

东南大学

《微机系统与接口实验》

实验报告

实验三 “简单”编程练习 (预习报告)

姓 名：薛宇飞

学 号：04020235

同 组：

学 号：

专 业：信息工程

实 验 室：金智楼硬件实验室

实验时间：2022 年 5 月 12 日

报告时间：2022 年 5 月 12 日

评定成绩：

评阅教师：裴文江

目录

1	实验目的与内容	3
2	实验预习任务	3
3	实验任务	5
4	实验程序及分析	5
1	实验一	5
2	实验二	7
3	实验三	9
4	实验四	10
5	实验五	13
6	实验六	16
7	实验七	18
	参考文献	20

一. 实验目的与内容

1. 结合实验教材^[1-2]，利用已掌握的宏汇编语言，进行简单的程序设计练习；
2. 学习和掌握建立与运行汇编语言源程序各个步骤的命令；
3. 熟悉汇编程序的调试过程。

二. 实验预习任务

在一个有正、负数的数据块中，找出负数的个数，假设有数据-19, 28, 37, -46, 55, 61, -74, 数据块的长度存放在 CX 寄存器中，负数的个数存放在以 SUM 为符号的单元中。实验代码如下：

exp30.asm

```
1      DATA    SEGMENT
2      NUM      DB  -19,28,27,-46,55,61,-74
3      SUM      DB  ?
4      DATA    ENDS
5
6      MAIN     SEGMENT
7      ASSUME   CS:MAIN,DS:DATA
8      START:   MOV    AX,DATA
9              MOV    DS,AX
10             MOV    CX,7
11             MOV    AL,00
12             LEA    SI,NUM
13      AGAIN:   MOV    BL,[SI]
14             CMP    BL,00
15             JGE    NEXT
16             INC    AL
17      NEXT:    INC    SI
18             LOOP   AGAIN
19             MOV    SUM,AL
20             MOV    AH,4CH
21             INT    21H
22      MAIN     ENDS
23      END      START
```

实验流程如图 1：

实验结果如图 2所示：

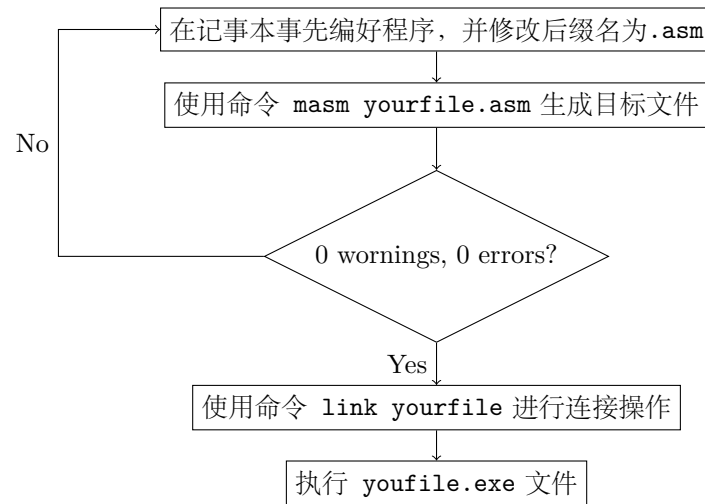


图 1: 执行流程

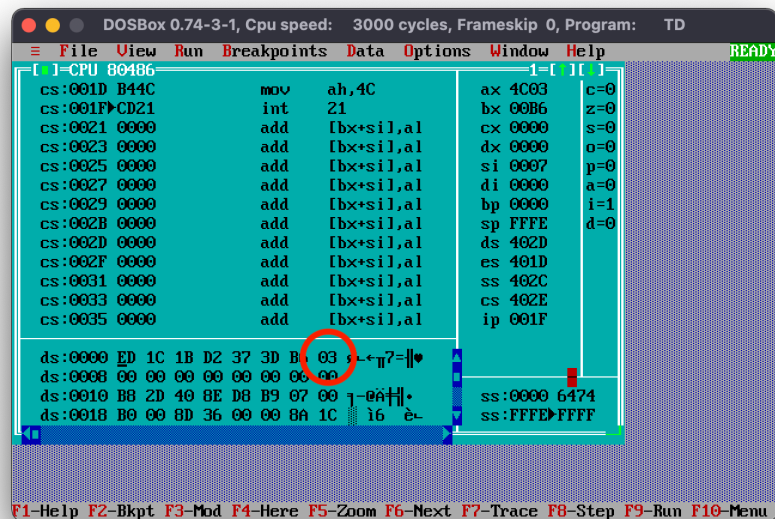


图 2: 预习实验结果

三. 实验任务

实验全部资料及完整代码详见薛宇飞的 [GitHub 主页](#)^[3]。具体实验任务如下：

1. 在一个数据块中找出最大数：
假设有数据 22、46、32、72、84、16、156，且为不带符号的正整数，数据块的长度存放在 CX 寄存器中，找出其中的最大数存放在以 MAXN 为符号的单元中。
2. 求无符号字节数据之和，和数为 8 位二进制数：
假设有数据 38、55、26、12、23，数据块的长度存放在 CX 寄存器中，和数存放在以 SUM 为符号的单元中。
3. 求无符号字节数据之和，和数为 16 位二进制数：
假设有数据 58、25、45、73、64、43，数据块的长度存放在 CX 寄存器中，和数存放在以 SUM 为符号的字单元中。
4. 求两个十进制数相乘的积 (53348×9)，被乘数和乘数均以非压缩 BCD 码表示，并存放在内存中，乘积以非压缩 BCD 码的格式存放在以 SUM 为起始符号的单元中。
5. 试分别用数据传送指令和串传送指令编写程序，将以 STR1 为首地址的字节存储单元中的数据 30H、31H、32H、33H、34H、35H、36H、37H、38H、39H、40H、41H、42H、43H、44H、45H，传送到以 STR2 为首地址的字节存储单元中。
6. 假设任务 1 中的数据为有符号数，请找出其中的最大数存放在以 MAXN 为符号的单元中。
7. 将任务 4 的乘积在屏幕上显示出来。**提示：**用 DOS 系统功能调用的字符串显示的功能。

要求：上述所有任务的程序运行结束后，均要返回 DOS。

四. 实验程序及分析

(一) 实验一

实验流程如图 3：

实验要求代码、正确结果（图4）如下：

```
exp31.asm
1  DATA    SEGMENT
2  NUM      DB 22,46,32,72,84,16,156    ;定义数据段
3  MAXN     DB ?    ;定义结果存放处
4  DATA    ENDS
5
6  MAIN     SEGMENT
7  ASSUME   CS:MAIN,DS:DATA
8  START:   MOV  AX,DATA
```

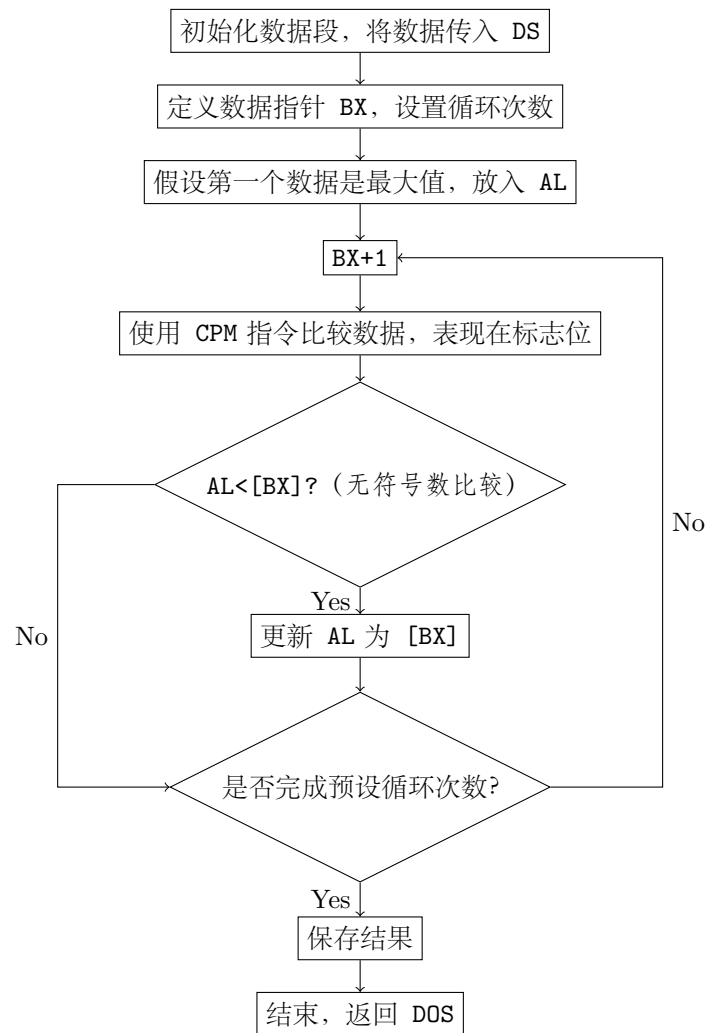


图 3: 执行流程

```
9      MOV    DS,AX      ;DATA 数据段传入DS
10     LEA    BX,NUM      ;获取NUM偏移地址，传入指针BX
11     MOV    CX,06       ;循环次数为6
12     MOV    AL,[BX]     ;初始假设第一个数据是最大值
13 AGAIN: INC    BX       ;BX指针后移1位
14     CMP    AL,[BX]     ;比较数据大小并更新标志位
15     JNBE   NEXT        ;如果AL>[BX]，跳转至NEXT
16     MOV    AL,[BX]     ;如果AL<[BX]，更新AL
17 NEXT:  LOOP  AGAIN     ;循环CX次
18     MOV    MAXN,AL     ;将结果存入MAXN
19     MOV    AH,4CH
20     INT    21H
21 MAIN  ENDS
22 END  START
```

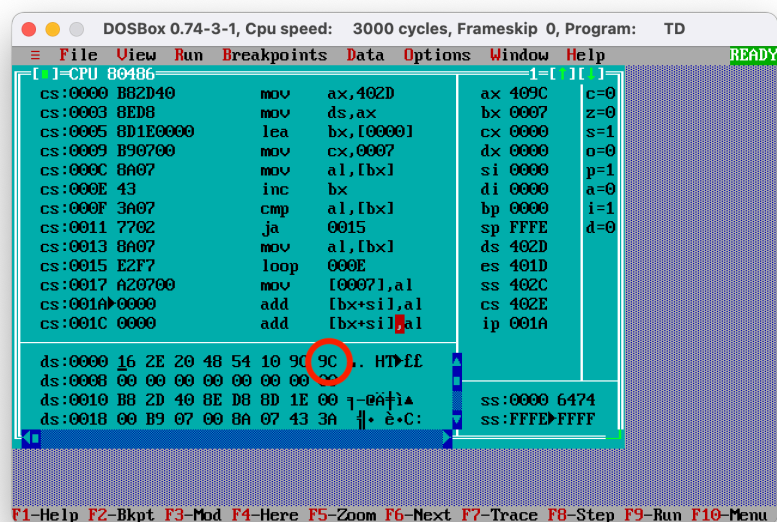


图 4: 实验结果

注意 1: JA/JNBE 用法小结

与 CPM 指令配合使用，严格大于时候跳转。

(二) 实验二

实验流程如图 5:

实验要求代码、正确结果 (图6) 如下:

exp32.asm

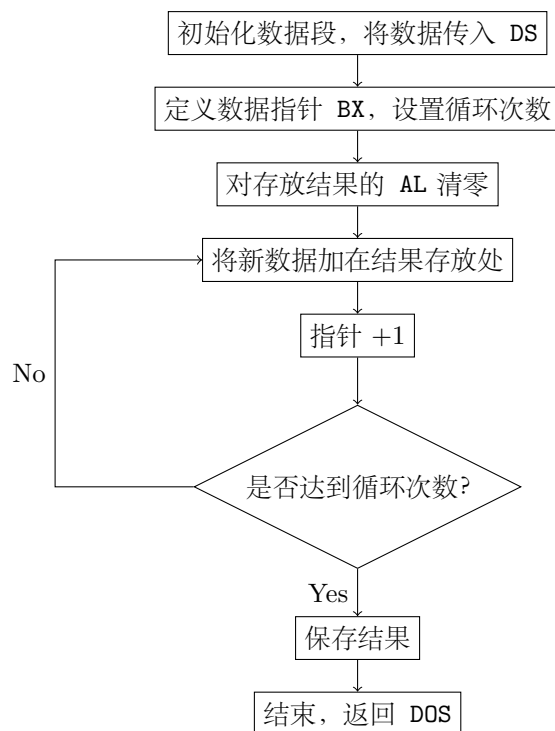


图 5: 执行流程

```
1  DATA    SEGMENT
2  NUM      DB 38,55,26,12,23
3  SUM      DB ?
4  DATA    ENDS
5
6  CODE SEGMENT
7  ASSUME CS:CODE,DS:DATA
8  START:  MOV  AX,DATA
9          MOV  DS,AX      ;DATA 数据段传入DS
10         MOV  CX,05      ;设置循环次数
11         LEA  BX,NUM      ;取NUM的地址给BX
12         SUB  AL,AL       ;将存放结果的AL清零
13  NEXT:  ADD  AL,[BX]     ;添加数据
14         INC  BX         ;指针加1
15         LOOP NEXT       ;循环,次数为CX
16         MOV  SUM,AL      ;储存结果
17         MOV  AH,4CH
18         INT  21H
19  CODE ENDS
20  END  START
```

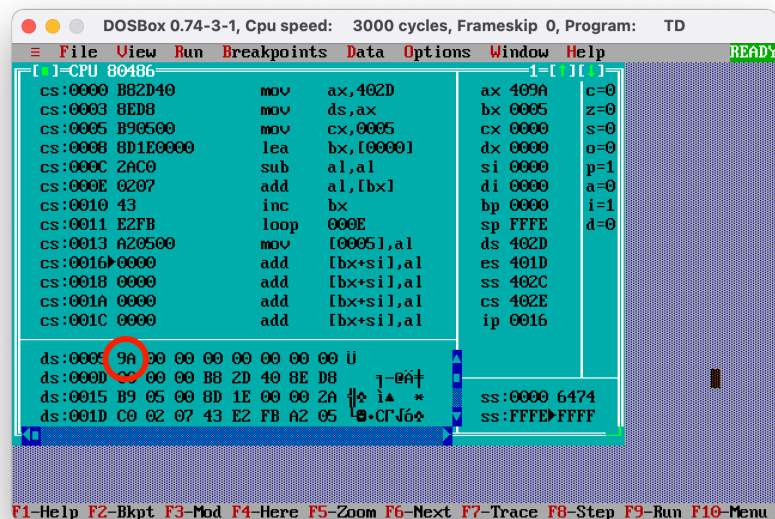



图 6: 实验结果

(三) 实验三

实验流程如图 7:

实验要求代码、正确结果(图8)如下:

exp33.asm

```
1  DATA    SEGMENT
2  NUM      DB 58,25,45,73,64,43
3  SUM      DW ?
4  DATA    ENDS
5
6  CODE SEGMENT
7  ASSUME CS:CODE,DS:DATA
8  START:  MOV  AX,DATA
9          MOV  DS,AX      ;将DATA数据传入DS
10         LEA  BX,NUM      ;NUM地址传送给BX
11         MOV  CX,6        ;循环次数为6
12         SUB  AX,AX        ;AX清零
13  NEXT:  ADD  AL,[BX]      ;这里用于低位的加法
14         ADC  AH,0         ;上一步加法可能会进位在这一步加进来
15         INC  BX          ;指针加1
16         LOOP NEXT        ;循环CX次
17         MOV  SUM,AX       ;传送结果
18         MOV  AH,4CH
19         INT  21H
```

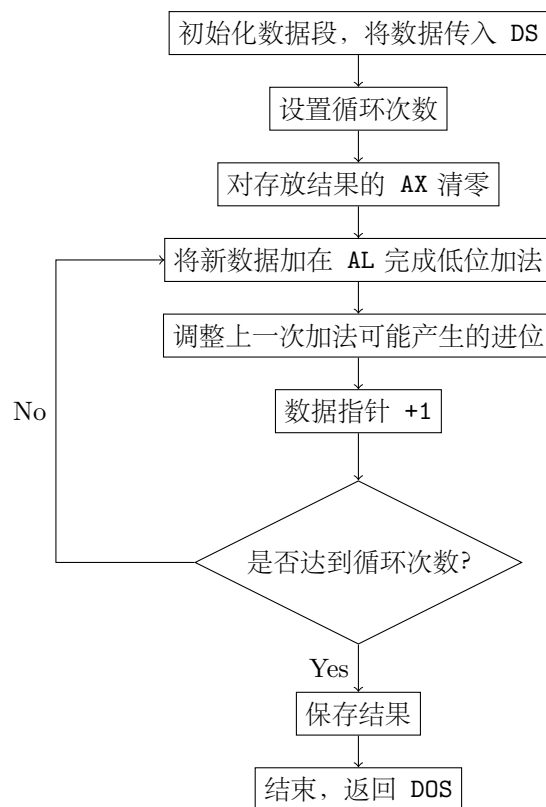


图 7: 执行流程

```

20 CODE ENDS
21 END START
    
```

注意 2: ADC 指令

ADC AH,0 通过加 0 并加标志位来将标识位考虑进去。

(四) 实验四

实验流程如图 9:

实验要求代码、正确结果 (图10) 如下:

exp34.asm

```

1 DATA SEGMENT
2 DATA1 DB 08,04,03,03,05
3 DATA2 DB 09
4 SUM DB 6 DUP(00) ;把00复制6次，占用6个BYTE
5 DATA ENDS
6
7 CODE SEGMENT
8 ASSUME CS:CODE,DS:DATA
    
```

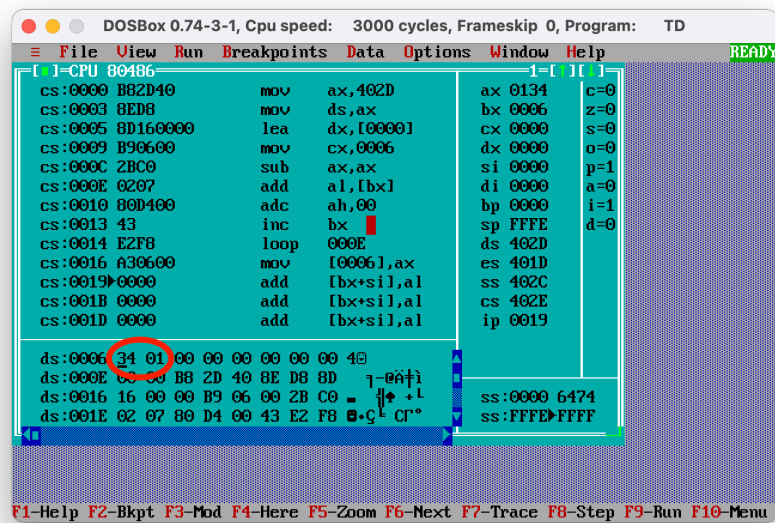


图 8: 实验结果

```

9      START: MOV AX,DATA
10          MOV DS,AX                ;手动分配段地址
11          MOV SI,OFFSET DATA2      ;DATA2的偏移地址给了SI
12          MOV DL,[SI]               ;把SI内所指向的存储器的内容给了DL
13          MOV SI,OFFSET DATA1      ;DATA1的偏移地址给了SI
14          MOV DI,OFFSET SUM         ;SUM的偏移地址给了DI
15          MOV CX,05                 ;循环次数为5
16      NEXT: MOV AL,[SI]              ;一位位开始算，先从第一位开始
17          INC SI                     ;指针加一，指向下一位
18          MUL DL                     ;和09相乘
19          AAM                        ;非压缩BCD码的乘法调整指令
20          ADD AL,[DI]                ;[DI]中有上一个算出来的进位，先把进位加上
21          AAA                        ;调整成非压缩BCD码
22          MOV [DI],AL                ;调整好之后，把正确的低位给了[DI]
23          INC DI                     ;存放下一个数字的位置
24          MOV [DI],AH                ;把上一个调整产生的进位放到DI中
25          LOOP NEXT
26          MOV AH,4CH
27          INT 21H
28      CODE ENDS
29      END START

```

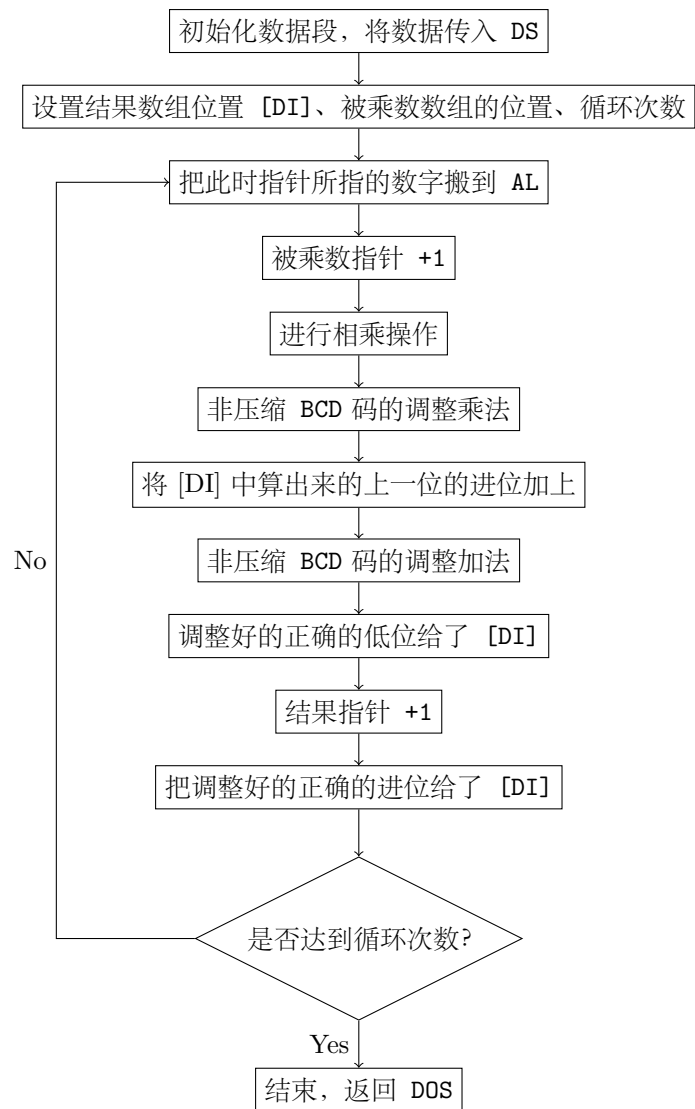


图 9: 执行流程

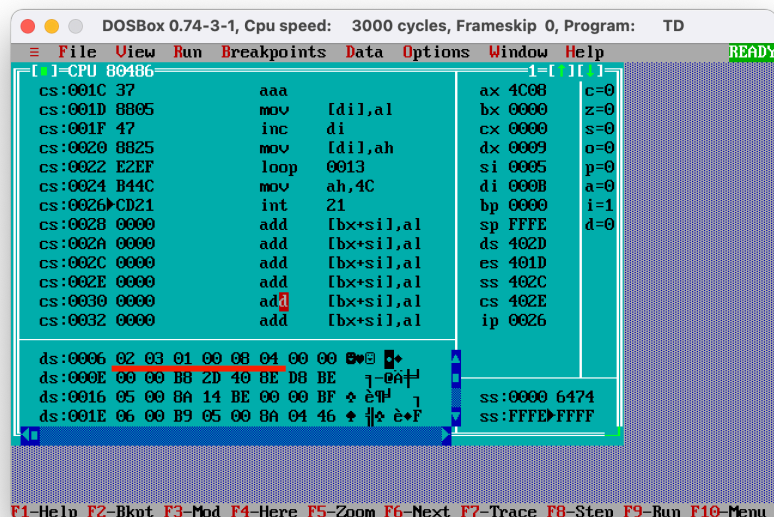


图 10: 实验结果

(五) 实验五

子实验 1 流程如图 11: 子实验 2 流程如图 12:

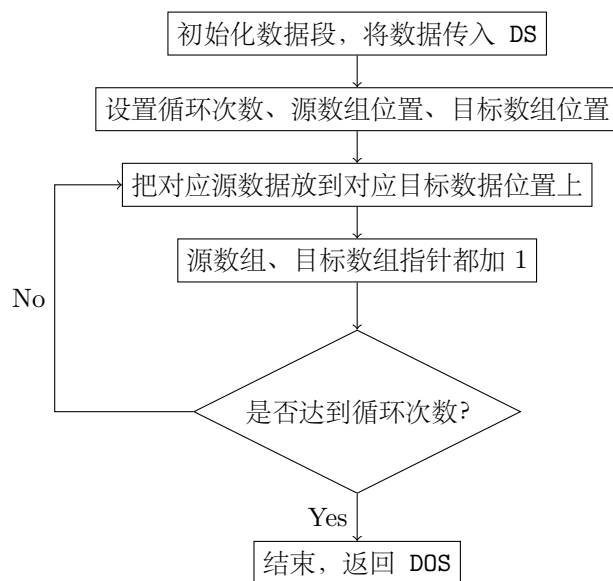


图 11: 执行流程

实验要求代码、正确结果（图13以及图14）如下：

exp351.asm

```

1  DATA  SEGMENT
2  STR1   DB  30H,31H,32H,33H,34H
3         DB  35H,36H,37H,38H,39H
    
```

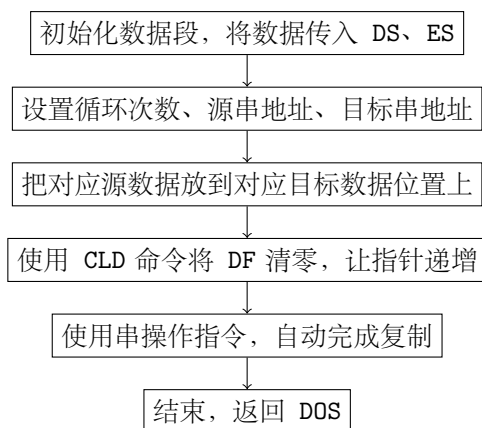


图 12: 执行流程

```

4          DB 40H,41H,42H,43H,44H,45H
5  COUNT EQU $-STR1          ;定义数据长度
6  STR2   DB COUNT DUP(0)    ;复制产生一个一样长度的数据段
7  DATA  ENDS
8
9  CODE SEGMENT
10 ASSUME DS:DATA,CS:CODE
11 START: MOV AX,DATA
12         MOV DS,AX
13         LEA SI,STR1        ;源数组
14         LEA DI,STR2        ;目标数组
15         MOV CX,COUNT       ;数据个数复制给CX
16 NEXT:  MOV AL,[SI]
17         MOV [DI],AL        ;通过AL寄存器间接传递
18         INC SI
19         INC DI              ;指针加1
20         LOOP NEXT
21         MOV AH,4CH
22         INT 21H
23 CODE ENDS
24 END START

```

exp352.asm

```

1  DATA SEGMENT
2  STR1 DB 30H,31H,32H,33H,34H,35H,36H,37H
3        DB 38H,39H,40H,41H,42H,43H,44H,45H
4  COUNT EQU $-STR1
5  STR2 DB COUNT DUP(?)

```

```
6 DATA ENDS
7
8 CODE SEGMENT
9 ASSUME DS:DATA,ES:DATA,CS:CODE
10 START: MOV AX,DATA
11         MOV DS,AX
12         MOV ES,AX ;初始化DS和ES
13         LEA SI,STR1
14         LEA DI,STR2 ;初始化源串和目标串地址
15         MOV CX,COUNT ;初始化串长度
16         CLD ;DF清零表示指针+
17         REP MOVSB ;重复移动
18         MOV AH,4CH
19         INT 21H
20 CODE ENDS
21 END START
```

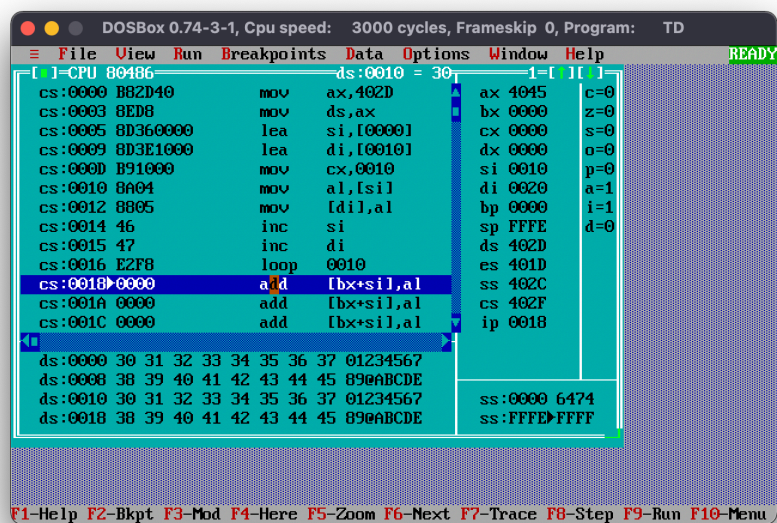


图 13: 实验结果

注意 3: 数据传送指令和串传送指令两种传送方式各自特点

数据传送指令需要定义两个指针，分别更新；串传送指令只需定义好源和目标的位置、长度，即可自动进行。

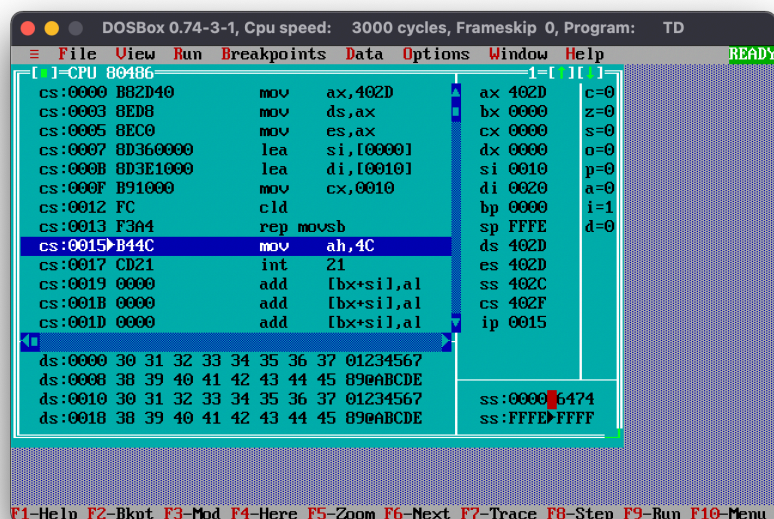


图 14: 实验结果

(六) 实验六

实验流程如图 15:

实验要求代码、正确结果 (图16) 如下:

exp36.asm

```

1  DATA    SEGMENT
2  NUM      DB 22,46,32,72,84,16,156    ;定义数据段
3  MAXN     DB ?    ;定义结果存放处
4  DATA    ENDS
5
6  MAIN     SEGMENT
7  ASSUME   CS:MAIN,DS:DATA
8  START:   MOV    AX,DATA
9           MOV    DS,AX    ;DATA数据段传入DS
10          LEA    BX,NUM    ;获取NUM偏移地址,传入指针BX
11          MOV    CX,07    ;循环次数为7
12          MOV    AL,[BX]   ;初始假设第一个数据是最大值
13  AGAIN:   INC    BX      ;BX指针后移1位
14          CMP    AL,[BX]   ;比较数据大小并更新标志位
15          JNLE   NEXT     ;如果AL>[BX],跳转至NEXT
16          MOV    AL,[BX]   ;如果AL<[BX],更新AL
17  NEXT:    LOOP   AGAIN    ;循环CX次
18          MOV    MAXN,AL   ;将结果存入MAXN
19          MOV    AH,4CH

```

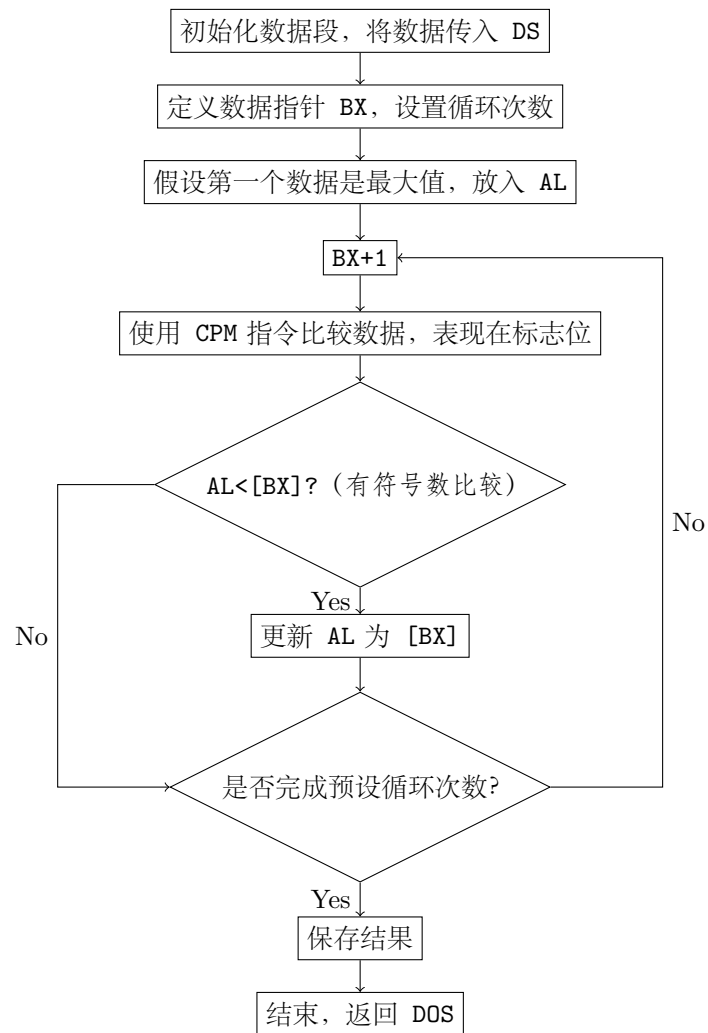



图 15: 执行流程

```
20      INT     21H
21  MAIN ENDS
22  END START
```

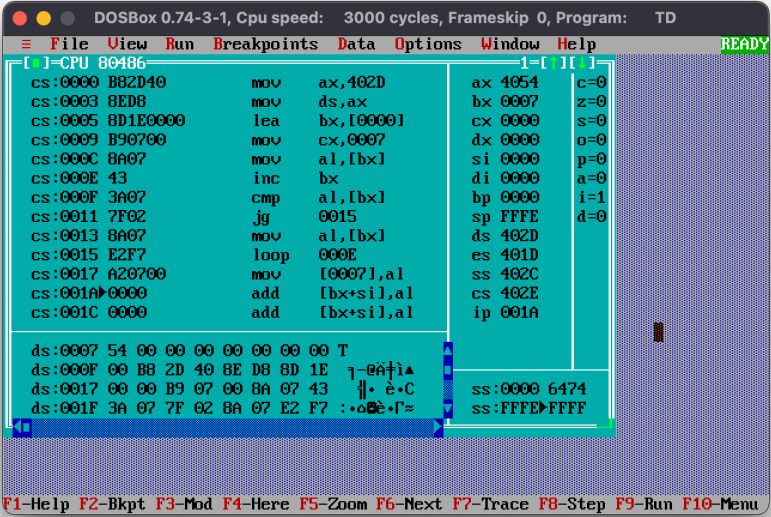


图 16: 实验结果

注意 4: 题目变化

只需要将比较大小改为有符号数的 JNLE 即可

(七) 实验七

实验流程如图 17:

实验要求代码、正确结果 (图18) 如下:

```
exp37.asm
1      DATA     SEGMENT
2  DATA1  DB  08,04,03,03,05
3  DATA2  DB  09
4  SUM     DB  6 DUP(00)    ;把00复制6次, 占用6个BYTE
5  DATA   ENDS
6
7  CODE SEGMENT
8  ASSUME CS:CODE,DS:DATA
9  START:  MOV AX,DATA
10         MOV DS,AX          ;手动分配段地址
11         MOV SI,OFFSET DATA2 ;DATA2的偏移地址给了SI
```

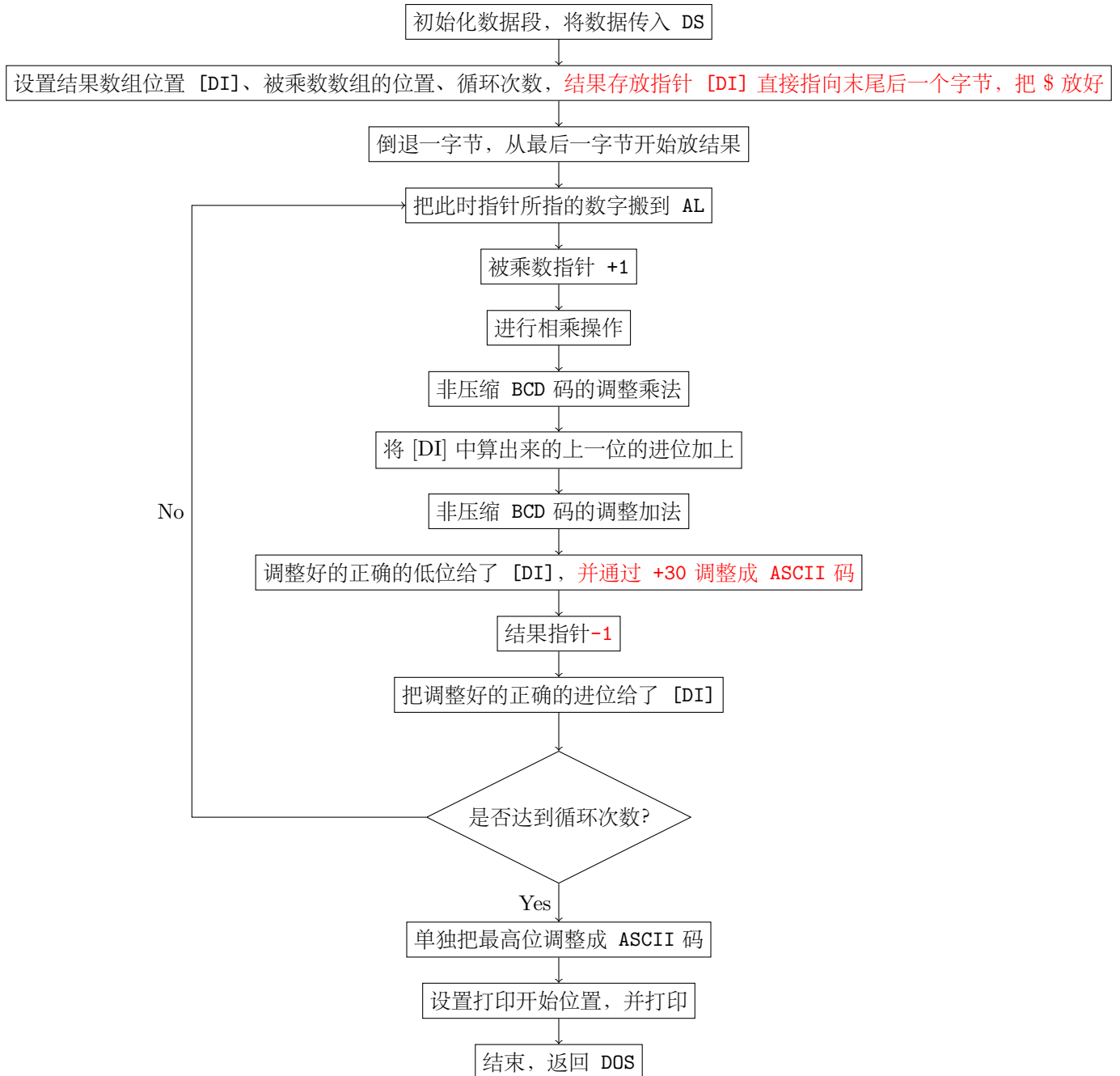


图 17: 执行流程

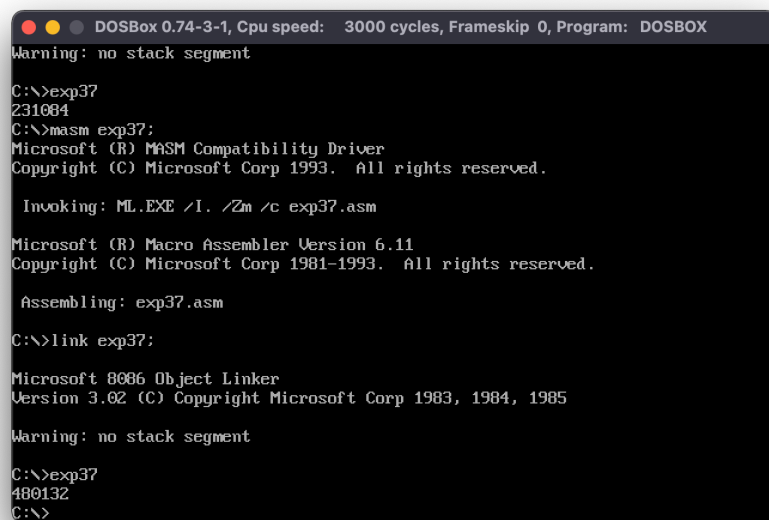
```
12      MOV DL,[SI]           ;把SI内所指向的存储器的内容给了DL
13      MOV SI,OFFSET DATA1  ;DATA1的偏移地址给了SI
14      MOV DI,OFFSET SUM     ;SUM的偏移地址给了DI
15      ADD DI,6
           ;调整指针指到末尾（已经出了数据段，是放$的地方）
16      MOV BYTE PTR [DI],24H ;$放在最后
17      DEC DI                ;开始从开辟的SUM空间的最后一个开始计算
18      MOV CX,05             ;循环次数为5
19 NEXT:  MOV AL,[SI]         ;一位位开始算，先从第一位开始
20      INC SI                ;指针加一，指向下一位
21      MUL DL                ;和09相乘
22      AAM                   ;非压缩BCD码的乘法调整指令
23      ADD AL,[DI]           ;[DI]中有上一个算出来的进位，
           ;先把进位加上
24      AAA                   ;调整成非压缩BCD码
25      MOV [DI],AL           ;调整好之后，把正确的低位给了[DI]
26      ADD BYTE PTR [DI],30H ;换算成ASCII码
27      DEC DI                ;存放下一个数字的位置
28      MOV [DI],AH           ;把上一个调整产生的进位放到DI中
29      LOOP NEXT
30      ADD BYTE PTR [DI],30H ;换算成ASCII码
31      MOV DX,OFFSET SUM     ;设置字符串的首地址为打印首地址
32      MOV AH,09H
33      INT 21H               ;打印
34      MOV AH,4CH
35      INT 21H
36
37 CODE ENDS
38 END START
```

注意 5: 字符串顺序相反问题

由于高位放在高地址，但是打印是从低位开始，所以会造成结果相反的情况，一种解决方案是先开辟 SUM 空间，然后直接挪到最后，把 '\$' 放好，然后逆着求解，最后正着打印。

参考文献

- [1] 李继灿. 新编 16/32 位微型计算机原理及应用（第五版）[M]. 5 版. 北京: 清华大学出版社, 2013.
- [2] 微机教学组. 《微计算机实验讲义》[A]. 南京: 东南大学, 2015.
- [3] <https://github.com/xyfool-66/SEU-Microcomputer-Experiments/tree/master>.



```
DOSBox 0.74-3-1, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Warning: no stack segment

C:\>exp37
231084
C:\>masm exp37;
Microsoft (R) MASM Compatibility Driver
Copyright (C) Microsoft Corp 1993. All rights reserved.

    Invoking: ML.EXE /I. /Zm /c exp37.asm
Microsoft (R) Macro Assembler Version 6.11
Copyright (C) Microsoft Corp 1981-1993. All rights reserved.

    Assembling: exp37.asm
C:\>link exp37;
Microsoft 8086 Object Linker
Version 3.02 (C) Copyright Microsoft Corp 1983, 1984, 1985
Warning: no stack segment

C:\>exp37
480132
C:\>
```

图 18: 实验结果