**计算机科学与工程学院实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **实验课程名称** | | **汇编语言程序设计** | | | **实验总成绩** |  |
| **专业** | **计算机科学与技术** | | **班级** | **计算机1803** | **指导教师签字** |  |
| **学号** | **20184484** | | **姓名** | **胥卜凡** | **实验报告批改时间** |  |
| **实验报告分项成绩**   |  |  |  | | --- | --- | --- | | **序号** | **实验项目** | **成绩** | | **1** | **循环程序设计实验** |  | | **2** |  |  | | **3** |  |  | | **4** |  |  | | **5** |  |  | | | | | | | |
| **实验课程总结**  从以下方面总结：1.实验体现知识应用和初步研究能力；2. 反映基本观察、发现问题和分析问题能力；3. 实验项目内容或者实验课程是否存在问题及下一年度改进意见）4.其他方面  总体来看，本次实验极大地丰富了我的动手实践能力，让我熟练掌握了循环语句的书写方法，对于分支结构的设计也有了更深的理解。  具体来说，为完成本次实验，我首先观察了题目，发现需要对多个数进行判断，这里需要用到循环语句的设计，需要用LOOP等操作进行循环的实现。其次，对于数的正负判断有许多方法，鉴于代码的简单性和简洁性，我采用与0比较进行跳转的方法。  综上所述，本次实验极大地增加了我的观察问题、发现问题、分析问题的能力。 | | | | | | |

**实验一**

**（一）实验内容：**

**题目：**

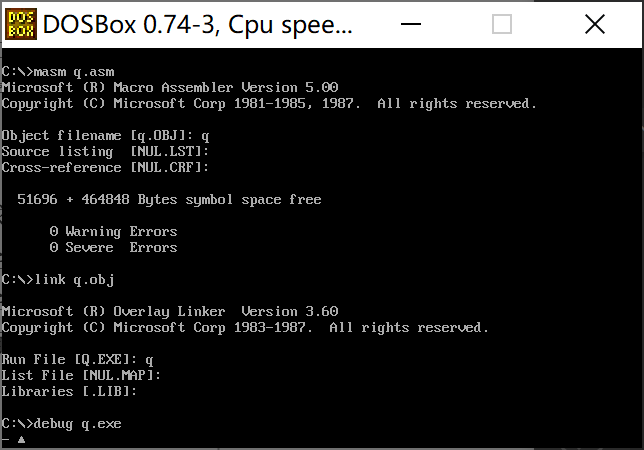
　 已知内存DATA开始的存储区存放若干个字节数据，数据个数在 COUNT单元中存放。编制程序求其中正数平均值及负数平均值， 并分别存入MEANP和MEANM单元。

**要求：**

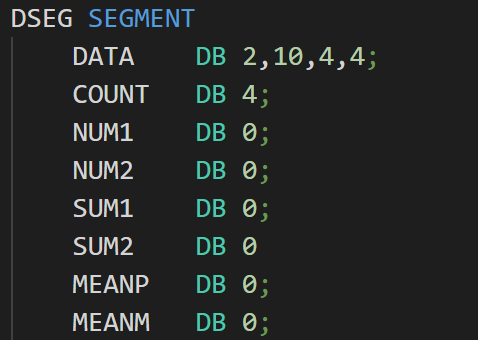
要求用几组不同的数据验证程序的正确性:  
1. 全部正数  
2. 全部负数  
3. 有正数，负数和0，注意0既不是正数，也不是负数。

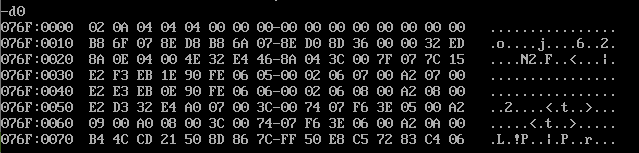
**（二）实验步骤：**

①编写程序，并进行汇编、连接，进入调试模式。。

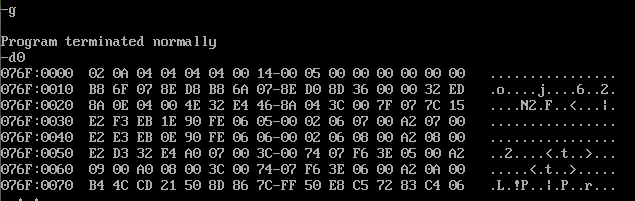


②首先查看第一组数据（2,10,4,4）



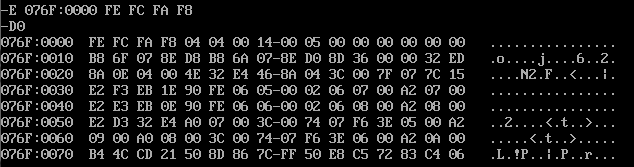


③执行程序，查看结果



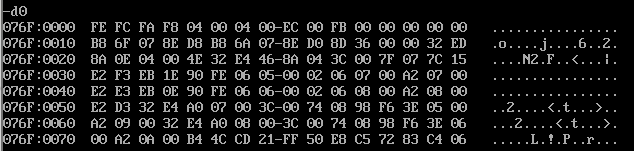
可以看到，MEANP上存储了平均值5

④用E命令更改内存数据为（0FEH,0FCH,0FAH,0F8H）(-2,-4,-6,-8)



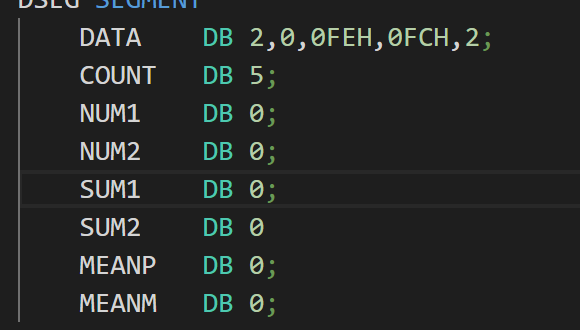
⑤从头开始重新执行到目标处

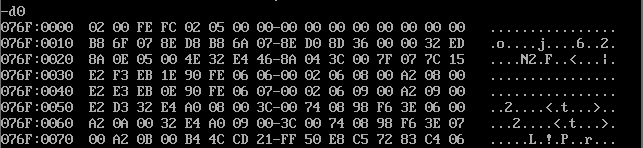




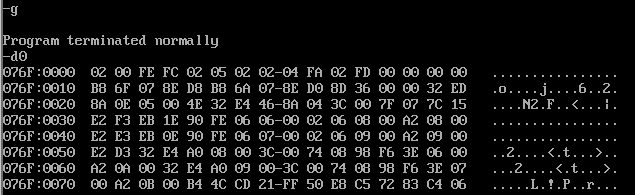
可以看到MEANM上存储了平均值FB（-5）

⑥重新编译源文件，改为以下数据（2，0，-2，-4，2）





⑦执行文件并查看结果



可以发现MEANP变为了2

MEANM变为了FD（-3）

**（三）遇到问题及思考：**

①负数出现溢出的现象

答：没有事先使用CBW进行字节转换

②如何判断一个带符号数是正（包括0）还是负的？

答：上次实验使用了80H进行判断，这次我们与0进行比较，JG大于0，JL小于0。

③如何判断多个数的情况?

答：使用LOOP进行循环操作。

**附录：**

1. **源代码:**

**实验四:**

SSEG SEGMENT    STACK

 STK DB 80 DUP (0)

SSEG ENDS

DSEG SEGMENT

    DATA    DB 2,10,4,4;

    COUNT   DB 4;

    NUM1    DB 0;

    NUM2    DB 0;

    SUM1    DB 0;

    SUM2    DB 0

    MEANP   DB 0;

    MEANM   DB 0;

DSEG ENDS

CSEG SEGMENT

    ASSUME DS:DSEG,CS:CSEG

    ASSUME SS:SSEG

START:

    MOV AX,DSEG;

    MOV DS,AX;

    MOV AX,SSEG;

    MOV SS,AX

    LEA SI,DATA;

    XOR CH,CH;

    MOV CL,COUNT;

    DEC SI

AGAIN:

        XOR  AH,AH

        INC SI

        MOV AL,[SI]

        CMP AL,0

        JG  CASE2

        JL  CASE3

        LOOP AGAIN

        JMP ENDSS

CASE2:  INC NUM1

        ADD AL,SUM1

        MOV SUM1,AL

        LOOP AGAIN

        JMP ENDSS

CASE3:  INC NUM2

        ADD AL,SUM2

        MOV SUM2,AL

        LOOP AGAIN

ENDSS:

        XOR  AH,AH

        MOV  AL,SUM1

        CMP  AL,0

        JE   ER

        CBW

        IDIV NUM1

        MOV  MEANP,AL

ER:

        XOR  AH,AH

        MOV  AL,SUM2

        CMP  AL,0

        JE   ENDSSS

        CBW

        IDIV NUM2

        MOV  MEANM,AL

ENDSSS:

        MOV  AH,4CH

        INT  21H

CSEG ENDS

    END START