汇编语言程序设计
实验报告

东北大學

实验名称	上机过程及顺序结构与分支结构程序设计实验
班　　级	软信-1503
学　　号	20155362
姓　　名	薛旗
日　　期	2016.10.24
成　　绩	
评 阅 人	

软件学院

一、 实验目的与意义

实验一：熟悉上机练习过程和伪指令实验

1. 掌握汇编语言程序汇编、连接的方法；

2. 熟悉 DEBUG 的基本命令及其功能（U、D、T、P、G、A、E、R）；

3. 熟悉并掌握伪指令的用法，能够验证指令的功能

实验二：顺序结构程序设计实验

1.掌握顺序程序的结构及设计调试方法

2.了解用 EDIT 编辑程序进行汇编语言的编写

3.熟悉 MASM 5.0 对汇编程序源代码进行编译连接，掌握使用 DEBUG 对程序进行调试的方法。

实验三：分支程序设计实验

1.了解分支程序的设计方法，熟练运用相应指令进行分支程序设计。

2.掌握用 DEBUG 查看程序运行结果

二、 实验环境

操作系统：Windows 7

调试软件：Masm for Windows 集成实验环境

版本号：MASM 5.0

上机地点：信息 B405

三、 实验的预习内容

实验一：熟悉上机练习过程和伪指令实验

(1) 课前预习主要内容

DEBUG 常用命令：

①修改命令 R

★显示所有寄存器和标志位状态；

★显示当前 CS：IP 指向的指令。

②汇编命令 A

A[地址]；从指定的地址开始输入符号指令；如省略地址，则接着上一个 A 命令的最后一个单元开始；若第一次使用 A 命令省略地址，则从当前 CS:IP 开始（通常是 CS：100）。

③反汇编命令 U

U [地址]；从指定地址开始反汇编 32 个字节的机器指令；省略地址时,则接着上一个 U 命令的最后一个单元开始；若第一次使用 U 命令省略地址，则从当前 CS:IP 开始（通常是 CS：100）。

④运行程序命令 G

从 CS:IP 指向的指令开始执行程序，直到程序结束或遇到 INT 3。

⑤跟踪命令（单步执行命令）T

a. T；从当前 IP 开始执行一条指令。

b. T 数值；从当前 IP 开始执行多条指令。

⑥修改内存单元内容 E

默认为 DS 段

(2) 实验思路：

按照 PPT 要求打入代码，进行汇编连接，运行程序，通过调试，使用 DEBUG 命令 U,D,T,P,G,A,E 等查看程序及寄存器的使用情况。

实验二：顺序结构程序设计实验

（1）课前预习主要内容

①换码指令 XLAT

操作:BX 和 AL 内容之和指出的内存字节单元的内容送到 AL 中。

AL←(BX+AL)

XLAT ； al←ds:[bx+al]

受影响的状态标志位: 没有

说明: XLAT 指令用于查表。表的开始地址即表头地址由 BX 寄存器给出。AL 中的原始值是要寻址的表中元素地址的位移量,规定表中第一个字节的位移量为 0。

②顺序结构

程序按顺序执行，不发生人和转移

（2）实验思路

通过十进制与 ASCII 的转化关系和十六进制与 ASCII 的转化关系完成实验。将结果转化为一个高位和低位存在寄存器分别输出。

实验三：分支程序设计实验

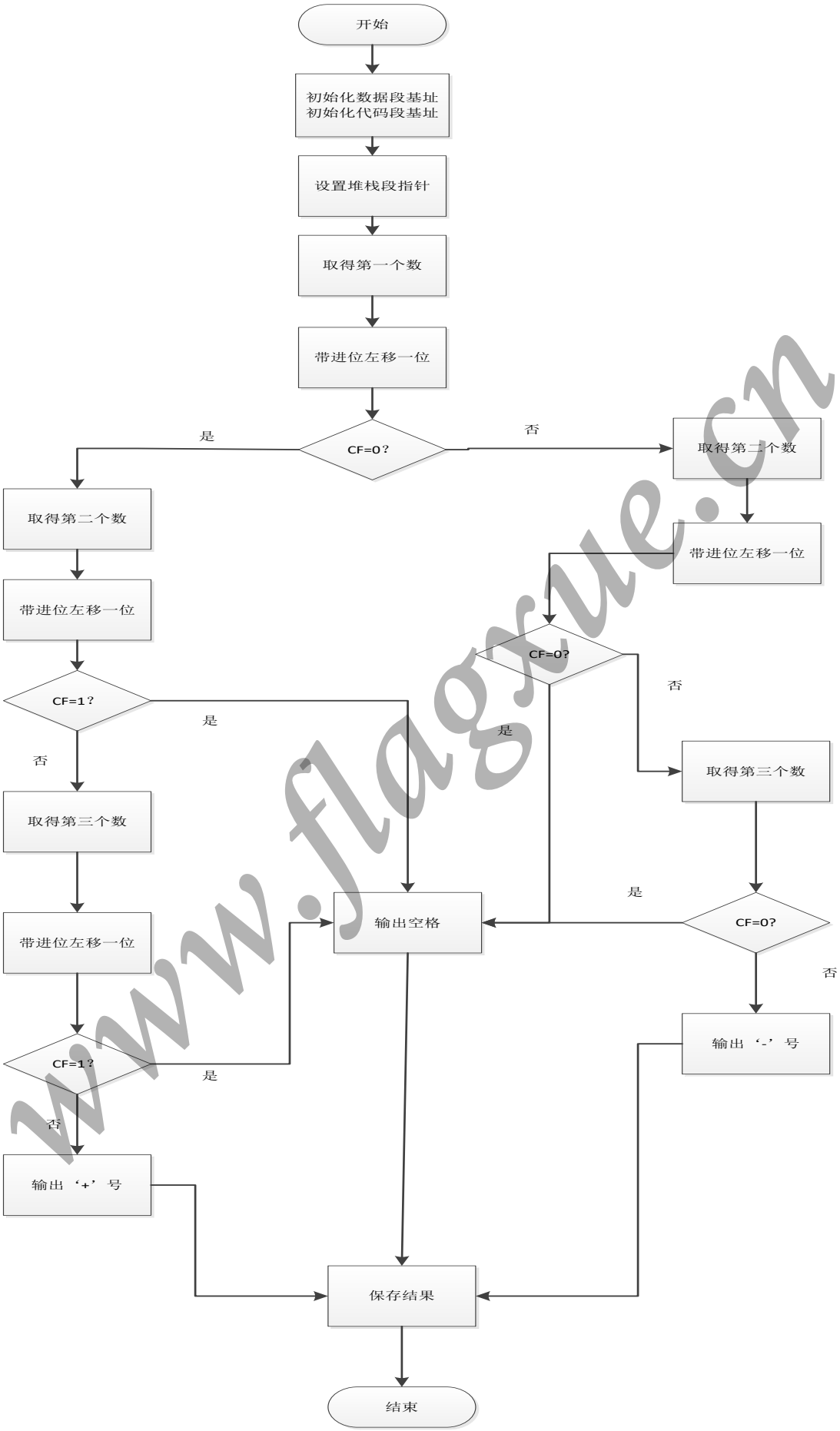
（1）课前预习主要内容

分支结构是指计算机根据实际情况或条件，作出判断和选择，转而执行不同的程序段的一种结构。本实验主要运用了分支结构。

（2）实验思路

通过 RCL 指令将数据的最高位送入 CF 中，通过判断 CF 位为 0 还是 1 判断数据的正负情况，依次进行比较，按实验要求输出指定结果。

（3）流程图



四、 实验的步骤与调试方法

实验一：熟悉上机练习过程和伪指令实验

实验步骤：在 MASM 中输入 PPT 中的代码，运行 DEBUG，在 DEBUG 中输入指令，观察结果。

实验二：顺序结构程序设计实验

定义数据段定义数字，输入数字并转化成对应的地址偏移量，将数据地址传送至 BX，求得平方后除以 10 得商和余数。

实验过程中遇到问题，使用 T 单步执行命令查看程序运行情况。

实验三：分支程序设计实验

首先得到第一个数据的有效地址，将其送入寄存器得到 CF 的值，之后同样方法得到第二个数，通过 RCL 指令将最高位送入 CF，与第一位进行比较。第三个数据同理，依次进行比较，按实验要求输出结果。

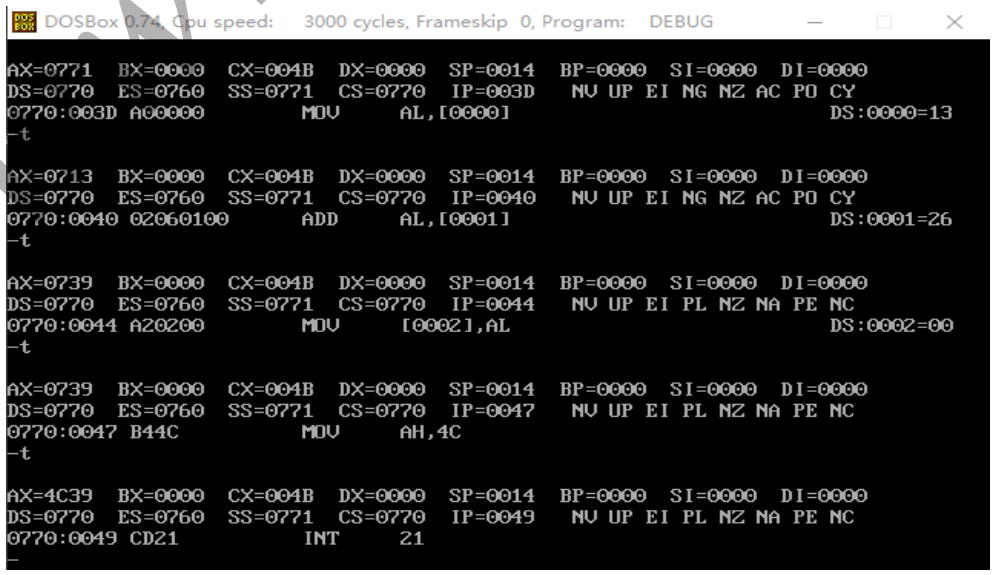
实验刚开始时不能很好运用条件判断，对分支语句的运用不娴熟。通过实验学习，更好的了解了各种跳转指令。

五、 实验数据与实验结果

实验一：熟悉上机练习过程和伪指令实验

题目一：

1. 使用指令 T 进行单步执行操作



2.用 DEBUG 中的 E 命令修改原始数据，G 命令执行程序（将 13 改 4，将 26 改为 10）

```
-E DS:0000 4 10
-_
```

3.使用 D 命令查看数据是否更改（观察到数据已更改）

```
-D DS:0000
0770:0000  04 10 39 00 00 00 00 00-00 00 00 00 00 00 00 00   ..9.............
0770:0010  00 00 00 00 00 00 00 00-00 00 39 4C 00 00 49 00   ..........9L..I.
0770:0020  70 07 A6 01 00 00 00 00-00 00 00 00 00 00 00 00   p...............
0770:0030  B8 70 07 8E D8 B8 71 07-8E D0 BC 14 00 A0 00 00   .p....q.........
0770:0040  02 06 01 00 A2 02 00 B4-4C CD 21 00 00 00 00 00   ........L.!.....
0770:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0770:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0770:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
- ▲_
```

4.使用 G 命令执行程序（程序正常结束，程序运行成功）

```
-G 16

Program terminated normally
-_
```

5.使用 D 命令查看结果是否更改（发现结果更改 4）

```
-D DS:0002
0770:0000        14 00 00 00 00 00-00 00 00 00 00 00 00 00   ..............
0770:0010  00 00 00 00 00 00 00 00-00 00 14 4C 00 00 49 00   ..........L..I.
0770:0020  70 07 A6 01 00 00 00 00-00 00 00 00 00 00 00 00   p...............
0770:0030  B8 70 07 8E D8 B8 71 07-8E D0 BC 14 00 A0 00 00   .p....q.........
0770:0040  02 06 01 00 A2 02 00 B4-4C CD 21 00 00 00 00 00   ........L.!.....
0770:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0770:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0770:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0770:0080  00 00                                            ..
```

6.用 DEBUG 中的 A 命令修改程序，使之由加法改为减法

①-u 命令反汇编，找到 ADD 地址

```
-u
0770:0030 B87007        MOV     AX,0770
0770:0033 8ED8          MOV     DS,AX
0770:0035 B87107        MOV     AX,0771
0770:0038 8ED0          MOV     SS,AX
0770:003A BC1400        MOV     SP,0014
0770:003D A00000        MOV     AL,[0000]
0770:0040 02060100      ADD     AL,[0001]
0770:0044 A20200        MOV     [0002],AL
0770:0047 B44C          MOV     AH,4C
0770:0049 CD21          INT     21
0770:004B 0000          ADD     [BX+SI],AL
0770:004D 0000          ADD     [BX+SI],AL
0770:004F 0000          ADD     [BX+SI],AL
```

②使用 A 命令，对 ADD 地址进行修改，将 ADD 变为 SUB（观察到 ADD 指令变成了 SUB 指令）

```
-u 0770:0030
0770:0030 B87007        MOV     AX,0770
0770:0033 8ED8          MOV     DS,AX
0770:0035 B87107        MOV     AX,0771
0770:0038 8ED0          MOV     SS,AX
0770:003A BC1400        MOV     SP,0014
0770:003D A00000        MOV     AL,[0000]
0770:0040 2A060100      SUB     AL,[0001]
0770:0044 A20200        MOV     [0002],AL
0770:0047 B44C          MOV     AH,4C
0770:0049 CD21          INT     21
0770:004B 0000          ADD     [BX+SI],AL
0770:004D 0000          ADD     [BX+SI],AL
0770:004F 0000          ADD     [BX+SI],AL
```

③-t 单步执行结果

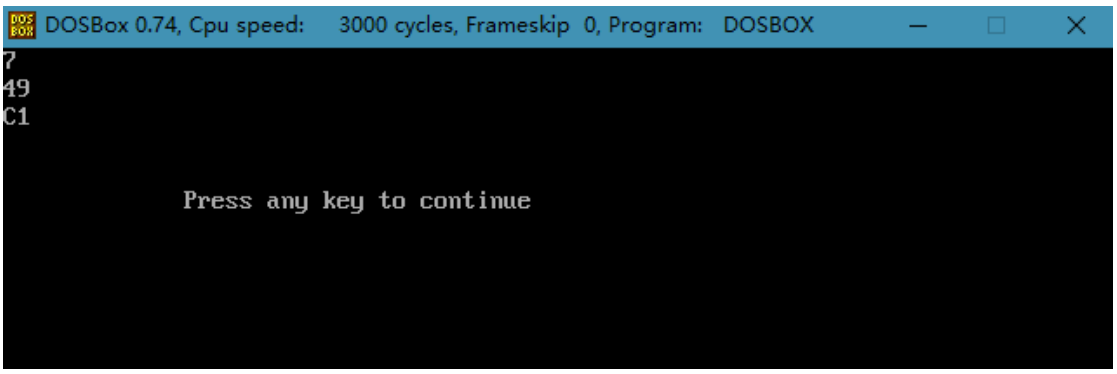
```
AX=0771  BX=0000  CX=004B  DX=0000  SP=0014  BP=0000  SI=0000  DI=0000
DS=0770  ES=0760  SS=0771  CS=0770  IP=003D   NV UP EI NG NZ AC PO CY
0770:003D A00000        MOV     AL,[0000]                      DS:0000=13
-t

AX=0713  BX=0000  CX=004B  DX=0000  SP=0014  BP=0000  SI=0000  DI=0000
DS=0770  ES=0760  SS=0771  CS=0770  IP=0040   NV UP EI NG NZ AC PO CY
0770:0040 2A060100      SUB     AL,[0001]                      DS:0001=26
-t

AX=07ED  BX=0000  CX=004B  DX=0000  SP=0014  BP=0000  SI=0000  DI=0000
DS=0770  ES=0760  SS=0771  CS=0770  IP=0044   NV UP EI NG NZ AC PE CY
0770:0044 A20200        MOV     [0002],AL                      DS:0002=00
-t
```

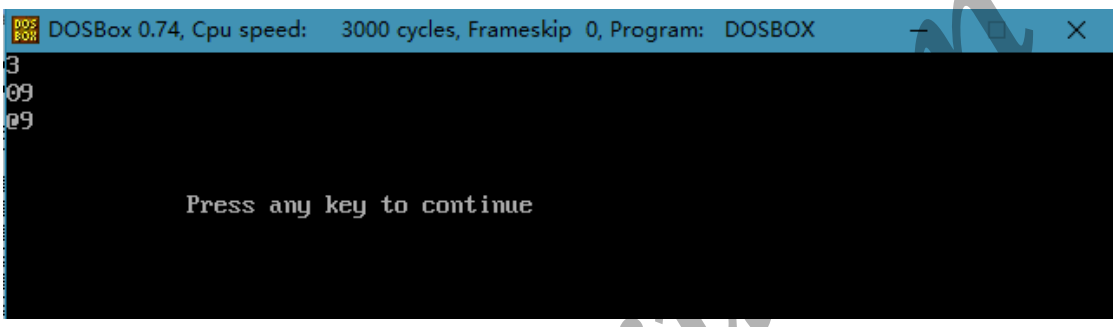
题目二：

使用 U 命令反汇编，使用 D 命令观察数据的存储情况

```
DOSBox 0.74, Cpu speed:   3000 cycles, Frameskip  0, Program:   DEBUG
0774:0000 B87007        MOV     AX,0770
0774:0003 8ED8          MOV     DS,AX
0774:0005 B87207        MOV     AX,0772
0774:0008 8ED0          MOV     SS,AX
0774:000A B44C          MOV     AH,4C
0774:000C CD21          INT     21
0774:000E 0000          ADD     [BX+SI],AL
0774:0010 0000          ADD     [BX+SI],AL
0774:0012 0000          ADD     [BX+SI],AL
0774:0014 0000          ADD     [BX+SI],AL
0774:0016 0000          ADD     [BX+SI],AL
0774:0018 0000          ADD     [BX+SI],AL
0774:001A 0000          ADD     [BX+SI],AL
0774:001C 0000          ADD     [BX+SI],AL
0774:001E 0000          ADD     [BX+SI],AL
-D DS:0000
0760:0000  CD 20 FF 9F 00 EA FF FF-AD DE 4F 03 A6 01 8A 03   . ........O.....
0760:0010  A6 01 17 03 A6 01 95 01-01 01 01 00 02 FF FF FF   ................
0760:0020  FF FF FF FF FF FF FF FF-FF FF FF FF 53 07 4C 01   ............S.L.
0760:0030  66 06 14 00 18 00 60 07-FF FF FF FF 00 00 00 00   f.....`.........
0760:0040  05 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0760:0050  CD 21 CB 00 00 00 00 00-00 00 00 00 00 00 00 00   .!..............
0760:0060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0760:0070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
-_
```

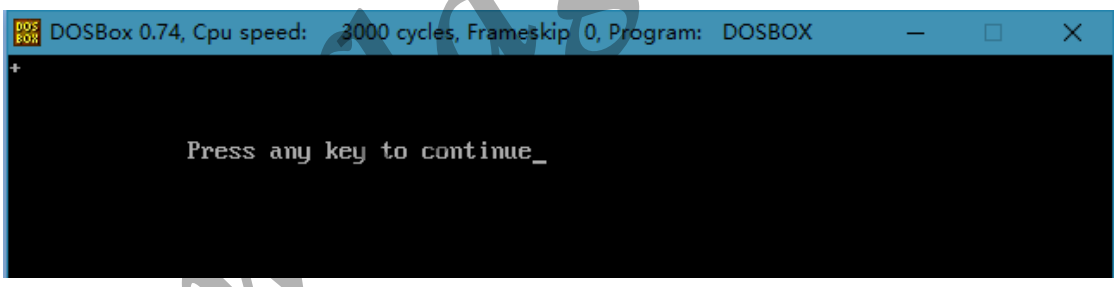
实验二：顺序结构程序设计实验





实验三：分支程序设计实验

（输入的三个数为 3,5,2 三个正数，故显示“+”号）



六、 实验用程序清单（要有注释）

实验一：

（题目一）

```
DSEG   SEGMENT                ;数据段开始
DATA1 DB 13H,26H
DATA2 DW 0
DSEG   ENDS                   ;数据段结束
```

```
SSEG    SEGMENT STACK          ;堆栈段开始
SKTOP DB 20 DUP(0)
SSEG    ENDS                   ;堆栈段结束

CSEG    SEGMENT                ;代码段开始
        ASSUME CS:CSEG,DS:DSEG
        ASSUME SS:SSEG
START: MOV AX,DSEG             ;初始化数据段基址
        MOV DS,AX
        MOV AX,SSEG            ;初始化代码段基址
        MOV SS,AX
        MOV SP,LENGTH SKTOP    ;设置堆栈指针
        MOV AL,DATA1
        ADD AL,DATA1+1
        MOV BYTE PTR DATA2,AL
        MOV AH,4CH
        INT 21H                ;返回DOS
CSEG    ENDS                   ;代码段结束
        END
```

（题目二）

```
    DATAS SEGMENT              ;数据段开始
    DATA1 DB 1,-1,'a'
```

```
DATA2 DW 2,-2,'B'
DATA3 DD 3,-3,'c'

DATAS ENDS                ;数据段结束

STACKS SEGMENT STACK      ;堆栈段开始
SKTOP DB 20 DUP(?)
STACKS ENDS               ;堆栈段结束

CODES SEGMENT             ;代码段开始
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS          ;初始化数据段基址
    MOV DS,AX
    MOV AX,STACKS         ;初始化堆栈段基址
    MOV SS,AX
    MOV AH,4CH
    INT 21H
CODES ENDS                ;代码段结束
END START                 ;源程序结束
```

实验二：

```
DSEG   SEGMENT            ;数据段开始
```

```
DATA   DB 0D,1D,4D,9D,16D,25D,36D,49D,64D,81D
DSEG   ENDS                    ;数据段结束

SSEG   SEGMENT STACK           ;代码段开始
SKTOP DB 20 DUP(?)
SSEG   ENDS                    ;堆栈段结束

CSEG   SEGMENT                 ;代码段开始
       ASSUME CS:CSEG,DS:DSEG
       ASSUME SS:SSEG
START: MOV AX,DSEG             ;初始化数据段基址
       MOV DS,AX
       MOV AX,SSEG             ;初始化代码段基址
       MOV SS,AX
       MOV SP,SIZE SKTOP
       MOV BX,OFFSET DATA      ;制表
       MOV AH,01H
       INT 21H                 ;从键盘读取数字
       PUSH AX;                ;保存输入的数

       MOV DL,0DH
       MOV AH,02H
```

```
INT 21H
MOV DL,0AH
MOV AH,02H
INT 21H                 ;换行

POP AX
SUB AL,30H              ;转化成0-9的数字
XLAT
PUSH AX                 ;AX入栈
XOR AH,AH
MOV CH,10
DIV CH
ADD AL,30H
MOV CH,AH
MOV DL,AL
MOV AH,02H
INT 21H                 ;转化成十进制
MOV DL,CH
ADD DL,30H
MOV AH,02H
INT 21H
```

```
MOV DL,0DH
MOV AH,02H
INT 21H
MOV DL,0AH
MOV AH,02H
INT 21H                ;输出结果

POP AX                 ;出栈
MOV DL,16
DIV DL
MOV CH,AH              ;除以16得余数
MOV CL,AL              ;除以16得商
ADD AL,30H
MOV DL,AL
MOV AH,02H
INT 21H                ;输出高位
ADD CH,30H
MOV DL,CH
MOV AH,02H             ;输出低位
INT 21H

MOV AH,4CH
```

```
        INT 21H                 ;返回DOS
CSEG    ENDS                    ;代码段结束
        END
```

实验三：

```
DSEG    SEGMENT                 ;数据段开始
NUM     DB 3,5,2
DSEG    ENDS                    ;数据段结束

SSEG    SEGMENT STACK           ;堆栈段开始
SKTOP DB 20 DUP(?)
SSEG    ENDS                    ;堆栈段结束

CSEG    SEGMENT                 ;代码段开始
        ASSUME CS:CSEG,DS:DSEG
        ASSUME SS:SSEG
START: MOV AX,DSEG              ;初始化数据段基址
        MOV DS,AX
        MOV AX,SSEG             ;初始化代码段基址
        MOV SS,AX
        MOV SP,LENGTH SKTOP     ;设置堆栈指针
        MOV SI,0                ;置偏移地址为0
AGAIN:RCL NUM[SI],1             ;带进位左移一位
```

```
        JNC NEXT1               ;若CF=0，跳到NEXT1
        INC SI                  ;得到下一个数
        RCL NUM[SI],1
        JNC NEXT2
        INC SI
        RCL NUM[SI],1
        JNC NEXT2               ;1负2负3正，到NEXT2
        MOV DL,2DH
        JMP FINISH

NEXT1:  INC SI
        RCL NUM[SI],1
        JC NEXT2                ;1正2负到NEXT2
        INC SI
        RCL NUM[SI],1
        JC NEXT2                ;1正2正3负到NEXT2
        MOV DL,2BH              ;1正2正3正，输出正号
        JMP FINISH

NEXT2:  MOV DL,20H
        JMP FINISH
```

```
FINISH:MOV AH,02H
       INT 21H
       MOV AH,4CH
       INT 21H          ;返回DOS
CSEG   ENDS             ;代码段结束
       END
```

七、 思考题（必需回答）写明如下问题

1. 按照操作顺序写出上机操作的步骤；
   答：①配置 EditPus
   ②使用集成开发环境编译连接
   ③运行
   ④使用 DEBUG 调试
2. 上机操作的步骤与顺序可以调整吗？为什么？
   答：不可以。
   ①首先得使用编辑程序完成汇编语言源程序的编辑，得到.asm 扩展名的文件；
   ②汇编程序将.asm 文件转化成二进制代码文件(.obj)
   ③经过汇编程序处理而产生的目标文件.obj 还不能直接运行，（一是目标文件中还是浮动地址，需要再定位；二是若为多模块程序，各个模块分别汇编后还需把它们连接起来。）
   ④在编写汇编语言源程序时产生的错误，除了一般语法上和格式上的错误可由汇编和连接程序法限制除外，逻辑上的错误必须用调试程序查找。调试程序 DEBUG 用于试验和检测用户程序。
3. 写出用 DEBUG 中的 E 命令修改内存单元内容的两种方法；
   答：-E<地址>　修改当前所给地址中的值
   -E<地址><内容表>　从当前偏移地址开始的内存单元修改为所指定的内容
4. 写出用 DEBUG 中的 A 命令修改指令的方法及应注意的问题；
   答：-A 从当前 CS:IP 处开始汇编
   -A［地址］ 指定输入汇编语言指令的位置
   注意：不能使用符号地址，地址必须放在方括号中
5. 总结 DEBUG 中 E、D、U 命令的功能；
   答：E　修改内存单元内容，默认为 DS 段
   D　显示内存单元的内容
   U　反汇编
6. 总结 DEBUG 中 P、G、T 命令的功能；
   答：P　执行汇编程序，单步跟踪

G 执行汇编指令，遇断点停止

T 从当前段地址开始执行指定的代码数

7. 写出数据定义伪指令 DB、DW 和 DD 存储整数的格式；

答：DB 一个字节存储一个整数，由低地址到高地址依次存储

DW 两个字节存储一个整数，数据高位字节在高地址，低位在低地址，由低向高依次存储

DD 四个字节存储一个整数，高位字节在高地址，低位字节在低地址，由低向高依次存储

8. 同一个正数分别用 DB、DW 和 DD 定义，存储格式有何变化？

答：存储空间大小不同，高地址存高位，低地址存低位，未占用的空间，高位补 0

9. 同一个负数分别用 DB、DW 和 DD 定义，存储格式有何变化？

答：存储空间大小不同，高地址存高位，低地址存低位，未占用的空间，高位补 F

10. 自定义标号分别用 DW 和 DD 定义，存储格式有何变化？

答：DW 存储的是偏移量，占用两个字节，DD 存储的是段寄存器和偏移量，占四个字节，高位存段寄存器，低位存偏移量

11. DB、DW 和 DD 在定义数据串时有何不同？

答：DB 数据占用一个字节

DW 数据占用两个字节

DD 数据占用四个字节

12. 说明十六进制数转换为 ASCII 码的方法；

答：将 16 进制数除以 16，商和余数分别存入不同的寄存器，将商和余数加上 30H

13. 字符串（例如“Microsoft”）使用什么数据定义伪指令？为什么？

答：DB。字符串里的符号、字母都是 ASCII 码中的，所有元素占用空间都是 1Byte

14. 顺序结构的程序能有多个出口吗？为什么？

答：不能。汇编程序都是 INT 21H 和 4CH 功能终止的，程序只有一个出口。

八、 结束语

实验教学与理论教学相辅相成，提高了我分析问题和解决问题的能力，加深了我对汇编语言程序的顺序结构和分支结构的理解，熟悉了编译、连接、运行和调试过程，获益匪浅。

九、 参考文献

1.《汇编语言程序设计》(第四版)，高福祥 齐志儒主编，东北大学出版社

2.《汇编语言实验教程》，张坤编著，清华大学出版社

3.《汇编语言及其应用教程》，李浪，熊江，齐忠主编，华中科技大学出版社

实验成绩

考查内容	分数	得分
做好实验内容的预习，写出预习报告	10	
了解实验题目的调试方法	10	
按实验要求预先设计好程序	10	
认真记录实验数据并分析实验结果	10	
实验后按要求书写实验报告，记录实验用数据及运行结果	30	
创新能力强，在实验中设计的程序有一定的通用性，算法优化	20	
实验过程中，具有严谨的学习态度，认真、踏实、一丝不苟的科学作风	10	