学生学号	0121610870910	实验课成绩	

武汉理工大学

学 生 实 验 报 告 书

课 程 名 称　汇编语言程序设计

开 课 学 院　计算机科学与技术学院

指 导 老 师　李宁

学 生 姓 名　冯钢果

专 业 班 级　软件 1604 班

2017 — 2018 学年 第 二 学期

实验课程名称：汇编语言程序设计

实验项目名称	汇编语言的调试和运行、汇编语言程序格式			实验成绩	
实验者	冯钢果	专业班级	软件 1604 班	组别	
同组者	（无）			实验日期	2018 年 4 月 24 日

第一部分：实验分析与设计（可加页）

一、实验内容描述

“汇编语言的调试和运行”实验内容：

1.1 键入 DEBUG 进入 DEBUG 控制状态,显示提示符 ’- ’。

1.2 用命令 F100 10F ’A’ 将’A’的 ASCII 码填入内存。

1.3 用命令 D100 10F 观察内存中的十六进制码及屏幕右边的 ASCII 字符。

1.4 用命令 F110 11F 41 重复上二项实验,观察结果并比较。

1.5 用命令 E100 30 31 32 …… 3F 将 30H-3FH 写入地址为 100 开始的内存单元中,再用 D 命令观察结果,看键入的十六进制数是什么字符的 ASCII 码?

1.6 用 DEBUG 调试和运行下列程序，记录所用的 DEBUG 命令和运行结果。

```
MOV WORD PTR [1100],3445
MOV WORD PTR [1102],5678
MOV WORD PTR [1106],6732
MOV AX,[1100]
SUB AX,[1102]
ADD AX,[1106]
MOV [1104],AX
HLT
```

其中粗斜体的数字可以修改。

1.7 内存操作数及各种寻址方式使用

程序内容：

```
MOV   AX,1234
MOV   [1000],AX
MOV   BX,1002
MOV   BYTE PTR[BX],20
MOV   DL,39
INC   BX
MOV   [BX],DL
DEC   DL
MOV   SI,3
MOV   [BX+SI],DL
MOV   [BX+SI+1],DL
MOV   WORD PTR[BX+SI+2],2846
```

“汇编语言程序格式”内容：

4.1 编一段程序，在内存中自 SQTAB（0200H）地址开始的连续 10 个单元中存放 0-9 的平方值。要求利用简单的查表法 NUM（0210）单元中指定数（0-9）的平方值，并将所求平方值存入 RESULT（0211）单元。

4.2 通过编辑、编译和连接、调试书中 119、146 页的例题熟悉掌握编译和连接程序的使用。

4.3 作书中作业 4.5　4.8　4.9　4.10　4.13　4.14 ，并编写小程序验证。

4.4 编写 4.17 程序，并上机运行。

二、 实验基本原理与设计

（方法或要求）

1.1 用 A 命令键入上述程序,并用 T 命令逐条运行。

1.2 每运行一条有关内存操作数的指令,要用 D 命令检查并记录有关内存单元的内容并注明是什么寻址方式。

4.1 程序要结构化:简明、易读和易调试;

4.2 执行速度较快。

4.3 占用存储空间较少。

三、 主要仪器设备及耗材

1. 一台电脑;
2. 软件 DosBox;
3. 参考材料：IBM-PC 汇编语言程序设计;
4. 笔和草稿纸。

第二部分：实验调试与结果分析（可加页）

一、 调试过程

1.1-1.7 分别按照实验步骤进行，并按照命令显示主要操作对象的值的变化。并分别记录每次实验的值。

思考题：BYTE PTR 及 WORD PTR 伪操作不能去掉，会报错。

4-1 代码如下：

```
DATAS SEGMENT
    SQTAB DB 0,1,4,9,16,25,36,49,64,81
DATAS ENDS
STACKS SEGMENT
STACKS ENDS
CODES SEGMENT
    ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
    MOV AX,DATAS
    MOV DS,AX
    MOV AH,1
    INT 21H
    AND AL,0FH
    LEA BX,SQTAB
    XLAT
    MOV BL,10
    MOV AH,0
    DIV BL
    MOV BH,AH
    OR AL,30H
    MOV DL,AL
    MOV AH,2
    INT 21H
    OR BH,30H
    MOV DL,BH
    MOV DL,BH
    MOV AH,2
    INT 21H
    MOV AH,4CH
    INT 21H
    CODES ENDS
END START
```

4-2 代码如下：

程序一：

```
data_seg1 segment
data_seg1 ends
data_seg2 segment
data_seg2
code_seg segment
```

```
    assume cs:code_seg,ds:data_seg1,es:data_seg2
start
    mov ax,data_seg1
    mov ds,ax
    mov ax,data_seg2
    mov es,ax
code_seg ends
    ends start
```

程序二：

```
data segment
    source_buffer db 40 dup('a')
data ends
extra segment
    dest_buffer db 40 dup(?)
extra ends
code segment
main proc far
    assume cs:code,ds:data,es:extra
start:
    push   ds
    sub ax,ax
    push   ax
    mov ax,data
    mov ds,ax
    mov ax,extra
    mov es,ax
    mov si,source_buffer
    lea di,dest_buffer
    cld
    mov cx,40
    rep movsb
    ret
main   endp
code   ends
    end start
```

4-3 键入书中代码；

4-4 代码如下：

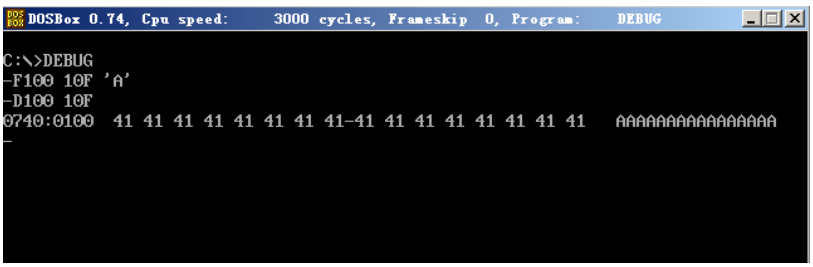
```
D_SEG SEGMENT
AUGW LABEL WORD
AUGEND DD 99251
SUM DD ?
D_SEG ENDS
E_SEG SEGMENT
ADDW LABEL WORD
ADDEND DD -15962
E_SEG ENDS
```

```
C_SEG SEGMENT
MAIN PROC FAR
ASSUME CS:C_SEG, DS:D_SEG, ES:E_SEG
START: PUSH DS
SUB AX, AX
PUSH AX
MOV AX, D_SEG
MOV DS, AX
MOV AX, E_SEG
MOV ES, AX
MOV AX, AUGW
MOV BX, AUGW+2
ADD AX, ES: ADDW
ADC BX, ES: ADDW+2
MOV WORD PTR SUM, AX
MOV WORD PTR [SUM+2], BX
RET
MAIN ENDP
C_SEG ENDS
END START
```

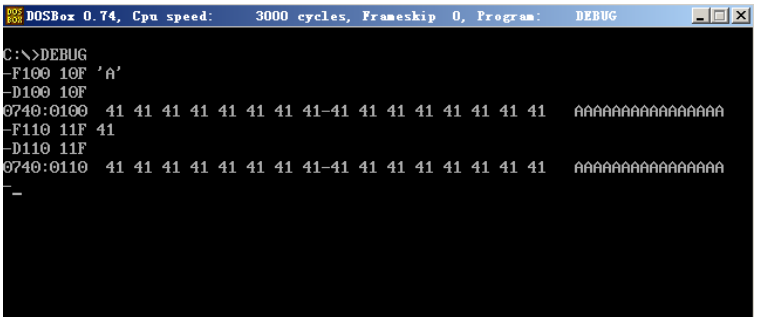
二、实验结果及分析（包括结果描述、实验现象分析、影响因素讨论、综合分析和结论等）

1.1 结果：

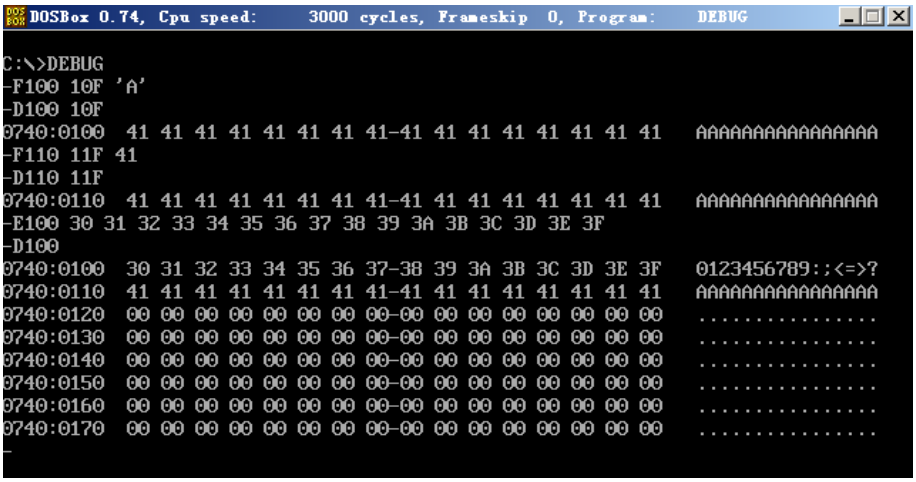
1.2 结果：



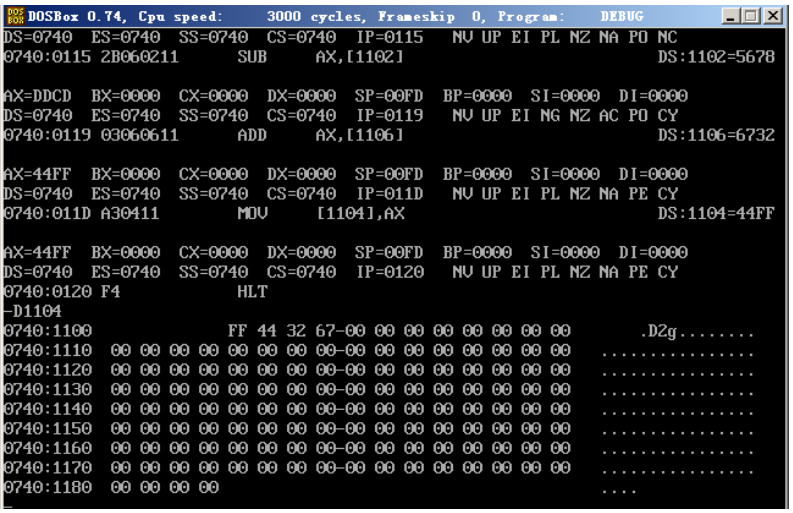
1.3-1.4 结果：



1.5 结果：

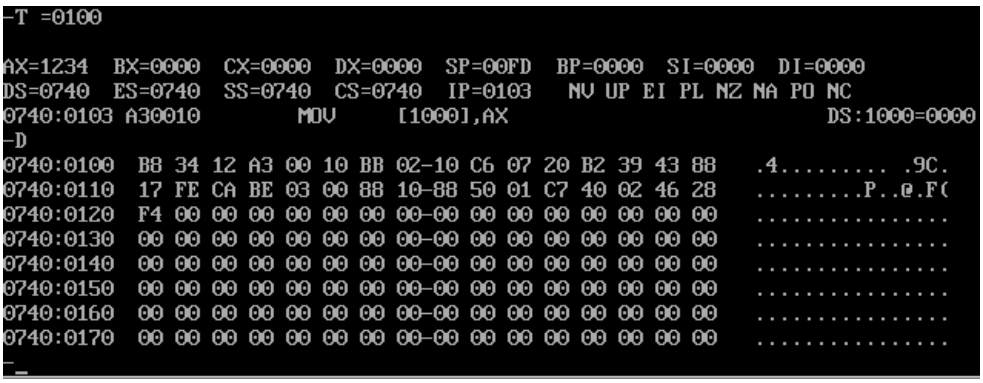


1.6 结果：



1.7 各步骤如下：

MOV AX,1234　　执行：立即寻址



MOV [1000],AX　　执行：直接寻址

MOV BX,1002　　执行：立即寻址

MOV BYTE PTR[BX],20　　执行：寄存器间接寻址

```
-D1002 100F
0740:1000       20 00 00 00 00 00-00 00 00 00 00 00 00 00      ..............
-
```

MOV DL,39　　执行：立即寻址

INC BX　　执行：寄存器寻址

```
-T

AX=1234  BX=1003  CX=0000  DX=0039  SP=00FD  BP=0000  SI=0000  DI=0000
DS=0740  ES=0740  SS=0740  CS=0740  IP=010F   NV UP EI PL NZ NA PE NC
0740:010F 8817          MOV     [BX],DL                        DS:1003=00
```

MOV [BX],DL　　执行：寄存器间接寻址

```
-T

AX=1234  BX=1003  CX=0000  DX=0039  SP=00FD  BP=0000  SI=0000  DI=0000
DS=0740  ES=0740  SS=0740  CS=0740  IP=0111   NV UP EI PL NZ NA PE NC
0740:0111 FECA          DEC     DL
```

DEC DL　　执行：寄存器寻址

```
-T

AX=1234  BX=1003  CX=0000  DX=0038  SP=00FD  BP=0000  SI=0000  DI=0000
DS=0740  ES=0740  SS=0740  CS=0740  IP=0113   NV UP EI PL NZ NA PO NC
0740:0113 BE0300        MOV     SI,0003
-
```

MOV SI,3　　执行：立即寻址

```
-T

AX=1234  BX=1003  CX=0000  DX=0038  SP=00FD  BP=0000  SI=0003  DI=0000
DS=0740  ES=0740  SS=0740  CS=0740  IP=0116   NV UP EI PL NZ NA PO NC
0740:0116 8810          MOV     [BX+SI],DL                     DS:1006=00
-
```

MOV [BX+SI],DL　　执行：基址变址寻址

```
-D1006
0740:1000                   38 00-00 00 00 00 00 00 00 00        8.........
0740:1010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0740:1020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0740:1030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0740:1040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0740:1050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0740:1060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0740:1070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0740:1080  00 00 00 00 00 00                                 ......
```

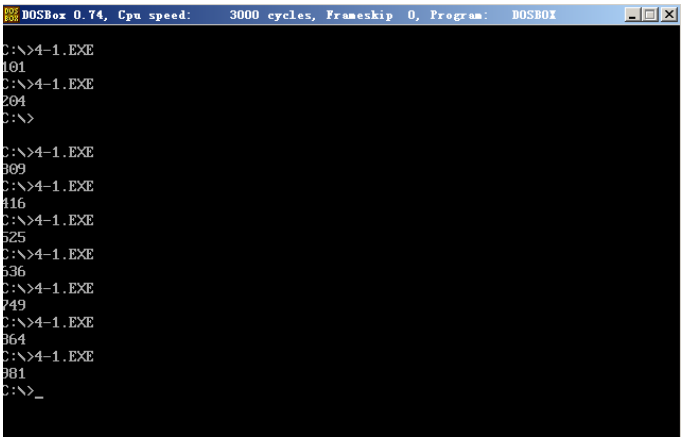
MOV [BX+SI+1],DL　　执行：基址变址寻址

```
-D1006
0740:1000                   38 38-00 00 00 00 00 00 00 00        88........
0740:1010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0740:1020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0740:1030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0740:1040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0740:1050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0740:1060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0740:1070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0740:1080  00 00 00 00 00 00                                 ......
-
```

MOV WORD PTR[BX+SI+2],2846　　执行：基址变址寻址

```
-D1006
0740:1000                   38 38-46 28 00 00 00 00 00 00        88F(......
0740:1010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0740:1020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0740:1030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0740:1040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0740:1050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0740:1060  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0740:1070  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0740:1080  00 00 00 00 00 00                                 ......
```

4-1 截图：



4-2 成功运行；

4-3 结果如下：

题 4-5 结果：

42H 59H 54H 45H 0DH EEH 00H - 01H 02H 01H 02H - 00H - 01H 02H 01H 02H - BYTE_VAR

00H 00H 01H 00H 02H 00H | | | -- FBH FFH 00H 59H 42H 45H 54H 56H 02H WORD_WAR 重复以上内容 4 次

题 4.8 结果

PLENTH=22=16H,它表示 PARTNO\PNAME\COUNT 总共占用的存储单元数。

题 4.9 结果

L=6

题 4.10 结果

（1）MOV AX,OFFSET LNAME

（2）MOV SL,WORD PTR CODE_LIST

（3）CODE_LENGTH EQU $-CODE_LIST

4-4 键入后正常运行。

三、实验小结、建议及体会

总结和体会：经过本次实验，我体会到了学习汇编注重实践，掌握编译和连接程序的使用，掌握汇编源程序的书写格式，熟悉了伪指令的用法，可以编写简单的源程序，用 DEBUG 运行基本指令和察看运行结果，理解了指令不同的寻址方式，在实践中学会了某些汇编语句的原理。

实验课程名称：汇编语言程序设计

实验项目名称	分支程序设计			实验成绩	
实验者	冯钢果	专业班级	软件 1604 班	组别	
同组者	（无）			实验日期	2018 年 5 月 1 日

第一部分：实验分析与设计（可加页）

一、实验内容描述

1.1 练习逻辑运算指令、比较指令和条件转移指令的功能、用法以及与标志位的关系和可用的寻址方式；

1.2 用地址表法，完成下面要求的多分支程序的设计。根据 MODE 单元中的模式字（0-7）分别转向 L0-L7 标号处执行。L0-L7 处分别完成显示‘0’-‘7’字符。

当 MODE=0 时，转 L0 标号，完成显示‘0’
当 MODE=1 时，转 L1 标号，完成显示‘1’
当 MODE=2 时，转 L2 标号，完成显示‘2’
当 MODE=3 时，转 L3 标号，完成显示‘3’
当 MODE=4 时，转 L4 标号，完成显示‘4’
当 MODE=5 时，转 L5 标号，完成显示‘5’
当 MODE=6 时，转 L6 标号，完成显示‘6’
当 MODE=7 时，转 L7 标号，完成显示‘7’

1.3 编制程序实现如下操作：

设有 10 个学生成绩，分别统计低于 60 分、60～69 分、70～79 分、80～89 分、90～99 分及 100 分的人数，并存放于 S5、S6、S7、S8、S9、S10 单元中。

程序清单：

```
;           统计学生成绩
DATA        SEGMENT
GRADE   DW 95H，60H，75H，92H，71H，86H，54H，89H，83H，76H
N           EQU    ($—GRADE)/2
ORG    30H
S5          DW0
S6              DW0
S7          DW0
S8          DW0
S9          DW0
S10         DW0
DATA        ENDS
;
STACK           SEGMENT    STACK
STA              DB 20 DUP (0)
  TOP            EQU $—STA
STACK         ENDS
;
CODE        SEGMENT
MAIN        PROC      FAR
```

```
            ASSUME   CS：CODE，  DS：DATA，  SS：STACK
START：      PUSH    DS
            SUB      AX，AX
                PUSH    AX
            MOV     AX，DATA
            MOV     DS，AX
            MOV     CX，N
            LEA      BX，GRADE              ；成绩表首地址
COMPARE：
            MOV      AX，[BX]
            CMP      AX，60H                ；<60？
            JL       FIVE
            CMP      AX，70H                ；<70?
            JL       SIX
            CMP      AX，80H                ；<80?
            JL       SEVEN
            CMP      AX，90H                ；<90?
            JL       EIGHT
            CMP      AX，100H               ；=100
            JNE      NINE
            INC      S10
            JMP      CHA
NINE：      INC      S9
            JMP      CHA
EIGHT：     INC      S8
            JMP      CHA
SEVEN：     INC      S7
            JMP      CHA
SIX：       INC      S6
            JMP      CHA
FIVE：      INC      S5
            JMP      CHA
CHA：       ADD      BX，2                  ；循环学生人数
            LOOP     COMPARE
            RET
MAIN        ENDP
CODE        ENDS
            END      START
```

执行程序后，将结果分别填入下列表中

N(总人数)	S5	S6	S7	S8	S9	S10

二、实验基本原理与设计

1.1 在 BLOCK 开始的内存单元中有若干以字节为单位的正、负数，自编程序，试统计其中正数的个数存放于 M_DATA 单元中，负数的个数存放于 P_DATA 单元中。

1.2 在 BLOCK 开始的内存单元中有若干以字节为单位的奇、偶数，自编程序，试统计其中偶数的个数存放于 M_DATA 单元中，奇数的个数存放于 P_DATA 单元中。部分程序清单

```
BLOCK       DB   37，—90，—32，60，—7，—120
   COUNT      EQU  $—BLOCK
   P_DATA      DB   COUNT  DCP (0)
   M_DATA     DB   COUNT  DCP (0)

    ……
          LEA   SI，BLOOK
          LEA    DI，P_DATA
          LEA    BX，M_DATA
          MOV   CX，COUNT
   RETRY：  MOV     AL，[SI]
          CMP   AL，0
          JGE      PP
          MOV   [BX]，AL
          INC      BX
          JMP      LOOP1
   PP：      MOV   [DI]，AL
          INC      DI
   LOOP1：  INC     SI
   LOOP    RETRY
          ……
```

要求：

1.1 分析问题，画出算法框图。写实验预习报告。

1.2 用汇编语言格式编写程序。

1.3 在 DOS 下编辑、汇编、连接程序。

1.4 用 Debug 调试程序。

1.5 在 DOS 下运行并分析结果。

1.6 写实验报告，画出算法框图。

三、**主要仪器设备及耗材**

1. 一台电脑；
2. 软件 DosBox；
3. 参考材料：IBM-PC 汇编语言程序设计；
4. 笔和草稿纸。

第二部分：实验调试与结果分析（可加页）

一、 **调试过程**（包括调试方法描述、实验数据记录，实验现象记录，实验过程发现的问题等）

内容 1：运行: AND AL,98H;AL = 0BDH

运行: OR AL,01001010B;AL=94H

运行：XOR AL,00001111B;AL=39H

内容 2 代码：

```
DATAS SEGMENT
CR EQU 0DH;回车
LF EQU 0AH;换行
C1 DB 'Please input MODE(0-7):$'
C2 DB CR,LF,'The result is:$'
C3 DB CR,LF,'Please input MODE(0-7):$'
Error DB '  ERROR!',CR,LF,'$'
QUE DB CR,LF,'Do you want to continue? Please input Y/N.$'
MODE DB ?
DATAS ENDS
STACKS SEGMENT
;此处输入堆栈段代码
STACKS ENDS
CODES SEGMENT
ASSUME CS:CODES,DS:DATAS,SS:STACKS
START:
MOV AX,DATAS
MOV DS,AX
ONE:MOV AH,09H
MOV DX,OFFSET C1
int 21H;显示输出字符串
MOV AH,01H;键入一个字符
int 21H
MOV MODE,AL
JMP THR
FOUR:MOV AH,09H
MOV DX,OFFSET C3
int 21H
MOV AH,01H;键入一个字符
int 21H
MOV MODE,AL
THR:CMP AL,'0'
JE L0
CMP AL,'1'
JE L1
CMP AL,'2'
JE L2
CMP AL,'3'
JE L3
```

```
    CMP AL,'4'
    JE L4
    CMP AL,'5'
    JE L5
    CMP AL,'6'
    JE L6
    CMP AL,'7'
    JE L7
    CMP AL,'0'
    JB ERR
    CMP AL,'7'
    JAE ERR
    ERR:MOV AH,09H
    MOV DX,OFFSET Error
    int 21H
    JMP ONE
    L0: MOV AH,09H
    MOV DX,OFFSET C2
    int 21H
    MOV DL,'0'
    JMP L
    L1: MOV AH,09H
    MOV DX,OFFSET C2
    int 21H
    MOV DL,'1'
    JMP L
    L2: MOV AH,09H
    MOV DX,OFFSET C2
    int 21H
    MOV DL,'2'
    JMP L
    L3: MOV AH,09H
    MOV DX,OFFSET C2
    int 21H
    MOV DL,'3'
    JMP L
    L4: MOV AH,09H
    MOV DX,OFFSET C2
    int 21H
    MOV DL,'4'
    JMP L
    L5: MOV AH,09H
    MOV DX,OFFSET C2
    int 21H
    MOV DL,'5'
    JMP L
    L6: MOV AH,09H
```

```
MOV DX,OFFSET C2
int 21H
MOV DL,'6'
JMP L
L7: MOV AH,09H
MOV DX,OFFSET C2
int 21H
MOV DL,'7'
JMP L
L:  MOV AH,02H
int 21H
MOV AH,09H
MOV DX,OFFSET QUE
int 21H
MOV AH,01H;键入一个字符
int 21H
CMP AL,'y'
JE FOUR
CMP AL,'Y'
JE FOUR
CMP AL,'N'
JE TWO
CMP AL,'n'
JE TWO
TWO:MOV AH,4CH
INT 21H
CODES ENDS
END START
```

内容 2 键入程序执行。

要求 1：实验代码：

```
DATA  SEGMENT
ORG 1000H
BLOCK  DB 37,-90,-32,60,-7,-120
COUNT  EQU $-BLOCK
ORG 2000H
P_DATA  DB COUNT  DUP (0)
ORG 3000H
M_DATA  DB COUNT  DUP (0)
DATA  ENDS
CODE  SEGMENT
ASSUME CS:CODE,DS:DATA
START:
MOV AX,DATA
MOV DS,AX
LEA SI,BLOCK
LEA DI,P_DATA
LEA BX,M_DATA
```

```
MOV CX,COUNT
NEXT:
MOV AL,[SI]
CMP AL,00H
JGE  L1
MOV [BX],AL
INC BX
JMP  LOOP1
L1: MOV [DI],AL
INC DI
LOOP1: INC SI
LOOP  NEXT
MOV AH,4CH
INT 21H
CODE  ENDS
END START
```

要求 2：实验代码：

```
DATA  SEGMENT
ORG 1000H
BLOCK  DB 3,7,2,6,17,0
COUNT  EQU $-BLOCK
ORG 2000H
P_DATA  DB COUNT  DUP (0)
ORG 3000H
M_DATA  DB COUNT  DUP (0)
DATA  ENDS
CODE  SEGMENT
ASSUME CS:CODE,DS:DATA
START:
MOV AX,DATA
MOV DS,AX
LEA SI,BLOCK
LEA DI,M_DATA
LEA BX,P_DATA
MOV CX,COUNT
NEXT:
MOV AL,[SI]
TEST  AL,01H
JZ  L1
MOV [BX],AL
INC BX
JMP  LOOP1
L1: MOV [DI],AL
INC DI
LOOP1: INC SI
LOOP  NEXT
MOV AH,4CH
```

```
INT 21H
CODE  ENDS
END START
```

二、实验结果及分析（包括结果描述、实验现象分析、影响因素讨论、综合分析和结论等）

内容 1 运行结果：(AL) = 98H

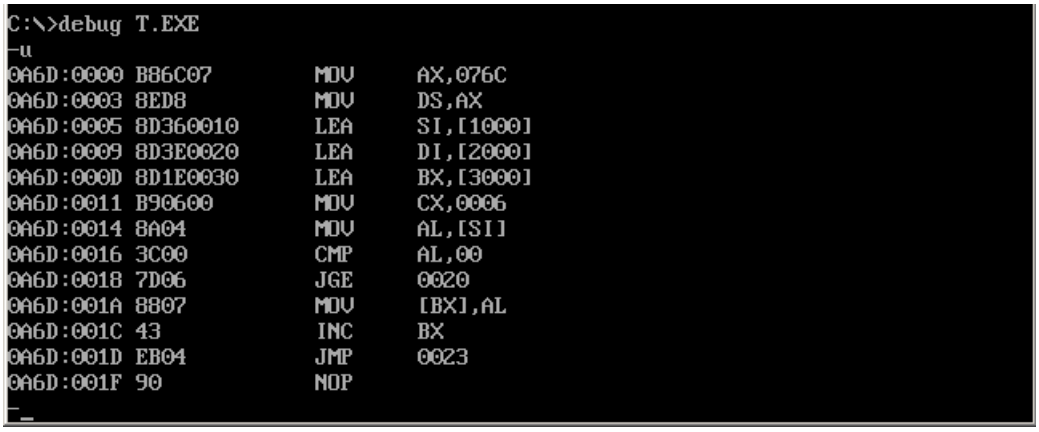
(AL) = 0DEH

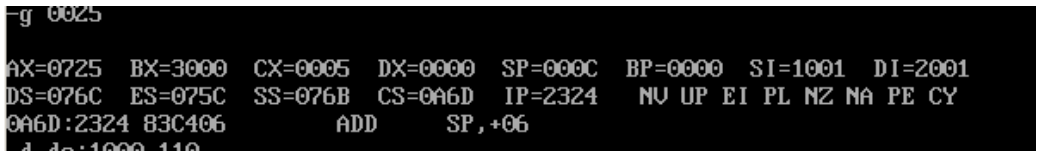
(AL) = 36H

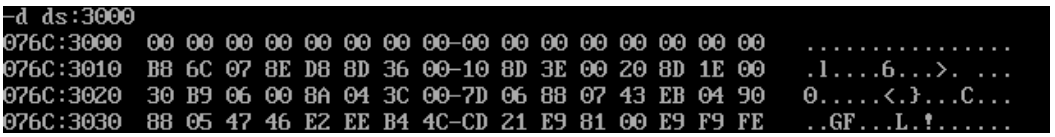
内容 2 结果：

N(总人数)	S5	S6	S7	S8	S9	S10
10	1	1	3	3	2	0

要求 1 实验截图：







要求 2 结果截图：

```
C:\>DEBUG.EXE T.EXE
-u
0A6D:0000 B86C07        MOV     AX,076C
0A6D:0003 8ED8          MOV     DS,AX
0A6D:0005 8D360010      LEA     SI,[1000]
0A6D:0009 8D3E0030      LEA     DI,[3000]
0A6D:000D 8D1E0020      LEA     BX,[2000]
0A6D:0011 B90600        MOV     CX,0006
0A6D:0014 8A04          MOV     AL,[SI]
0A6D:0016 A801          TEST    AL,01
0A6D:0018 7406          JZ      0020
0A6D:001A 8807          MOV     [BX],AL
0A6D:001C 43            INC     BX
0A6D:001D EB04          JMP     0023
0A6D:001F 90            NOP
-_
```

```
-g 0025

AX=0703  BX=2001  CX=0005  DX=0000  SP=0000  BP=0000  SI=1001  DI=3000
DS=076C  ES=075C  SS=076B  CS=0A6D  IP=2324   NV UP EI NG NZ NA PO NC
0A6D:2324 83C406        ADD     SP,+06
-d ds:1000
076C:1000  03 07 02 06 11 00 00 00-00 00 00 00 00 00 00 00   ................
076C:1010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076C:1020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
```

```
-d ds:2000
076C:2000  03 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076C:2010  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076C:2020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076C:2030  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076C:2040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076C:2050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
```

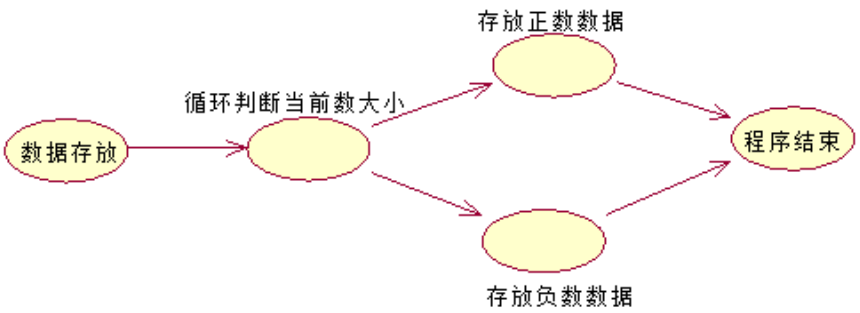
```
-d ds:3000
076C:3000  00 00 98 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
076C:3010  B8 6C 07 8E D8 8D 36 00-10 8D 3E 00 30 8D 1E 00   .l....6...>.0...
076C:3020  20 B9 06 00 8A 04 A8 01-74 06 88 07 43 EB 04 90    .......t...C...
076C:3030  88 05 47 46 E2 EE B4 4C-CD 21 E9 81 00 E9 F9 FE   ..GF...L.!......
076C:3040  E8 B7 EF EB EE 90 E8 49-F2 EB E8 90 E8 CB FB EB   .......I........
076C:3050  E2 90 E8 7D EC EB DC 90-E8 15 F7 EB D6 90 E8 51   ...}...........Q
076C:3060  FA EB D0 90 E8 39 F1 EB-CA 90 E8 79 F8 EB C4 90   .....9.....y....
```

分析结果：

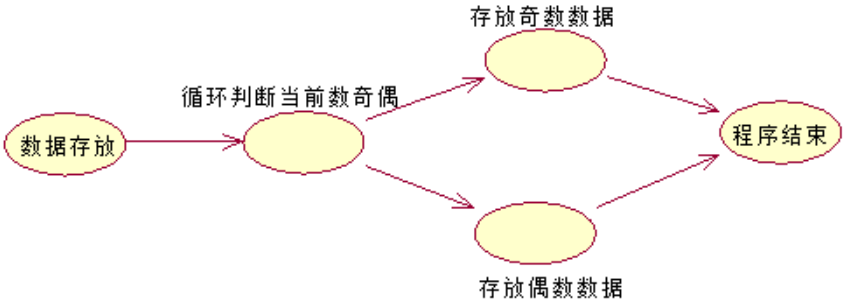
利用比较与转移指令实现分支，分别存放至数据区域，调试过程先 u 反汇编，使用 g 指令进行运行到终止地址处，使用 d：ds=地址地址 地址长度显示结果，且实验结果正确。

算法框图：

要求一框图：



要求二框图：



三、实验小结、建议及体会

总结和体会：经过本次实验，我体会到了学习汇编注重实践，只有不断实践，才能发现更多的问题，更加深刻的理解问题。在本次实验中，掌握了逻辑运算和转移指令，学会使用逻辑运算指令、标号和无条件转移指令、比较指令和条件转移指令，掌握了各种分支程序中形成和判断条件而产生的程序段的设计方法和技巧。只有不断实践才能锻炼自己的编程能力，有更深刻的记忆和认识。

实验课程名称：汇编语言程序设计

实验项目名称	循环程序设计			实验成绩	
实验者	冯钢果	专业班级	软件 1604 班	组别	
同组者	（无）			实验日期	2018 年 5 月 8 日

第一部分：实验分析与设计（可加页）

一、实验内容描述

1. 请编写一程序，从附加段中一个未排序的字数组中，找出最大数和最小数分别存放在 AX 和 BX 寄存器中。
2. 以 GRADE 为首地址的 10 个字的数组中保存有学生成绩。建立一个 10 个字的 RNAK 数组，并根据 GRADE 中的学生成绩将学生名次填入 RANK 数组中（提示：一个学生的名次等于成绩高于等于该学生的人数加 1）。

寄存器分配情况说明如下：

AX 存放当前被测学生的成绩
BX 存放当前被测学生的相对地址指针
CX 内循环计数值
DX 存放当前被测学生的名次计数值
SI 内循环测试时的地址指针
DI 外循环计数值

部分程序清单（将程序补充完整）

```
;                建立学生成绩名次表
DATA        SEGMENT
GRADE    DW  88H，75H，95H，63H，98H，78H，87H，73H，90H，60H
COUNT    EQU  ($-GRADE) /2
             ORG 20H
RANK     DW 10 DUP(?)
DATA     ENDS
;
STACK    SEGMENT  STACK
STA          DB 20 DUP (20H)
TOP          EQU  $—STA
STACK        ENDS
;
CODE     SEGMENT
         ASSUME  CS：CODE，DS：DATA，SS：STACK
START：  MOV  AX，DATA
         MOV  DS，AX
         MOV  AX，STACK
         MOV  SS，AX
         MOV  SP，TOP

         MOV  DI，_____    ；成绩的个数
         MOV  BX，0
```

```
        LOOP1：MOV   AX，GRADE[BX]，1
               MOV   CX，COUNT
               _____ SI，GRADE      ；成绩的存放地址
        NEXT：     CMP   AX，[SI]
               _____ NO_COUNT
               INC    WORD PTR RANK[BX]；存放学生名次
  NO_COUNT：ADD    SI，2
               ______ NEXT
               ADD    BX，2
               DEC    DI
               _____ LOOP1
               MOV   AX，4C00H
               INT     21H
     CODE      ENDS
               END    START
```

3、一个数组 DATAX，其中的数据排列规律是：头三项是 0，0，1，以每项的值均是前三项之和。试将项值小于等于 2000 以前的各项数据填入数组 DATAX 中。

算法： n≥ 4 时： ① $a_n=a_{n-1}+a_{n-2}+a_{n-3}$

② $a_n=2a_{n-1}-a_{n-4}$

要求：按上述两种算法编程

4、从 DATA_BUF(1000H)开始存放 50 个字节数据，编写程序将这些数据由小到大排序，排序后的数据仍放在该区域中。

要求原始数据在源程序中给出,排序前后的数据以每行 10 个的格式显示在屏幕上。

实验提示：

参见教材第 134 页，例 5.7 所用的数据如下：

```
stack    segment para stack 'stack'

           db        1024 dup (?)
 stack    ends
 data     segment para 'data'

 data_buf  db  50,49,48,47,46,45,44,43,42,41
           db  40,39,38,37,36,35,34,33,32,31
           db  30,29,28,27,26,25,24,23,22,21
           db  20,19,18,17,16,15,14,13,12,11
           db  10, 9, 8, 7, 6, 5, 4, 3, 2, 1
 msg1      db  'Raw Data:', 0dh, 0ah, '$'
 msg2      db  'Sorted Data:', 0dh, 0ah, '$'
data      ends
```

5、测量一字符串长度,并用十六进制数显示之

程序内容:

```
JMP START
DB 'This is a program to measure the length of a string。'
```

```
        DB  0D,0A
        DB  'the length of the string is: $'
START:  MOV   AH,9
        MOV   DX,102
        INT   21
        MOV   BX,101
        MOV   AL,24
        MOV   DH,FF
LOP:    INC   BX
        INC   DH
        CMP   AL,[BX]
        JNZ   LOP
        MOV   CH,2
        MOV   CL,4
LOP1:   ROL   DH,CL
        MOV   DL,DH
        AND   DL,0F
        ADD   DL,30
        CMP   DL,3A
          JC  J
          ADD DL,7
J:        MOV AH,2
          INT 21
          DEC CH
          JNZ LOP1
          INT 20
```

键入此程序,存入DD.COM文件中并在DOS命令状态下直接运行之,若未出现预期结果,用DEBUG检查有否错误?

二、实验基本原理与设计

1、分析问题，画出算法框图。写实验预习报告。
2、用汇编语言格式编写程序。
3、在 DOS 下编辑、汇编、连接程序。
4、用 Debug 调试程序。
5、在 DOS 下运行并分析结果。
6、写实验报告，画出算法框图。

三、主要仪器设备及耗材

1. 一台电脑；
2. 软件 DosBox；
3. 参考材料：IBM-PC 汇编语言程序设计；
4. 笔和草稿纸。

第二部分：实验调试与结果分析（可加页）

一、 **调试过程**（包括调试方法描述、实验数据记录，实验现象记录，实验过程发现的问题等）

1.题目 1 代码如下：

```
ESEG    SEGMENT
        UNORDLST DW 50 DUP(?)
        COUNT   EQU ($-UNORDLST)/2
ESEG    ENDS
CSEG SEGMENT PARA'code'
 ASSUME CS:CSEG,ES:ESEG
MINMAX PROC FAR
 PUSH   DS
 SUB AX,AX
 PUSH   AX
 MOV AX,ESEG
 MOV ES,AX
 MOV CX,COUNT
 DEC CX
 LEA DI,UNORDLST
 MOV AX,BX
CHEMIN:
 ADD DI,2
 CMP ES:[DI],BX
 JAE CHKMAX
 MOV BX,ES:[DI]
 JMP SHORT NEXTEL
CHKMAX:
 CMP ES:[DI],AX
 JBE NEXTEL
 MOV AX,ES:[DI]
NEXTEL:
 LOOP   CHEMIN
 RET
 MINMAX ENDP
CSEG ENDS
 END MINMAX
```

2.实验 2 代码如下：

```
DATA        SEGMENT
 GRADE      DW  88H, 75H, 95H, 63H, 98H, 78H, 87H, 73H, 90H, 60H
 COUNT      EQU  ($-GRADE) /2
                ORG 20H
 RANK       DW 10 DUP(?)
 DATA       ENDS
 ;
 STACK      SEGMENT  STACK
```

```
STA         DB 20 DUP (20H)
TOP         EQU  $—STA
STACK          ENDS
;
CODE        SEGMENT
          ASSUME  CS：CODE，DS：DATA，SS：STACK
  START：  MOV  AX，DATA
           MOV  DS，AX
           MOV  AX，STACK
           MOV  SS，AX
           MOV  SP，TOP

           MOV  DI，    10          ；成绩的个数
           MOV  BX，0
  LOOP1：  MOV  AX，GRADE[BX]，1
           MOV  CX，COUNT
             LEA     SI，GRADE    ；成绩的存放地址
  NEXT：      CMP  AX，[SI]
              JG    NO_COUNT
           INC   WORD PTR RANK[BX]；存放学生名次
NO_COUNT：ADD   SI，2
            LOOP       NEXT
           ADD   BX，2
           DEC   DI
            LOOP      LOOP1
           MOV  AX，4C00H
           INT    21H
CODE       ENDS
           END START
```

3.实验代码如下：

```
DATA  SEGMENT
DATAX  DW  100  DUP（0）
DATA  ENDS
S  SEGMENT  STACK
DW  50  DUP（0）
S  ENDS
CODE  SEGMENT
ASSUME  DS：DATA，  SS：S，CS：CODE
MAIN  PROC  FAR
PUSH  DS
XOR  AX，AX
PUSH  AX
MOV  AX，DATA
MOV  DS，AX
MOV  BX，OFFSET  DATAX
MOV  WORD  PTR[BX+4]，1
LEA  BX，DATAX+6
```

```
AGAIN: MOV  AX, [BX-2]
ADD  AX, [BX-4]
ADD  AX, [BX-6]
CMP  AX, 2000
JA  A
MOV  [BX], AX
ADD  BX, 2
JMP  AGAIN
A:  RET
MAIN  ENDP
CODE  ENDS
END  MAIN
```

4.实验关键代码如下：

```
;排序算法
mov cx,50
 loop01:
 MOV DI CX
 MOV BX,0
 LOOP02:
 MOV AL,DATA_BUF[BX]
 CMP AL,DATA_BUF[BX+1]
 JLE CONTINUE
 XCHG AL,DATA_BUF[BX+1]
 MOV DATA_BUF[BX],AL
 CONTINUE:
 ADD BX,1
 LOOP LOOP02
 MOV CX,DI
 LOP LOOP01
```

5.键入全部代码后，存为 DD.COM 代码，然后运行出错，使用 debug 调试。

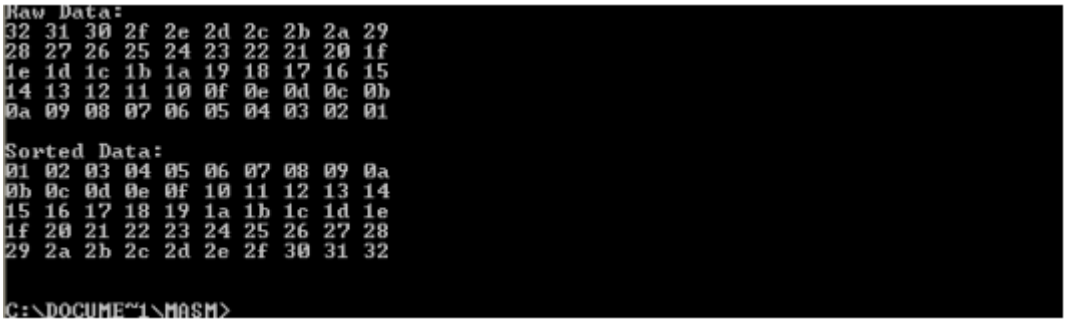
二、实验结果及分析（包括结果描述、实验现象分析、影响因素讨论、综合分析和结论等）

1.实验 1 运行前，未排序数组经过冒泡排序后，将最大最小值分别存入 AX、BX，如未排序数组为：3 1 7 5 9；运行后，AX=1，BX=9。

2.实验 2 也是冒泡排序，运行后 RANK 数组内容变为：98H，95H，90H，88H，87H，78H，75H，73H，63H，60H，符合题目要求。

3.程序类似与斐波那契数列，可使用循环，按照公式填入即可。最后判断填入结果不大于 2000 即可。

4.可以作为一个简单的排序算法，主要算法是排序，可以采用冒泡排序，然后每排进行排序即可。程序运行截图如下：

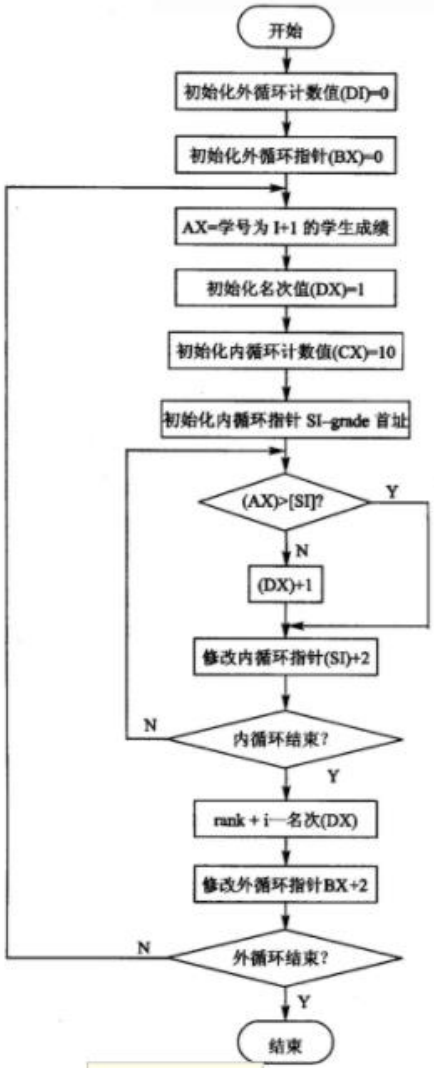


5. 程序调试后运行截图如下：

```
Please Input your data:
2154
BCD Code:1000 0101 0011 0010
Press Enter key to continue...
```

下面是本次实验的流程图：

实验 2 流程图：



三、实验小结、建议及体会

总结和体会：经过本次实验，我体会到了学习汇编注重实践，掌握编译和连接程序的使用，掌握汇编源程序的书写格式，掌握了循环程序设计的方法，学会正确分配与使用寄存器和控制循环的方法。最后深刻体会到汇编语言的趣味，对于算法的编写汇编语言仍有一些难度，需要克服这些问题，才能更深刻理解更高级的编程语言和编程技巧。

实验课程名称：汇编语言程序设计

实验项目名称	子程序结构			实验成绩	
实验者	冯钢果	专业班级	软件 1604 班	组别	
同组者	（无）			实验日期	2018 年 5 月 16 日

第一部分：实验分析与设计（可加页）

一、实验内容描述

1、将主程序中 BX 寄存器内的二进制数用十六进制数的形式在屏幕上显示出来。

实验参考程序：

```
code   segment
main   proc far
   assume cs:code
start: push ds
   xor ax,ax
   push ax
   mov   bx,1234h
call   sub1
   ret
main   endp
sub1   proc near
   mov ch,4
rotate:   mov cl,4
   rol bx, cl
   mov al,bl
   and al,0fh
   add al,30h
   cmp al,3ah
   jl printit
   add al,7h
printit: mov dl,al
   mov ah,2
   int 21h
   dec ch
   jnz rotate
   ret
sub1   endp
code   ends
      end
```

要求：

1)分析问题，画出算法框图。写实验预习报告。

2)用汇编语言格式编写程序。

3)在上题的基础，将 MEM 中的 4 个字节内容输出到屏幕显示。

4)在 DOS 下编辑、汇编、连接程序。

5)在 DOS 下运行并分析结果。
6)写实验报告，画出算法框图。
2、编制一程序，要求键入一个班的学生成绩，并存放于 50 字的 ERADE 数组中，然后根据 ERADE 中的成绩，把学生名次填入 50 字的 RANK 数组中，再按学号顺序把名次从终端上显示出来。
提示：
1)程序 MAIN
功能：根据输入的学生成绩，计算并显示出学生名次。
2)程序 INPUT
功能：接收一个班级学生的成绩，各成绩之间用空格隔开。
3)程序 RANKP
功能：计算一个班级学生的名次。(可参照循环程序(一)中的程序段)
4)程序 OUTPUT
功能：输出（显示）一个班级的学生名次
5)程序 DECIBIN
功能：十进制转换二进制，存入 BX
6)程序 BINDEC
功能：十进制转换二进制，并在屏幕上显示。
7)程序 DEC_DIV
功能：BX 的内容除以 CX 的内容，并在屏幕上显示一位商。
部分程序清单：
1） 键入学生成绩 (成绩之间用空格间隔，回车结束输入)

```
INPUT   PROC
        MOV SI,  0
        MOV     COUNT,  0
ENTER:  CALL    DECIBIN
        INC     COUNT
        CMP  DL,‘  ’
        JZ      STORE
        CMP  DL, 0DH
        JZ      EXIT
        JMP     RET1
STORE:  MOV   GRADE [SI], BX
        ADD     SI, 2
        JMP     ENTER
EXIT:  MOV    GRADE [SI],  BX
RET1:  RET
INPUT  ENDP
```

2）十进制转换成二进制

```
DECIBIN    PROC
        MOV  BX,  0
NEM:    MOV  AH,  1
           INT  21H
           MOV   DL,  AL
           CMP   AL,  30H
           JL    EXIT1
```

```
            CMP    AL,  39H
            JG     EXIT
            SUB    AL,  30H
            CBW
            XCHG   AX,  BX
            MOV    CX,  10
            MUL    CL
            XCHG   AX,  BX
            ADD    BX,  AX
            JMP    NEW
 EXIT1:  RET
 DECIBIN   ENDP
```

3）二进制转换成十进制

```
BINIDEC    PROC
               PUSH    BX
               PUSH    CX
               PUSH    SI
               PUSH    DI
               MOV     CX, 100
               CALL    DEC_DIV
               MOV     CX, 10
               CALL    DEC_DIV
               MOV     CX, 1
               CALL    DEC_DIV
               POP     DI
                 POP       SI
               POP CX
             POP       BX
             RET
BINIDEC      ENDP
```

4）十进制转换成 ASCII 码，并输出

```
DEC_DIV     PROC
              MOV  AX,  BX
              MOV  DX,  0
              DIV  CX
              MOV  BX,  DX
              MOV  DL,  AL
              ADD  DL,  30H
              MOV  AH,  02H
              INT  21H
              RET
 DEC_DIV    ENDP
```

二、实验基本原理与设计

实验要求：

1）自编程序　主程序 MAIN，子程序 OUTPUT 和子程序 RANKP。

2）将上述程序与自编程序统调。
3）输入本班级某门基础课成绩。

思考题
1）写出 4 位 BCD 码转二进制数的算法。
2）写出 AX 中进二制数转 BCD 码的算法。
3）将上述子程序结构改为模块化程序设计。

三、**主要仪器设备及耗材**

1. 一台电脑；
2. 软件 DosBox；
3. 参考材料：IBM-PC 汇编语言程序设计；
4. 笔和草稿纸。

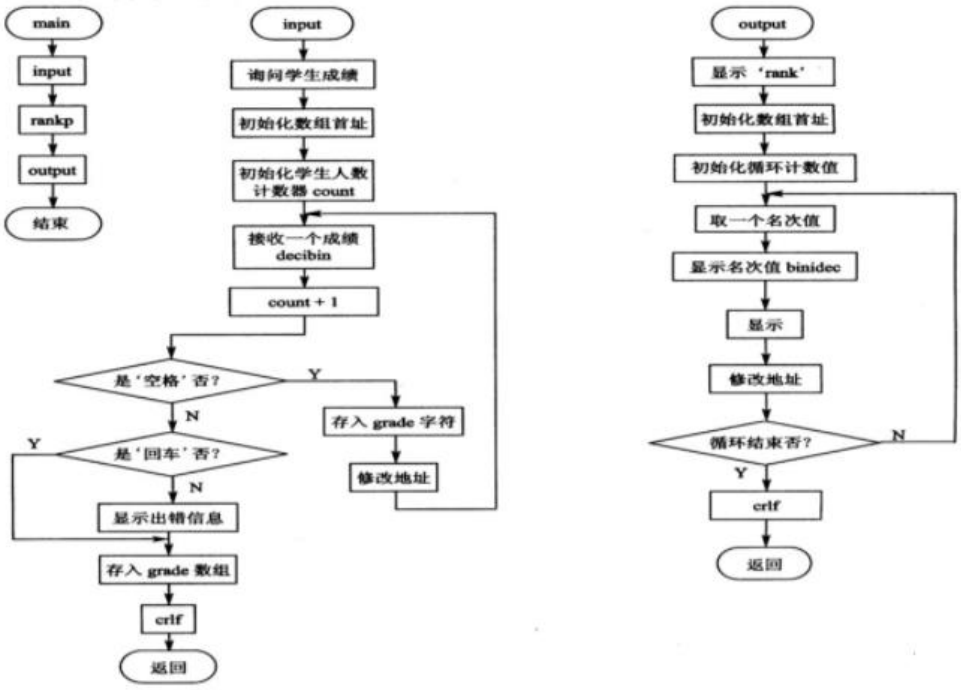
第二部分：实验调试与结果分析（可加页）

一、　　调试过程

1.如使用 MOV BX,1234H 将 BX 赋值为 1001000110100，然后再运行如下代码：

```
code segment
 assume cs:code
start:
 mov bx,1234H
 sub ax,ax
 mov ch,4
rotate:
 mov cl,4
 rol bx,cl
 mov al,bl
 and al,0fH
 add al,30H
 cmp al,3aH
 jl  print
 add al,7H
print:
 mov dl,al
 mov ah,2
 int 21H
 dec ch
 jnz rotate
 mov dl,'H'
 mov ah,2
 int 21h
 mov ah,4ch
 int 21h
code ends
 end start
```

2.程序框图如下：



代码如下：

```
DATA SEGMENT
     GRADE    DW   50   DUP()
     RANK     DW   50   DUP()
     ENTER    DB &#39
     ERROR    DB   13,10,&#39
     SHOW     DB   &#39
     COUNT    DW   0
     CR   DB   13,10,24H
     DATA     ENDS
STACK    SEGMENT
     DB   400 DUP(0)
     STACK    ENDS
CODE     SEGMENT
     ASSUME   CS:CODE,DS:DATA,SS:STACK
     START:
     MOV  AX,DATA
     MOV  DS,AX
     MOV  ES,AX
     CALL     INPUT
     CALL     RANKP
     CALL     OUTPUT
     MOV  AH,01H
     INT  21H
     MOV  AX,4CH
     INT  21H
     INPUT    PROC     NEAR
          LEA DX,ENTER
          MOV  AH,9
          INT  21H
```

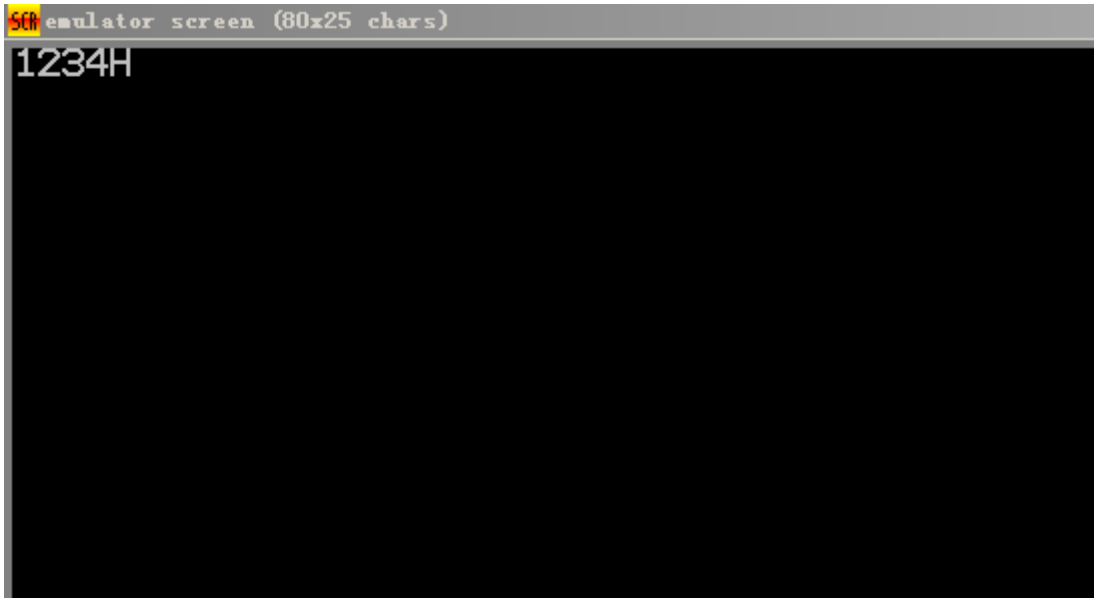
```
        MOV SI,0
        MOV COUNT,0
        _ENTER:
        INC COUNT
        CMP DL,&#39
        CMP DL,0DH
        JNE _ERROR
        MOV GRADE[SI],BX
        ADD SI,2
        JMP _ENTER
        _ERROR:
        LEA DX,ERROR
        MOV AH,9
        INT 21H
        _ENDENTER:
        MOV GRADE[SI],BX
        MOV DX,OFFSET   CR
        MOV AH,9
        INT 21H
        RET
        INPUT   ENDP
    RANKP   PROC    NEAR
        MOV DI,COUNT
        MOV BX,0
        _LOOP:
        MOV AX,GRADE[BX]
        MOV WORD    PTR RANK[BX],0
        LEA SI,GRADE
        _NEXT1:
        CMP AX,[SI]
        JG  _JUMP
        _JUMP:
        ADD SI,2
        LOOP    _NEXT1
        ADD BX,2
        DEC DI
        RET
        RANKP   ENDP
    OUTPUT  PROC    NEAR
        LEA DX,SHOW
        MOV AH,09H
        INT 21H
        MOV SI,0
        MOV DI,COUNT
        _NEXT2:
        MOV BX,RANK[SI]
        CALL    BINIDEC
```

```
        CMP DI,1
        MOV DL,&#39
        MOV AH,02H
        INT 21H
        _NOCOMMA:
        ADD SI,02H
        JNZ _NEXT2
        MOV DX,OFFSET   CR
        MOV AH,9
        INT 21H
        RET
        OUTPUT  ENDP
    DECIBIN PROC    NEAR
        MOV BX,0
        _CATCH:
        MOV AH,01H
        INT 21H
        MOV DL,AL
        SUB AL,30H
        JL  _ENDCHANGE
        CMP AL,39H
        JG  _ENDCHANGE
        CBW
        XCHG    AX,BX
        MOV CX,10
        MUL CX
        XCHG    AX,BX
        ADD BX,BX
        JMP _CATCH
        _ENDCHANGE:
        RET
        DECIBIN  ENDP
    BINIDEC PROC    NEAR
        PUSH    BX
        PUSH    CX
        PUSH    SI
        PUSH    DI
        MOV CX,10
        CALL    DEC_DIV
        MOV CX,1
        CALL    DEC_DIV
        POP DI
        POP SI
        POP CX
        POP BX
        RET
        BINIDEC ENDP
```

```
DEC_DIV PROC    NEAR
    MOV AX,BX
    MOV DX,0
    DIV CX
    MOV BX,DX
    MOV DL,AL
    ADD DL,30H
    MOV AH,02H
    INT 21H
    RET
    DEC_DIV ENDP
CODE    ENDS
END START
```

二、实验结果及分析（包括结果描述、实验现象分析、影响因素讨论、综合分析和结论等）

1.实验结果截图：（由于输入即是 1234H 的二进制等价，所以结果相同）



2.思考题：

（1）写出 4 位 BCD 码转二进制数的算法：

```
BCDTO2  PROC    NEAR
    PUSH    BX
    PUSH    CX
    PUSH    DX
    MOV BX,AX
    MOV AX,0
    MOV CX,4
    RETRY:
    PUSH    CX
    MOV CL,4
    ROL BX,CL
    MUL W10
    PUSH    BX
    AND BX,000FH
    ADD AX,BX
```

```
    POP BX
        LOOP    RETRY
        POP DX
        POP CX
        POP BX
        RET
        BCDTO2  ENDP
```

（2）写出 AX 中进二制数转 BCD 码的算法：

```
W1000   DW  1000,100,10,1
AX2TOBCD    PROC    NEAR
        XOR BX,BX
        MOV SI,OFFSET   W1000
        MOV CX,4
        RETRY:
        PUSH    CX
        MOV CL,4
        SEL BX,CL
        MOV DX,0
        DIV WORD    PTR[SI]
        OR  BX,AX
        MOV AX,DX
        POP CX
        ADD SI,2
        LOOP    RETRY
        MOV AX,BX
        AX2TOBCD    ENDP
```

三、实验小结、建议及体会

实验总结和体会：本次实验加强了我的实践操作能力，有些看似简单的理论知识，搬到实践运用上就显得不那么简单，往往容易犯低级错误，我觉得这次实验很有必要，它不仅加强了我的实践能力，更进一步帮助我理解汇编语言平时上课不能理解的地方，丰富了课堂教学内容，汇编语言并不是那么枯燥，认真学习还是能体会到语言的魅力的。

另外，通过本次试验我掌握了程序设计方法和合理划分层次，了解了子程序的调用与返回的方法，学会了子程序的嵌套与递归。