

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

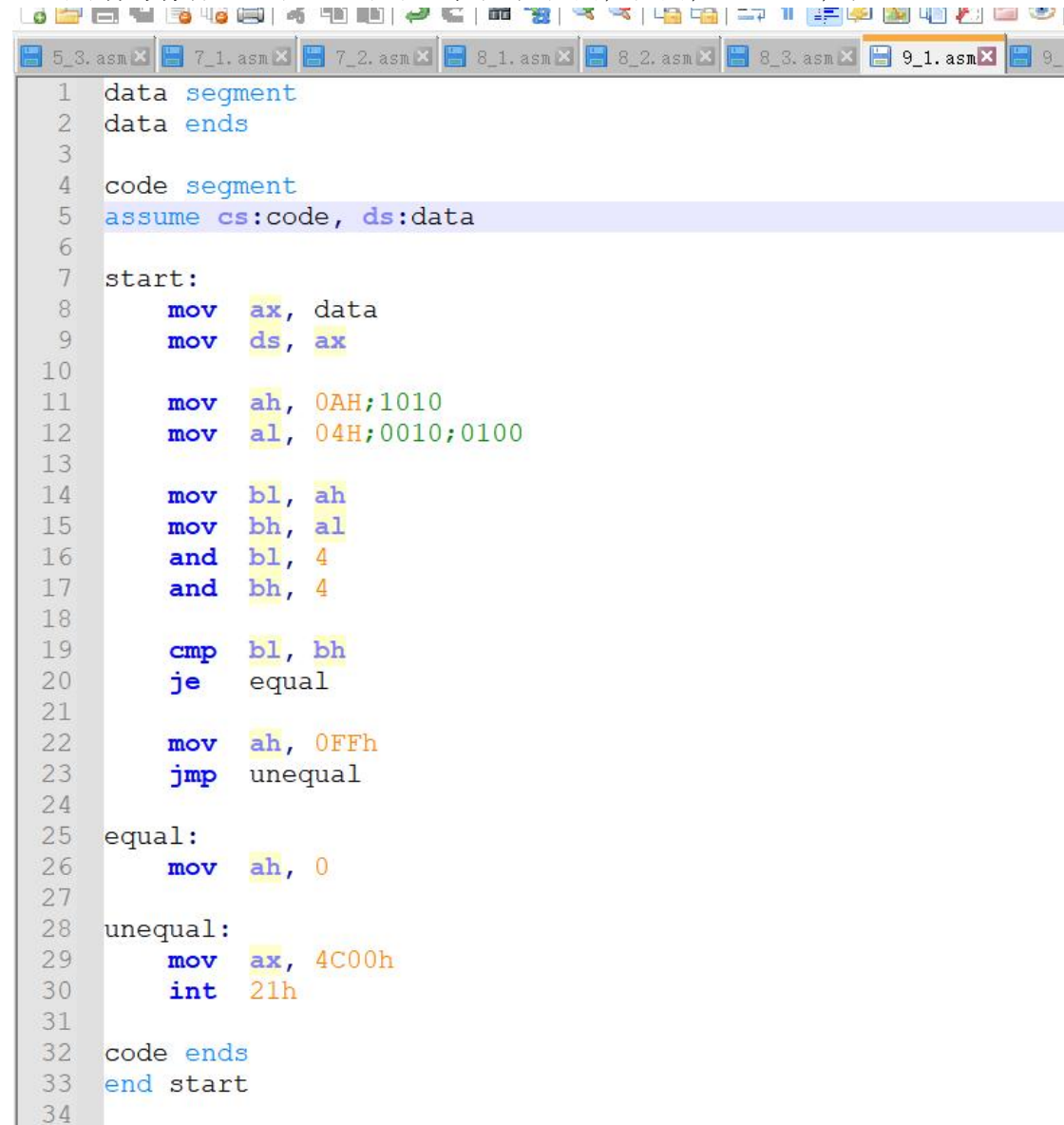
姓 名: 杨跃浙

指导老师: 杜库

实验项目	实验 9- 第九次上机实验			实验次序	09
实验地点	笃行南楼 A104	参与人员	杨跃浙	实验日期	06.05
一、实验目的 汇编语言程序设计					
二、实验环境 Windows 2011, DOSBox					

三、实验内容

1、判断寄存器 AH 和 AL 中第 3 位是否相同，相同，AH 置 0，否则 AH 置全 1。



```
1 data segment
2 data ends
3
4 code segment
5 assume cs:code, ds:data
6
7 start:
8     mov ax, data
9     mov ds, ax
10
11     mov ah, 0AH;1010
12     mov al, 04H;0010;0100
13
14     mov bl, ah
15     mov bh, al
16     and bl, 4
17     and bh, 4
18
19     cmp bl, bh
20     je equal
21
22     mov ah, 0FFh
23     jmp unequal
24
25 equal:
26     mov ah, 0
27
28 unequal:
29     mov ax, 4C00h
30     int 21h
31
32 code ends
33 end start
34
```

```

1 data segment
2 data ends
3
4 code segment
5 assume cs:code, ds:data
6
7 start:
8 mov ax, data
9 mov ds, ax
10
11 mov ah, 0AH
12 mov al, 02H
13
14 mov bl, ah
15 mov bh, al
16 and bl, 4
17 and bh, 4
18
19 cmp bl, bh
20 je equal
21
22 mov ah, 0FFh
23 jmp unequal
24
25 equal:
26 mov ah, 0
27
28 unequal:
29 mov ax, 4C00h
30 int 21h
31
32 code ends
33 end start
34

```

```

1 data segment
2 data ends
3
4 code segment
5 assume cs:code, ds:data
6
7 start:
8 mov ax, data
9 mov ds, ax
10
11 mov ah, 0AH:010
12 mov al, 04H:0010;0100
13
14 mov bl, ah
15 mov bh, al
16 and bl, 4
17 and bh, 4
18
19 cmp bl, bh
20 je equal
21
22 mov ah, 0FFh
23 jmp unequal
24
25 equal:
26 mov ah, 0
27
28 unequal:
29 mov ax, 4C00h
30 int 21h
31
32 code ends
33 end start
34

```

2、以 BUF 为首址的字节单元中，存放了 COUNT 个无符号数，找出其中最大数并送入 MAX 单元中。

BUF DB 5,6,7,58H,62,45H,127,

COUNT EQU \$-BUF

MAX DB ?

```

5_3.asm 7_1.asm 7_2.asm 8_1.asm 8_2.asm 8_3.asm 9_1.asm 9_2.asm
1 data segment
2     BUF DB 5, 6, 7, 58H, 62, 45H, 127
3     COUNT EQU $-BUF
4     MAX DB ?
5 data ends
6
7 code segment
8 assume cs:code, ds:data
9
10 start:
11     mov ax, data
12     mov ds, ax
13
14     mov cx, COUNT
15     mov si, OFFSET BUF
16     mov al, [si]
17     inc si
18     dec cx
19
20 find_max:
21     cmp cx, 0
22     je get_max
23     mov bl, [si]
24     inc si
25     dec cx
26     cmp al, bl
27     ja find_max
28     mov al, bl
29     jmp find_max
30
31 get_max:
32     mov MAX, al
33
34     mov ax, 4C00h
35     int 21h
36
37 code ends
38 end start
39

```

5_3.asm 7_1.asm 7_2.asm 8_1.asm 8_2.asm 8_3.asm 9_1.asm 9_2.asm

```

1 data segment
2     BUF DB 5, 6, 7, 58H, 62, 45H, 127
3     COUNT EQU $-BUF
4     MAX DB ?
5 data ends
6
7 code segment
8 assume cs:code, ds:data
9
10 start:
11     mov ax, data
12     mov ds, ax
13
14     mov cx, COUNT
15     mov si, OFFSET BUF
16     mov al, [si]
17     inc si
18     dec cx
19
20 find_max:
21     cmp cx, 0
22     je get_max
23     mov bl, [si]
24     inc si
25     dec cx
26     cmp al, bl
27     ja find_max
28     mov al, bl
29     jmp find_max
30
31 get_max:
32     mov MAX, al
33
34     mov ax, 4C00h
35     int 21h
36
37 code ends
38 end start
39

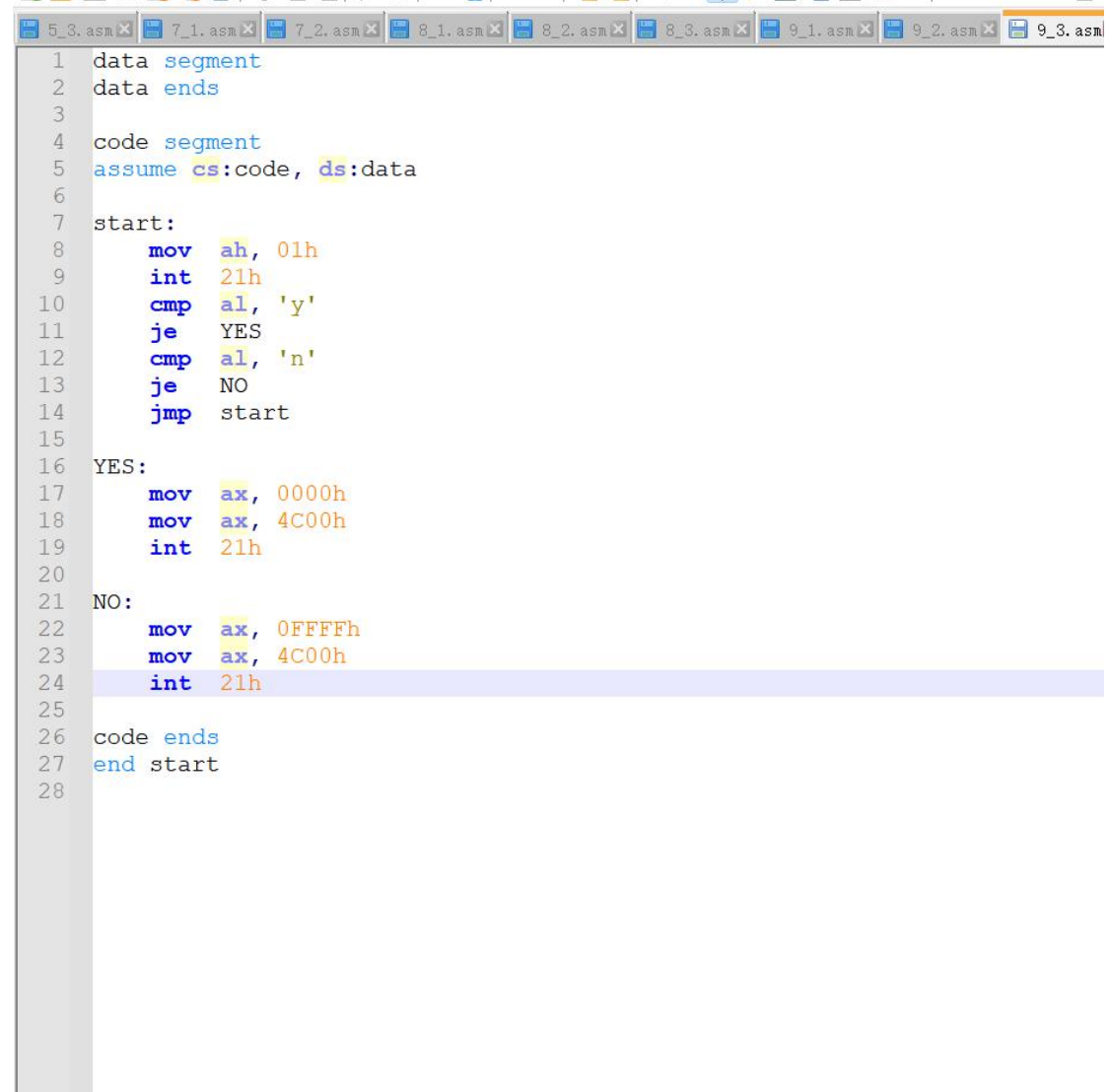
```

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

AX=077F	BX=007F	CX=0000	DX=0000	SP=0000	BP=0000	SI=0007	DI=0000
DS=076A	ES=075A	SS=0769	CS=076B	IP=0012	NU	UP	EI PL ZR NA PE NC
076B:0012	740C	JZ	0020				
-t							
AX=077F	BX=007F	CX=0000	DX=0000	SP=0000	BP=0000	SI=0007	DI=0000
DS=076A	ES=075A	SS=0769	CS=076B	IP=0020	NU	UP	EI PL ZR NA PE NC
076B:0020	A20700	MOV	[0007],AL	DS:0007=00			
-t							
AX=077F	BX=007F	CX=0000	DX=0000	SP=0000	BP=0000	SI=0007	DI=0000
DS=076A	ES=075A	SS=0769	CS=076B	IP=0023	NU	UP	EI PL ZR NA PE NC
076B:0023	B8004C	MOV	AX,4C00				
-t							
AX=4C00	BX=007F	CX=0000	DX=0000	SP=0000	BP=0000	SI=0007	DI=0000
DS=076A	ES=075A	SS=0769	CS=076B	IP=0026	NU	UP	EI PL ZR NA PE NC
076B:0026	CD21	INT	21				
-t							
AX=4C00	BX=007F	CX=0000	DX=0000	SP=FFFA	BP=0000	SI=0007	DI=0000
DS=076A	ES=075A	SS=0769	CS=F000	IP=14A0	NU	UP	EI PL ZR NA PE NC
F000:14A0	FB	STI					

Assembly language source file
length: 533 lines: 39
Ln: 29 Col: 12 Pos: 436
Windows (CR LF) UTF-8
INS

- 3、编指程序段, 调用系统功能的 1 号子功能输入一个字符, 并判断输入的字符。如字符是"y", 则转向 YES 程序段; 如字符是"n", 则转向 NO 程序段。(YES 和 NO 分别是两程序段入口处的标号)



```
1 data segment
2 data ends
3
4 code segment
5 assume cs:code, ds:data
6
7 start:
8     mov ah, 01h
9     int 21h
10    cmp al, 'y'
11    je YES
12    cmp al, 'n'
13    je NO
14    jmp start
15
16 YES:
17     mov ax, 0000h
18     mov ax, 4C00h
19     int 21h
20
21 NO:
22     mov ax, 0FFFFh
23     mov ax, 4C00h
24     int 21h
25
26 code ends
27 end start
28
```

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
J
AX=0179 BX=0000 CX=001E DX=0000 SP=FFFA BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=F000 IP=14A5  NU UP EI PL NZ NA PO NC
F000:14A5 CF      IRET
-t

AX=0179 BX=0000 CX=001E DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0004  NU UP EI PL NZ NA PO NC
076A:0004 3C79      CMP     AL,79
-t

AX=0179 BX=0000 CX=001E DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0006  NU UP EI PL ZR NA PE NC
076A:0006 7406      JZ      000E
-t

AX=0179 BX=0000 CX=001E DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=000E  NU UP EI PL ZR NA PE NC
076A:000E B80000     MOV     AX,0000
-t

AX=0000 BX=0000 CX=001E DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0011  NU UP EI PL ZR NA PE NC
076A:0011 B8004C     MOV     AX,4C00
-

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
AX=01FF BX=0000 CX=001E DX=0000 SP=FFFA BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=F000 IP=14A1  NU UP EI PL NZ NA PO NC
F000:14A1 FE3B      ???     [BX+SI]      DS:0000=CD
-t

n
AX=016E BX=0000 CX=001E DX=0000 SP=FFFA BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=F000 IP=14A5  NU UP EI PL NZ NA PO NC
F000:14A5 CF      IRET
-t

AX=016E BX=0000 CX=001E DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0004  NU UP EI PL NZ NA PO NC
076A:0004 3C79      CMP     AL,79
-t

AX=016E BX=0000 CX=001E DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0006  NU UP EI NG NZ NA PE CY
076A:0006 7406      JZ      000E
-t

AX=016E BX=0000 CX=001E DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0008  NU UP EI NG NZ NA PE CY
076A:0008 3C6E      CMP     AL,6E
-

```

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
AX=016E BX=0000 CX=001E DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0006  NU UP EI NG NZ NA PE CY
076A:0006 7406 JZ 000E
-t
AX=016E BX=0000 CX=001E DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0008  NU UP EI NG NZ NA PE CY
076A:0008 3C6E CMP AL,6E
-t
AX=016E BX=0000 CX=001E DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=000A  NU UP EI PL ZR NA PE NC
076A:000A 740A JZ 0016
-t
AX=016E BX=0000 CX=001E DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0016  NU UP EI PL ZR NA PE NC
076A:0016 B8FFFF MOV AX,FFFF
-t
AX=FFFF BX=0000 CX=001E DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=075A ES=075A SS=0769 CS=076A IP=0019  NU UP EI PL ZR NA PE NC
076A:0019 B8004C MOV AX,4C00

```

4、在 BUF1 和 BUF2 两个数据区中，各定义有 8 个带符号字数据，试编制一完整的源程序，求它们对应项之和(即 BUF1 的第 0 项对 BUF2 的第 0 项，BUF1 的第 1 项对 BUF2 的第 1 项)，并将和数存入以 S 为首址的数据区中。

DATA SEGMENT

BUF1 DW -5,2,4,-10,9,-8,10,20

BUF2 DW 4,-2,24,45,-25,20,30,10

S DW 8 DUP(0)

DATA ENDS


```

5_3.asm 7_1.asm 7_2.asm 8_1.asm 8_2.asm 8_3.asm 9_1.asm 9_2.asm 9_3.asm 9_4.asm
1 data segment
2   BUF1 DW -5, 2, 4, -10, 9, -8, 10, 20
3   BUF2 DW 4, -2, 24, 45, -25, 20, 30, 10
4   S DW 8 DUP (0)
5 data ends
6
7 code segment
8 assume cs:code, ds:data
9
10 start:
11   mov ax, data
12   mov ds, ax
13   mov es, ax
14
15   mov cx, 8
16   mov si, OFFSET BUF1
17   mov di, OFFSET BUF2
18   mov bx, OFFSET S
19
20 sum_loop:
21   mov ax, [si]
22   add ax, [di]
23   mov [bx], ax
24
25   add si, 2
26   add di, 2
27   add bx, 2
28   loop sum_loop
29
30   mov ax, 4C00h
31   int 21h
32
33 code ends
34 end start
35

```

```

DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DEBUG
-t
AX=001E BX=002E CX=0001 DX=0000 SP=0000 BP=0000 SI=0010 DI=0020
DS=076A ES=076A SS=0769 CS=076D IP=001F  NV UP EI PL NZ AC PO NC
076D:001F 83C302      ADD     BX,+02
-t
AX=001E BX=0030 CX=0001 DX=0000 SP=0000 BP=0000 SI=0010 DI=0020
DS=076A ES=076A SS=0769 CS=076D IP=0022  NV UP EI PL NZ AC PE NC
076D:0022 E2EF      LOOP    0013
-t
AX=001E BX=0030 CX=0000 DX=0000 SP=0000 BP=0000 SI=0010 DI=0020
DS=076A ES=076A SS=0769 CS=076D IP=0024  NV UP EI PL NZ AC PE NC
076D:0024 B8004C     MOV     AX,4C00
-d 0020
076A:0020 FF FF 00 00 1C 00 23 00 F0 FF 0C 00 2B 00 1E 00  .....#.....(...
076A:0030 B8 6A 07 8E D8 8E C0 B9-08 00 BE 00 00 BF 10 00  .j.....
076A:0040 BB 20 00 8B 04 03 05 89-07 83 C6 02 83 C7 02 83  .....
076A:0050 C3 02 E2 EF B8 00 4C CD-21 5E FC 26 8A 47 0C 2A  ....L.!^.&.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..
076A:0090 FB FE 50 E8 08 6A 83 C4-04 0B C0 75 03 E9 A5 00  ..P..j.....u...

```