83 EC 04 ..P."b....U....
075A:12B0 57 56 8B 5E WJ.^
-Q
C:\>debug 6_2.exe
-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 B10B MOV CL,0B
-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=0005 NV UP EI PL NZ NA PO NC
076A:0005 D3C0 ROL AX,CL
-T
AX=3412 BX=0000 CX=000B 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
-
```

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

```

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

```

0 Warning Errors
0 Severe Errors

C:\>link 6_2_2.OBJ

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

Run File [6_2_2.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>debug 6_2_2.exe
-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  NU 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  NU UP EI PL NZ NA PO NC
076A:0005 B44C          MOV   AH,4C

```

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 x 6_1.asm x 6_2.asm x 6_2_2.asm x 6_3.ASM x
1 DATA SEGMENT
2 count db 0
3 str db 20h,20h,'abc def',20h,90 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 bp, offset str
13    mov cx, 100
14    mov bl, 20h
15    mov count,0
16
17 count_loop:
18     mov al, [bp]
19     cmp al, bl
20     jnz not_space
21     inc count
22 not_space:
23     inc bp
24     loop count_loop
25
26     MOV ah, 4ch
27     Int 21h
28 CODE ENDS
29
30 END Start
31
32

```

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 NU 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 .....
076A:0020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....

```

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
Run File [6_3.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

C:\>debug 6_3.exe

-r
AX=FFFF BX=0000 CX=0094 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=0771 IP=0000  NU 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 .....
076A:0020 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 .....
076A:0040 00 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 .....
076A:0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
076A:0070 B8 6A 07 8E D8 BD 01 00-B9 64 00 B3 20 C6 06 00 .j.....d.. ...

```

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 x 6_1.asm x 6_2.asm x 6_2_2.asm x 6_3.ASM x 6_4.asm x
1 DATA SEGMENT
2 input db 'Enter a lowercase letter: $'
3 output db 0DH, 0AH, 'The uppercase letter is: $'
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 input_loop:
13
14    mov ah, 09h
15    mov dx, offset input
16    int 21h
17
18
19    mov ah, 01h
20    int 21h
21
22    cmp al, '!'
23    je exit_program
24
25    cmp al, 'a'
26    jl input_loop
27    cmp al, 'z'
28    jg input_loop
29    sub al, 20h
30
31
32    mov ah, 09h
33    mov dx, offset output
34    int 21h
35
36    mov dl, al
37    mov ah, 02h
38    int 21h
39
40    mov ah, 02h
41    mov dl, 0DH
42    int 21h
43    mov dl, 0AH
44    int 21h
45
46    jmp input_loop
47
48 exit_program:
49    mov ah, 4ch
50    int 21h
51 CODE ENDS
52
53 END Start
54
```

```
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:\>
```