（正文格式：宋体，小四，首行缩进2汉字，单倍行距）

一、设计题目

四则运算 ★★★★☆

从键盘输入一个简单的表达式，如“S=8+6\*9-2+7/5”，按回车结束输入，则屏幕显示S=61.4，小数点保留 1 位，四舍五入。假设输入的表达式中只含个位十进制数和“+”、“-”、“\*”、“/”运算符，且同一运算符最多出现 2 次。

思考：如果取消上述只能输入个位数的限制，允许输入 100 以下的任意十进制数，程序如何完成？

二、设计说明

该程序已完成思考题要求

1.正确描述整个程序的功能，完成什么样的工作。

我设计的程序能针对计算结果（包括中间结果和最终结果）在8位内且各单项式中至多有一次除法运算的数学表达式，进行输入、计算和最终结果的输出。并能对输入表达式的合规性做简单的判断。

具体工作按步骤划分如下：

(1)输入表达式

(2)检测输入是否符合要求

(3)将表达式中ASCⅡ表示的数字转换为十进制数

(4)将数字、符号依次间隔存入

(5)计算表达式

(6)将最终结果的十进制数转化为ASCⅡ

(7)输出结果

2.把整个工作划分成多个任务（子程序），并说明调用关系。

图一 子程序之间调用关系

MAIN为主程序，其中MAIN对SAVENUM和SAVESG多次调用，对OPERATE和OUTPUT调用一次；OPERATE对ADDITION\SUBTRACTION\MULTIPLY\DIVISION视输入情况调用零次至多次；OUTPUT对SORT调用一次。

3.确切地定义每个子程序的功能，它与其它子程序之间的参数传递说明。

SAVENUM:将DX中存入的一个数值存储到EXP数据中。DX寄存器传递参数。

SAVESG:将AL中存入的计算符号存储到EXP数据中。AL寄存器传递参数。

OPERATE:计算表达式结果。存储器传递参数。

OUTPUT:输出计算结果。存储器传递参数。

ADDITION:计算加法。存储器传递参数。

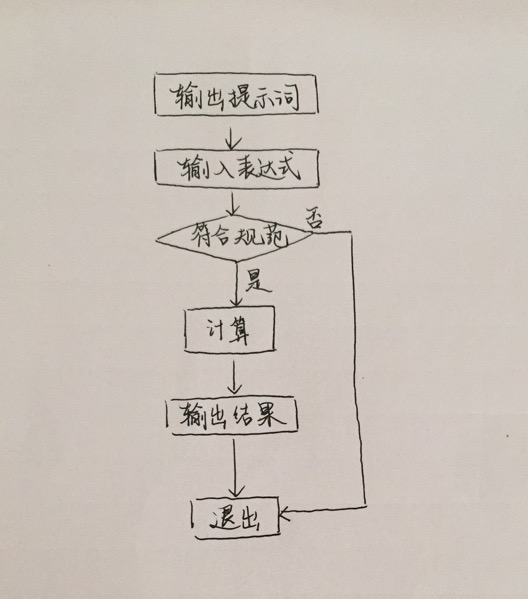
SUBTRACTION:计算减法。存储器传递参数。

MULTIPLY:计算乘法。存储器传递参数。

DIVISION:计算除法，处理此项的小数部分。存储器传递参数。

SORT:整理小数部分，进行四舍五入，并使其符号与整数一致。存储器传递参数。

4.程序框图。



图二 总流程图

5.子程序说明及其流程图。

SAVENUM:将数字存入两个字节中。

SAVESG:将符号存入一个字节中。

ADDITION:将加号前后两个数字相加，将结果存在加号后的两个字节中，加号前的两个字节清零。

SUBTRACTION:用减号前的数字减去减号后的数字，将结果存在减号后的两个字节中，减号前的两个字节清零。

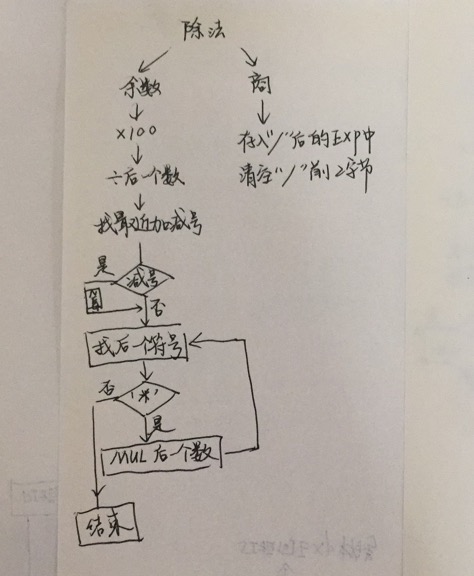
MULTIPLY:乘号前后的数字相乘，将结果存在乘号后的两个字节中，乘号前的两个字节清零。

以上几个子程序因算法简单，代码段小，故流程图不再赘示。

DIVISION:【如图三】包含除法和此项中小数部分的处理两部分。

除法部分：除号前的数字除以除号后的数字，将商存在后两个字节中，除号前两个字节清零。

此项中小数部分的处理：将余数乘100后再除后一个数字（这里称商为TMPRESULTP，一律看作正数），根据离前面最近的加号或减号确定TMPRESULTP的正负。为防止此单项式其后又有乘法，向后检查此项还有无乘号，若有则用TMPRESULTP乘乘号后的数字。



图三 DIVISION流程图

OPERATE:【如图四】先按从左到右的顺序进行乘除运算，后按从左到右的顺序进行加减运算，将整数部分的结果存入RESULT。计算加减法的时候，由于单项式里可能存在连乘、连除或乘除混合运算，数字间会因乘除号的间隔而很难判定加法还是减法。我在这里引入了SWITCH，SWITCH为0时为加法，SWITCH为1时为除法，减少了往前遍历所需的时间。

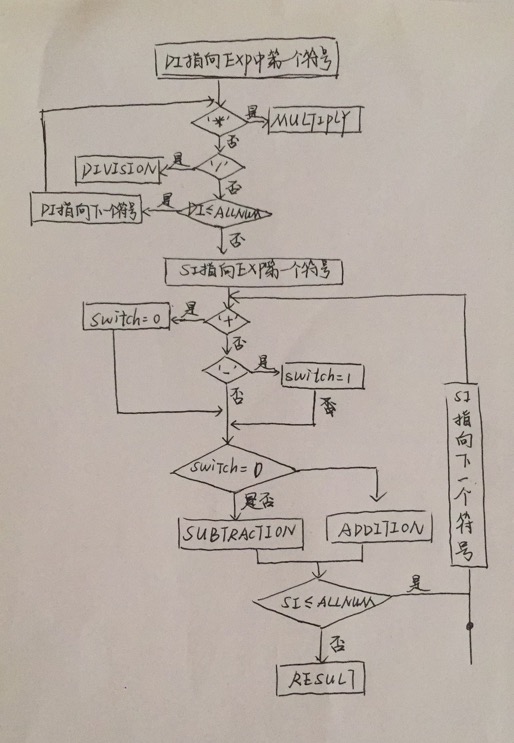
SORT:【如图五】该子程序可分为提整、跟随符号、四舍五入三部分。

提整:RESULTP/100的结果按符号加入到RESULT所代表的整数部分。

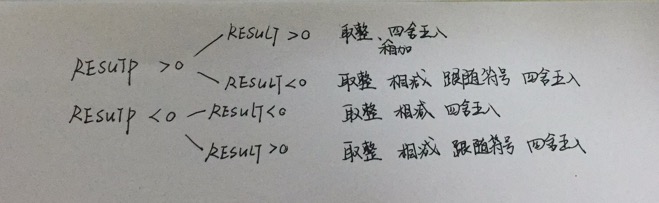
跟随符号:该程序输出时符号由RESULT部分判断，RESULT部分为负，则小数部分也要为负，RESULT部分为正时亦然。

四舍五入:RESULTP MOD 10是来检查小数点后第二位，比较其是否大于五，大于五，则在RESULTP/10的基础上加一，所得RESULTP的结果就是小数点后的一位。

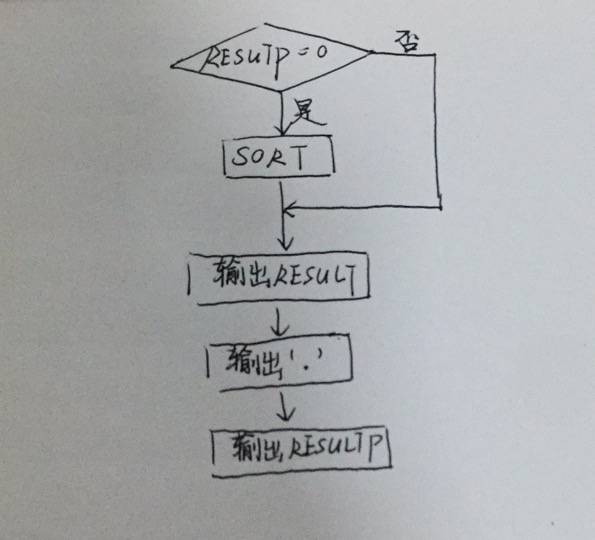
OUTPUT:【如图六】判断RESULTP是否等于于0，不等于0则调用SORT，然后依次输出RESULT（整数部分），小数点，RESULTP（小数点后一位）



图四 OPERATE流程图

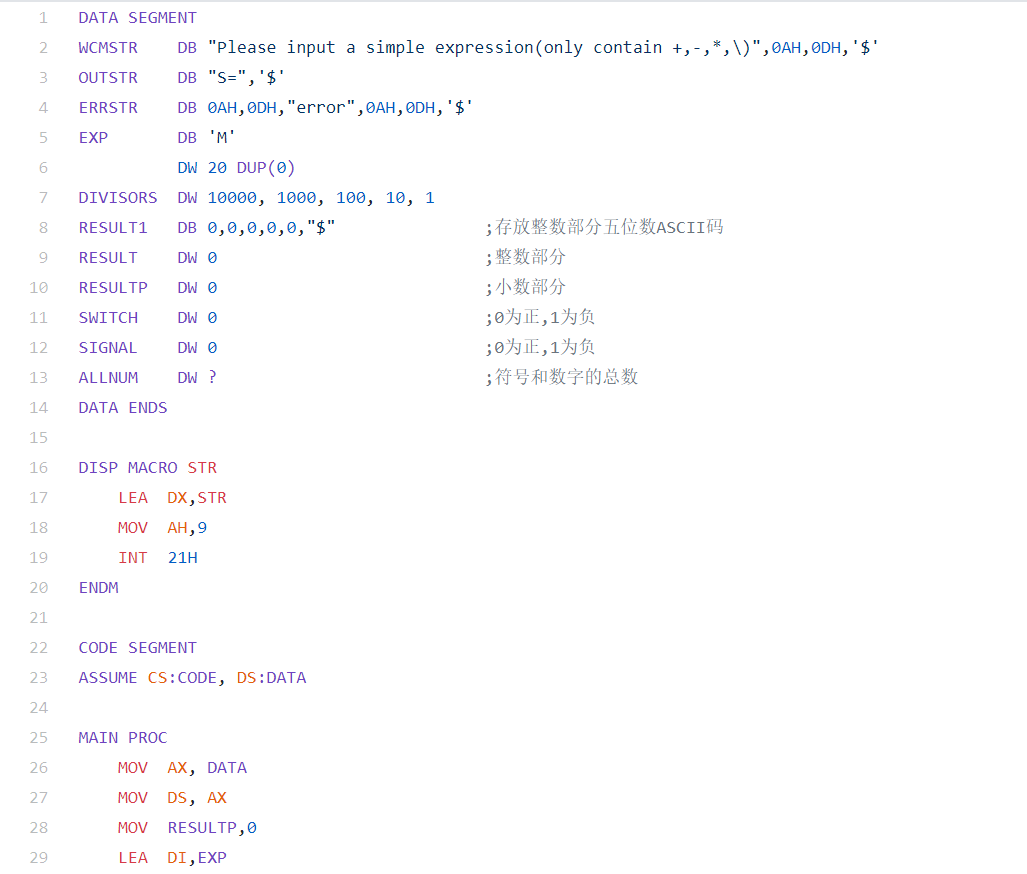


图五 SORT流程图

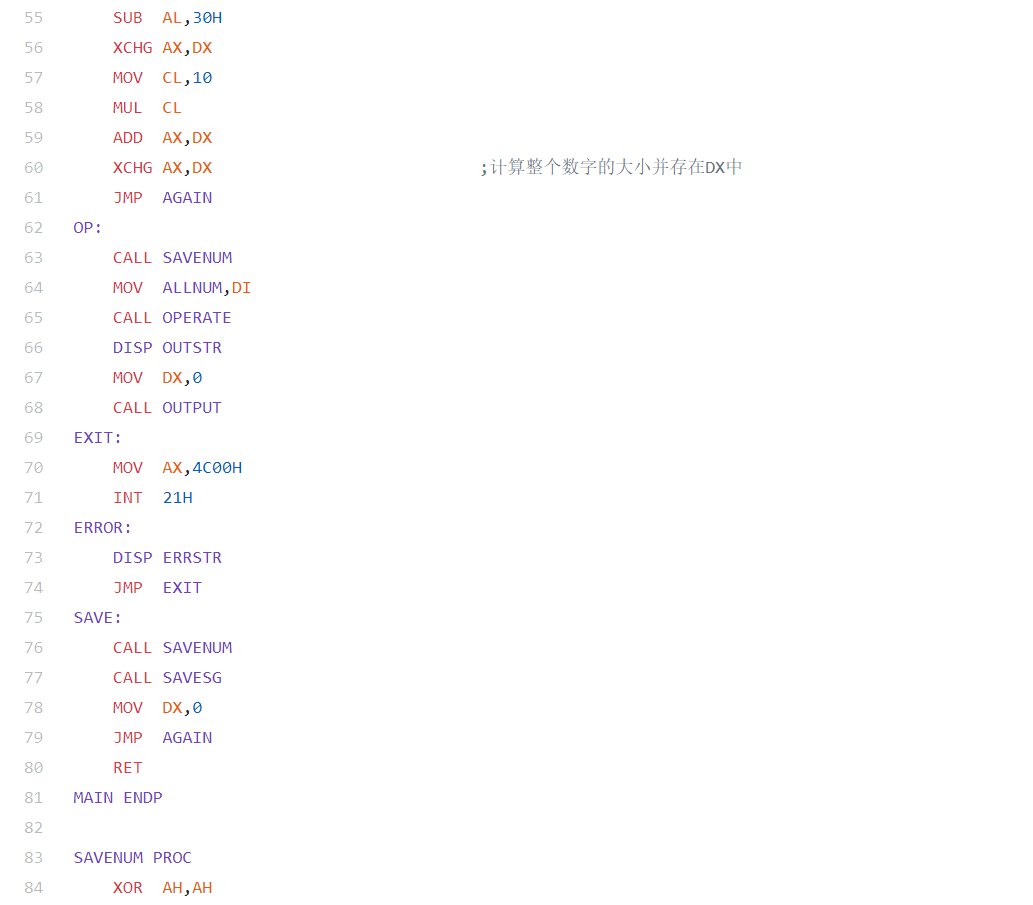


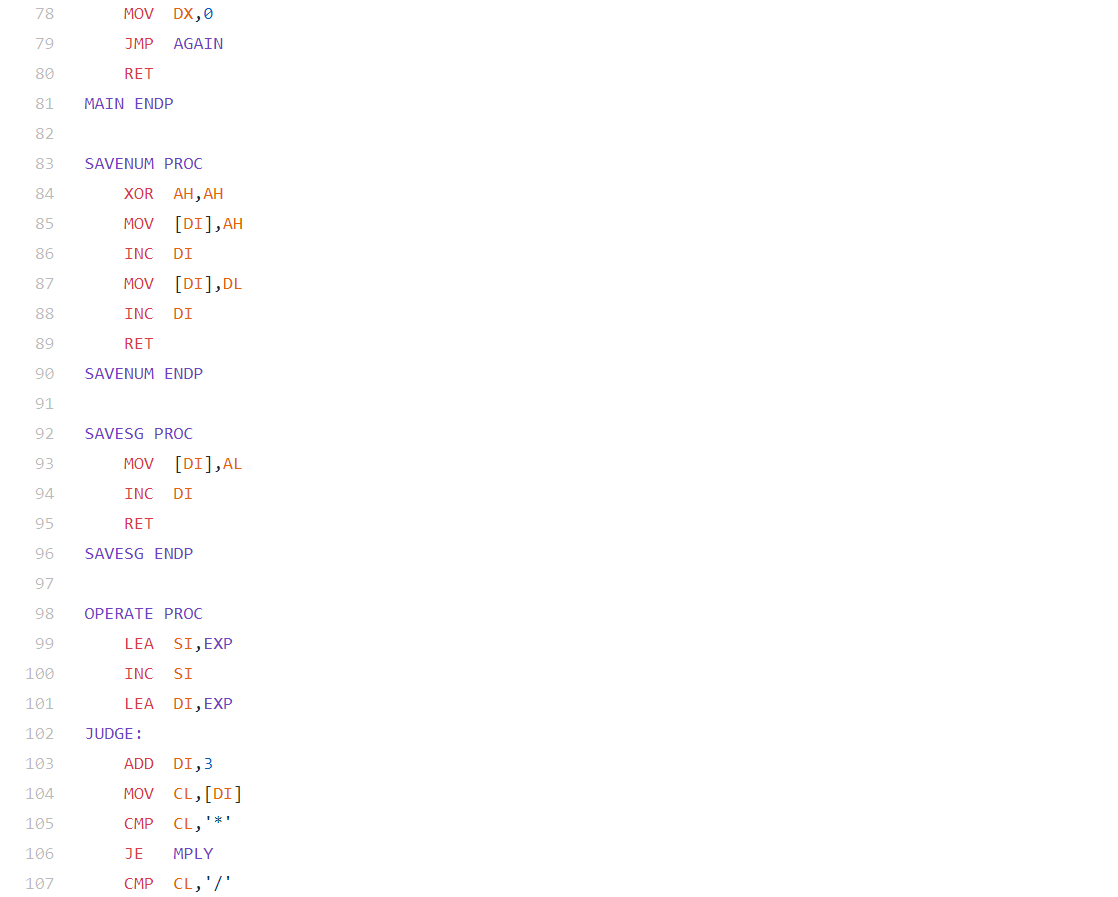
图六 OUTPUT流程

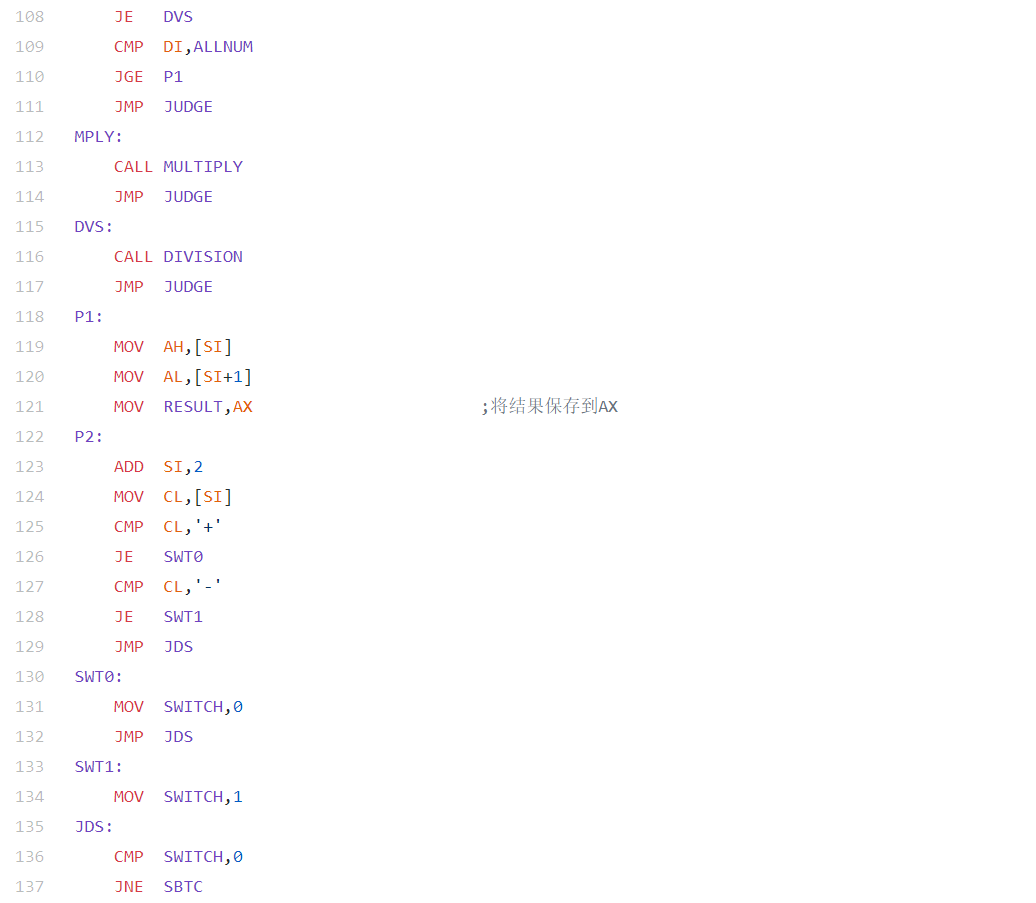
6.程序清单

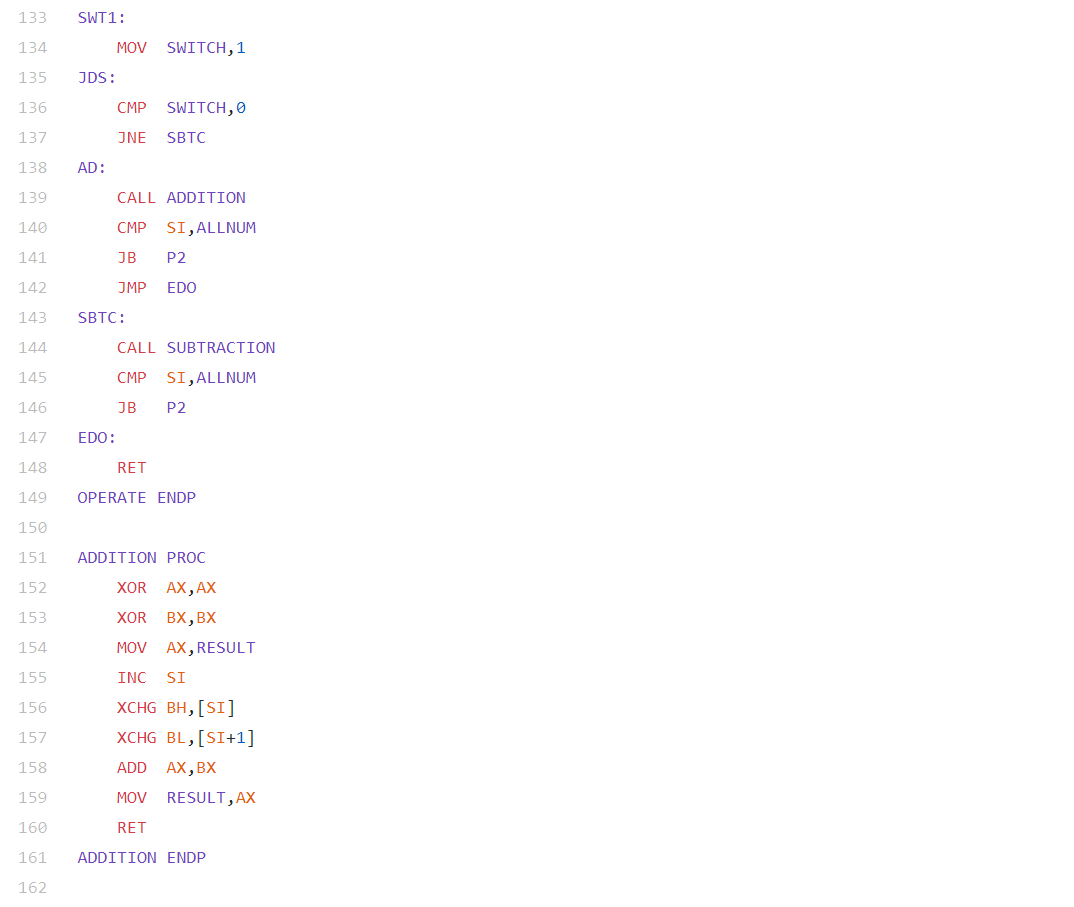


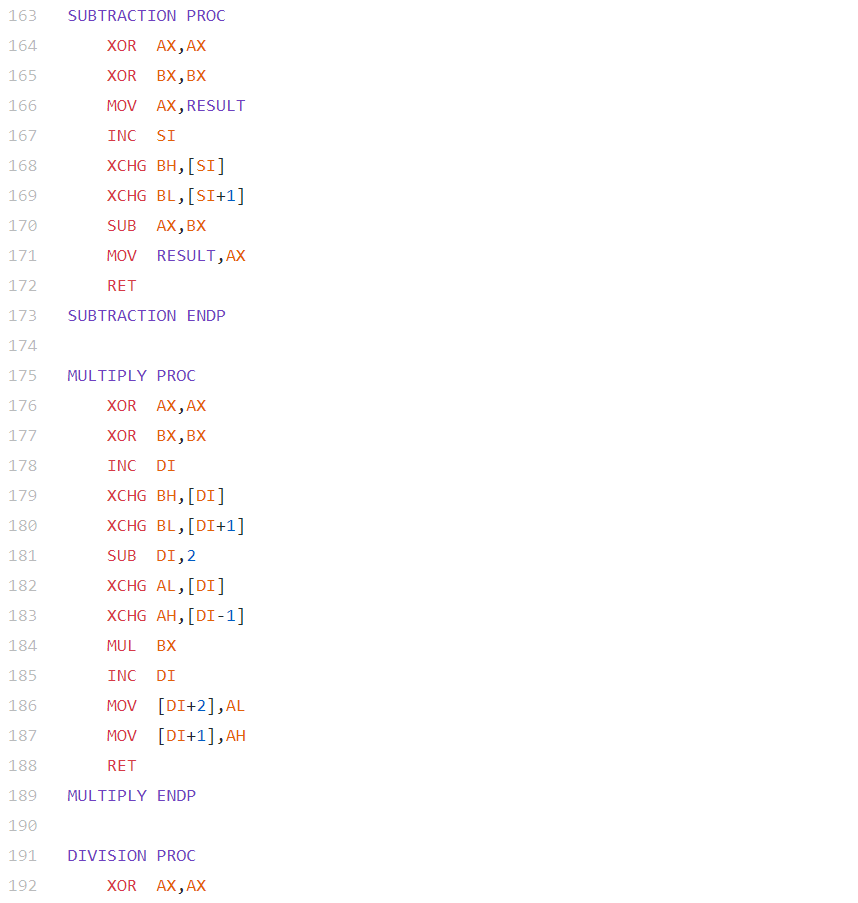


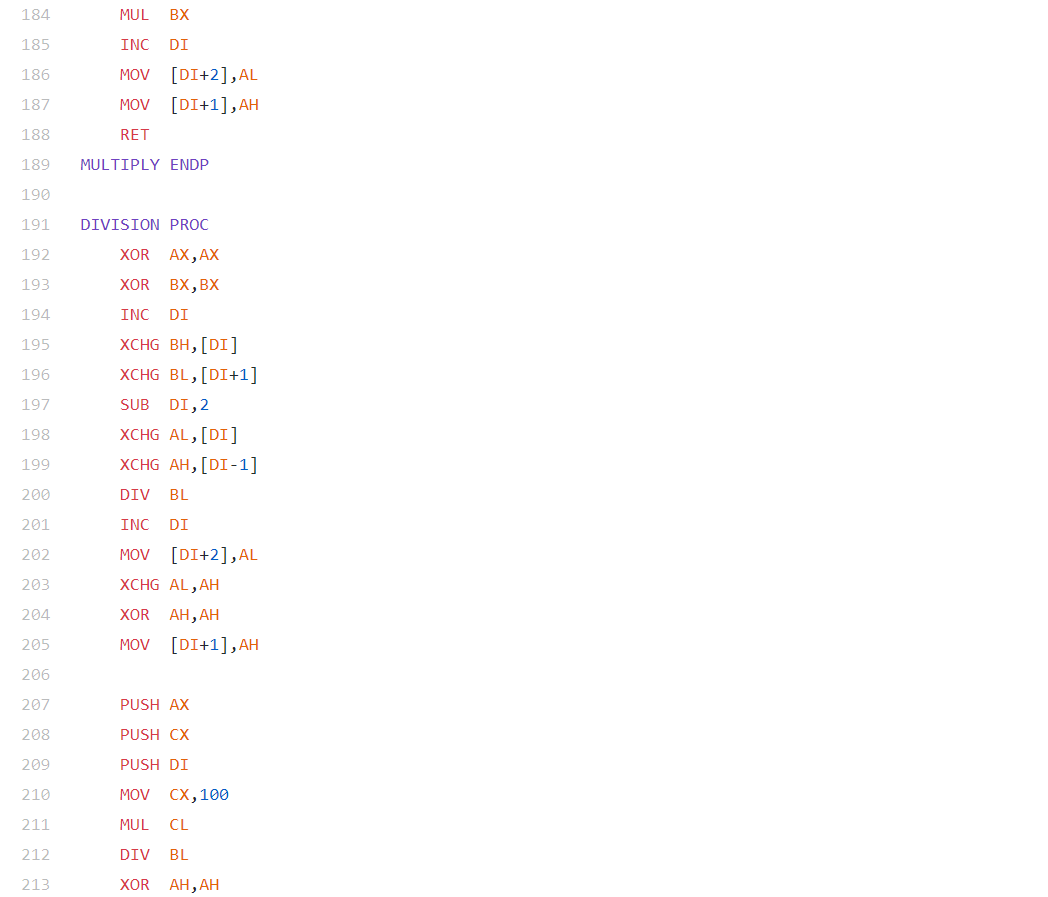


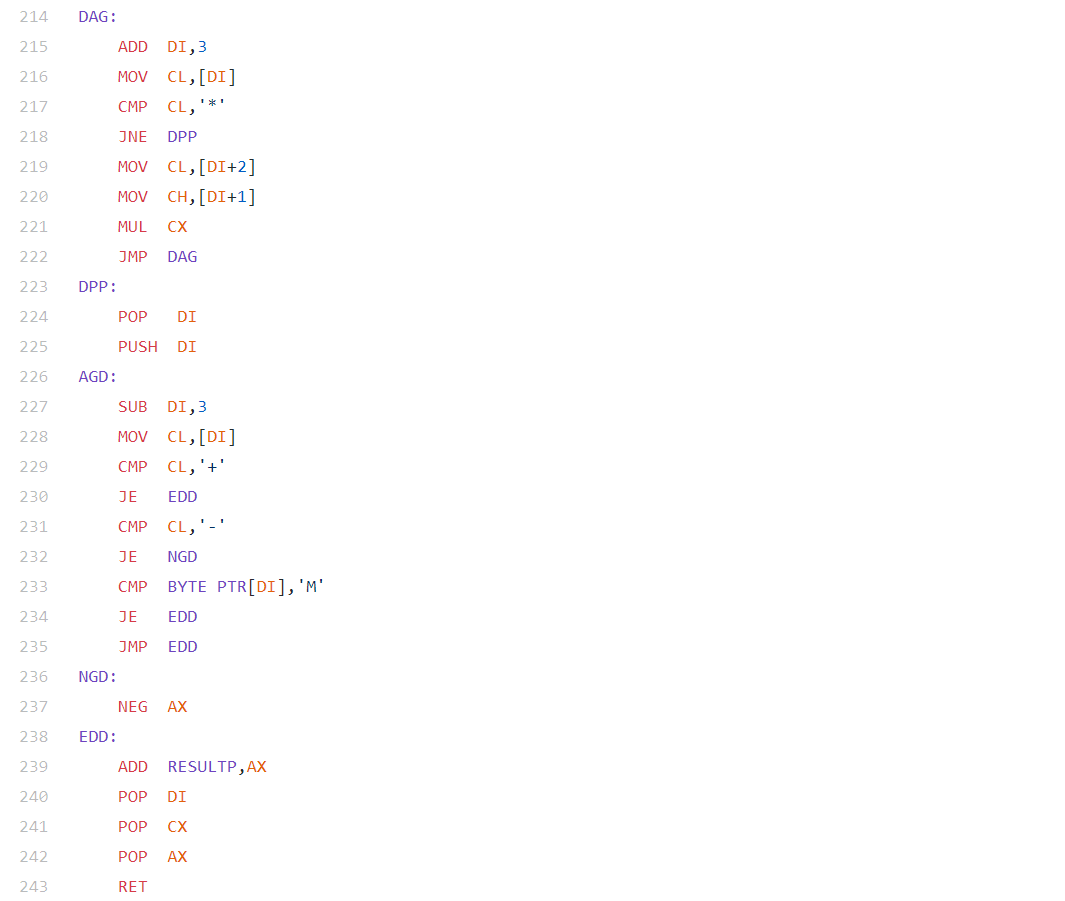


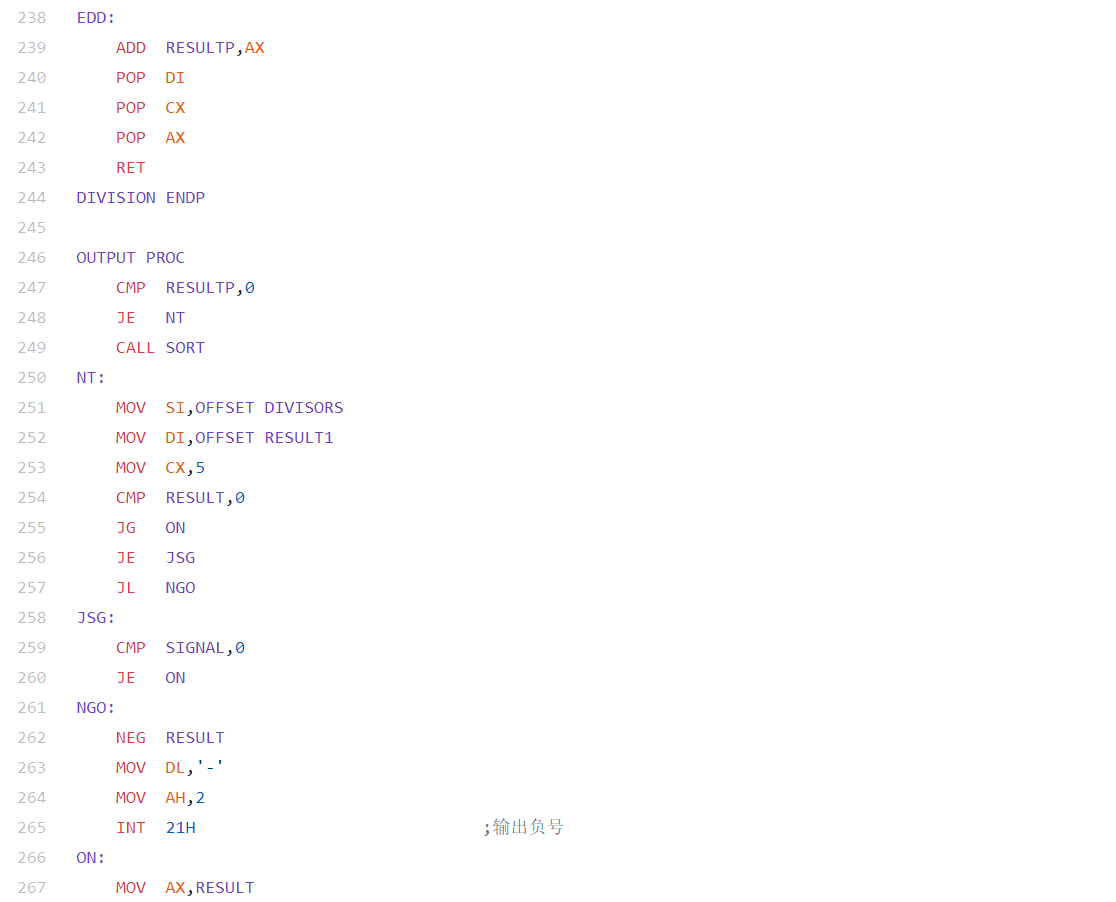




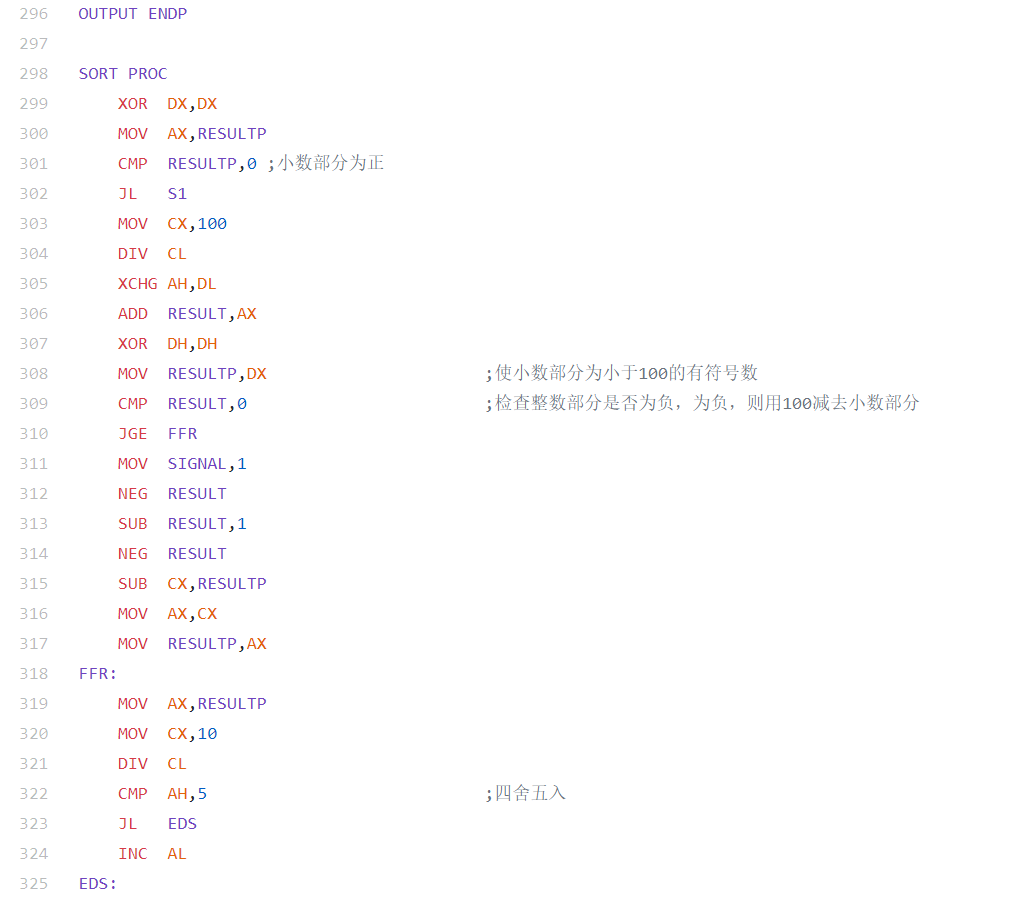


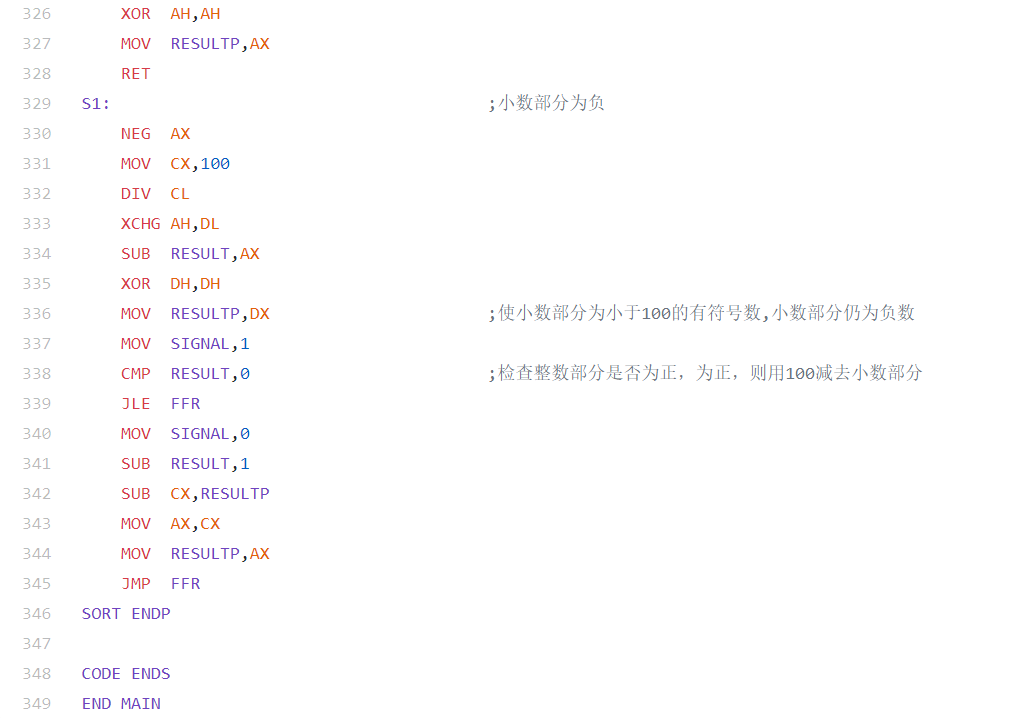














三、调试说明

1.调试情况。如上机遇到的问题及解决方法、观察到的现象及其分析、对程序设计技巧的总结和分析等。

(1)乘除运算后加减的运算

先遍历乘除，计算结果保留到符号后方；再用SWITCH表示加减号，防止遇到乘除号时不清楚加减，也能减少不必要的遍历。

(2)遇到了不显示结果的情况

多次尝试后我发现只有遇到除法的时候才会出现该种情况，用T语句逐步调试后发现是DIVISION方程出现了问题，使程序陷入了死循环。

(3)RESULT为0的时候难以判断符号

我试过用SF来检验RESULT的正负，但不太可靠。我于是用SIGNAL来表示符号，在SORT部分就把最终输出时的符号确定好。

(4)先除后乘时忽略余数的乘法

设计DIVISION时向后遍历，及时用余数乘乘数。

(5)RESULTP输出非数字符号‘:’

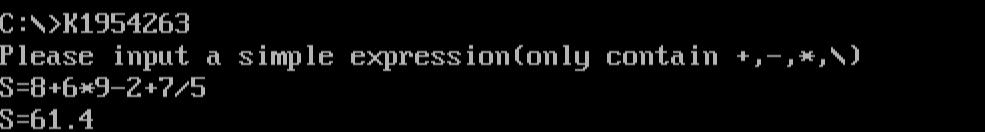
通过对比ASCⅡ，调用SORT子程序时，有时借位会使RESULTP=10，10B+30H对应的58B对应的ASCⅡ即是‘:’。改进方法就是如果RESULTP为0就不要调用RESULTP，以免出现RESULTP大于9的情况。

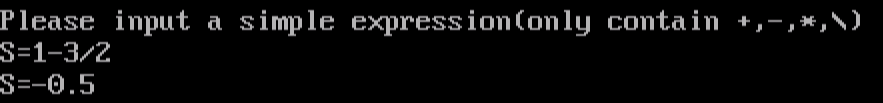
2.连接的要求说明。

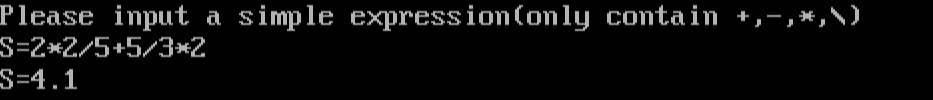
LINK K1954263.OBJ

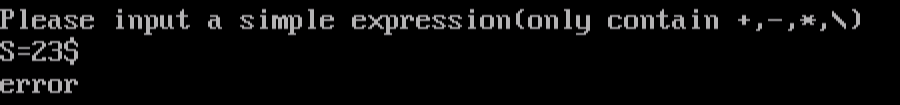
3.至少 4 组测试数据的运行结果，应考虑到数据的完整性和代表性，如临界

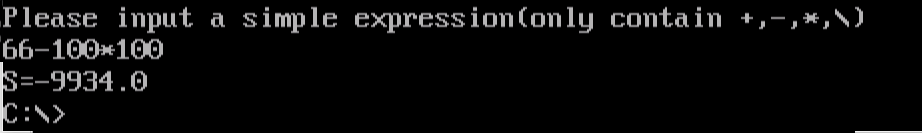
数据，并能判断数据的正确性。











4.运行结果分析。 在要求范围内，运行正确。

四、使用说明

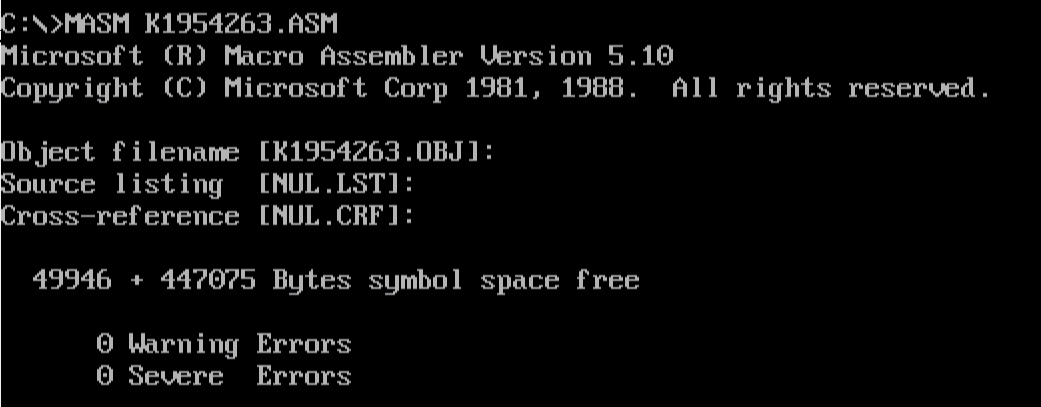
1. 程序运行的软硬件环境，适用范围：

硬件：常见的PC，台式机均可

软件：DOSbox （64 位的 Windows7 或者 Windows8/10）或Windows自带的命令提示符（32位的Windows7）

2. 程序的使用方法、调试方法、操作步骤：

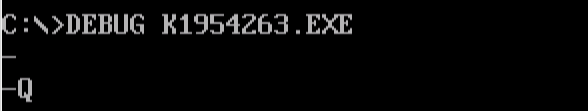
在DOSbox中，建议将编程所需要的程序都建立在 masm 目录（文件夹）下，第一次使用时，输入MASM K1954263.ASM，显示如下:



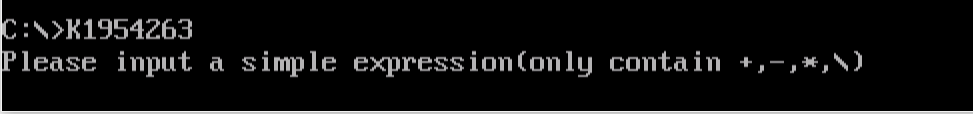
输入LINK K1954263.OBJ，显示如下:



输入DEBUG K1954263.EXE,显示如下,输入Q退出。



此后再输入K1954263即可正常进行操作，显示如下:



输入表达式即可运行。

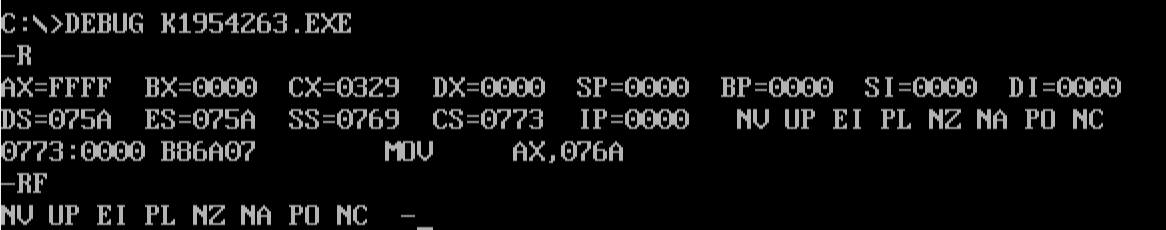
对于32位Win7，直接把ASM文件汇编连接得到的exe文件直接放到命令提示符里面运行就行了。

具体的调试方法如下：

输入DEBUG K1954263.EXE

(1)输入r，查看寄存器和标志位状态

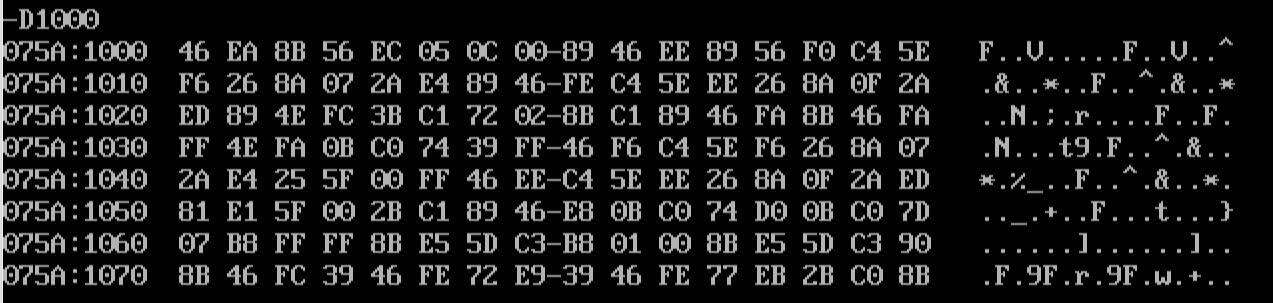
输入rf，查看和修改标志位状态



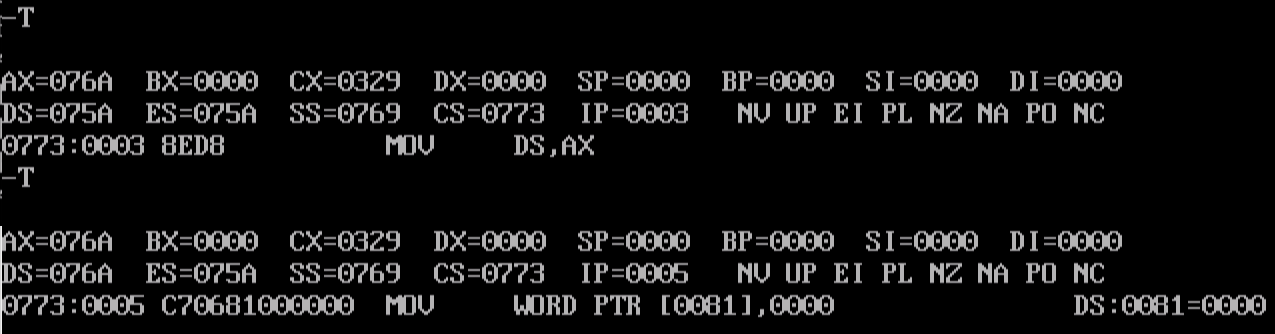
(2)输入D，显示内存单元内容的命令

①-D[地址]

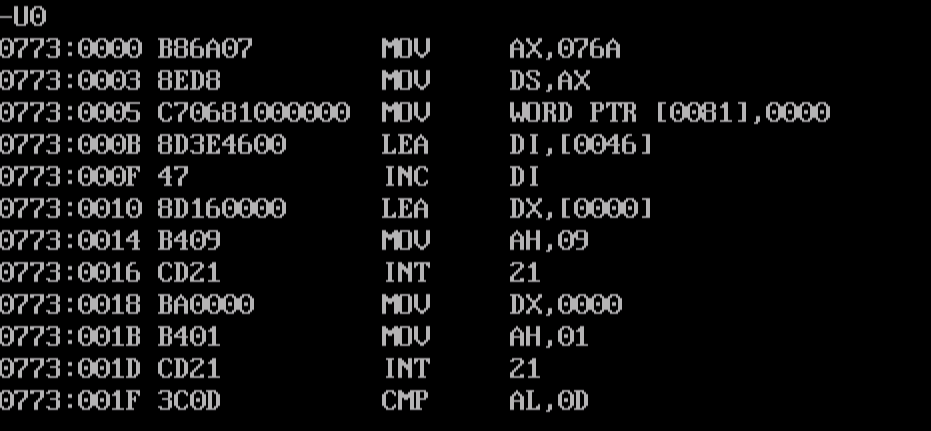
②-D [地址范围]



(3)输入T，单步调试



(4)输入U0，进行反汇编



(5)输入P，可以直接跳过子程序，得到子程序结果

(6)输入Q，退出DEBUG，返回DOS

3.要求输入信息的类型及格式。

输入的信息必须是合法的数学表达式，可以以“S=”开头（注意S必须是大写），计算符号只包括‘+’‘-’‘\*’‘/’，每个数字在100以内，单项式中至多出现一个除号（如2/3/4不被允许）

4.出错信息的含义及注意事项等。

输入时不能出现除‘S’、‘=’、‘+’、‘-’、‘\*’、‘/’、‘0’到‘9’以外的符号否则程序会输出“error”并自动退出程序；注意输入表达式时不要连续输入两个符号，如“S=+2”,注意除数不能为0

五、课程总结（必须详细，建议 1000~2000 字）

1.课程总结

这两周我都是在家中学习，环境比较安静。除了有一次网络连接不好的情况，其他都与线下听课没什么区别。在开课前我网购了Kip Irvine所著的汇编语言做教材，学习过程中结合书、PPT、老师提供的资料进行学习。我很少能在上机课时就完成任务，撰写实验报告时也常查资料防止自己的报告口头话、不够学术。不论是做作业还是上机往往都要从互联网、资料中去查询信息。寻址部分由于书中介绍不足，老师语速较快不理解，PPT看了好几遍，那一部分的习题也就花去了很多时间。对分支循环程序掌握的不够好，主要还是由算法引起的；基础知识、数据传送、输入输出与子程序设计部分掌握较好。相比学习高级语言，学习汇编对数据结构算法的要求较低，由于是线上的原因，和同学交流的机会较少，这是有点遗憾的地方。在上机、大作业遇到问题讨论时我只能选择与同样修习这门课的舍友进行交流。

学习了汇编，我更好的理解了计算机底层系统，理解了内存的分配和硬件系统。也解决了我之前学习中的疑问。在进入大学前，我一直不理解为什么有了Python这样简单的语言，学校里却仍要教C、汇编这种复杂的语言。学习汇编，能更好的控制计算机的硬件系统，也加深了我对计算机运行原理的理解。子程序的学习，减少了程序调试的难度。在以后编写高级语言时，也能更好地分配内存。

在教学上，有个人的小建议，希望老师能在开课前分享PPT和建议购买的书籍。

2.大作业总结

对于继续学习的要求和打算等。

自学内容：计算顺序的判断

RESULT的输出

自学时间：约3h

参考资料：<https://wenku.baidu.com/view/6e6ea67f31b765ce05081480.html>

教材、PPT

上机地点：家中

上机时数：四天（约40h）

完成情况：独立完成所有部分

独创地方：(1)SWICH的设计 简化加减法的判断。

(2)SIGNAL的设计 在整数部分为0，小数部分不为0的情况下判断最终结果的正负。

(3)小数的处理 将整数和小数部分分开处理，小数部分不为0的情况只在有除法出现的情况下，计算除法时将小数部分保留在两位以上，最后用SORT子程序来判断小数点后第二位四舍五入情况。

大作业中存在的不足：

(1)乘除混合情况的解决 除法只能在单项式中进行一次，如2/3/2，2/3商0余2，

因除法没有分配律，不能再用整数部分和小数部分分开计算的方法解决，将整数部分乘100

再加小数部分（相当于将两位小数扩大100倍）容易超出寄存器范围，在100以内数字范

围内以我个人所学无法完成。

(2)输入条件的判断不够完善 输入时不能根据输入情况判断连续多个符号输入的错

误，也无法判断输入数字是否在100以内，除数是否为0。

改进：增加子程序判断输入的合法性，多次利用CMP，Jcond指令，要能根据错

误输出错误的原因，来提醒使用者。

心得体会：

体会到了画流程图的重要性。设计程序时，需要清晰的思路，而不是想到什么就写

什么。合理的拆分出子程序并对子程序单独调试很重要，大大提高了我解决BUG的效率。

查ASCⅡ表-T单步调试来分析输出错误也是解决BUG的好方法。有想法很容易，但要

使想法落地，需要考虑的因素还有很多。多测试几组数据，许多数据会出错是有共同点

的，比如我在结果是小数点后为0的情况下输出就总会出错，检查ASCⅡ才发现是SORT

子程序调用判断的问题。汇编语言的学习使我更了解计算机的基础架构，大作业对计算

范围的要求使我理解高级语言编程时溢出时，输出一个很大的数字的原因。以后我也可

以从硬件的角度来评价自己编写的高级语言代码。彻底地理解了以前让我一头雾水的内

存空间汇编语言虽然复杂，但其仍是大有可为的。

我从书中了解到了FPU，接下来为了解决大作业中(1)存在的不足，我会继续学习

有关浮点数的知识。

六、签名