**汇编语言程序设计**

**期末报告**

**学 号\_\_\_\_\_\_\_\_\_20074411 \_\_\_\_\_\_\_\_\_\_**

**姓 名\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_**

**指导教师\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**提交日期\_\_\_\_\_\_\_\_\_\_2022.6.3\_\_ \_\_\_\_\_\_\_**

**成绩评价表**

|  |  |  |
| --- | --- | --- |
| **报告内容** | **报告结构** | **报告最终成绩** |
| **□丰富正确**  **□基本正确**  **□有一些问题**  **□问题很大** | **□完全符合要求**  **□基本符合要求**  **□有比较多的缺陷**  **□完全不符合要求** |  |
| **报告与练习题功能一致性** | **报告图表** | **总体评价** |
| **□完全一致**  **□基本一致**  **□基本不一致** | **□符合规范**  **□基本符合规范**  **□有一些错误**  **□完全不正确** |  |

**教师签字:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# 目录

## 第一题..............................................................................3

题目....................................................................................................................................................3

构思....................................................................................................................................................3

流程图................................................................................................................................................4

源代码................................................................................................................................................5

调试..................................................................................................................................................14

心得体会..........................................................................................................................................16

## 第二题.............................................................................17

题目..................................................................................................................................................18

构思..................................................................................................................................................18

流程图..............................................................................................................................................19

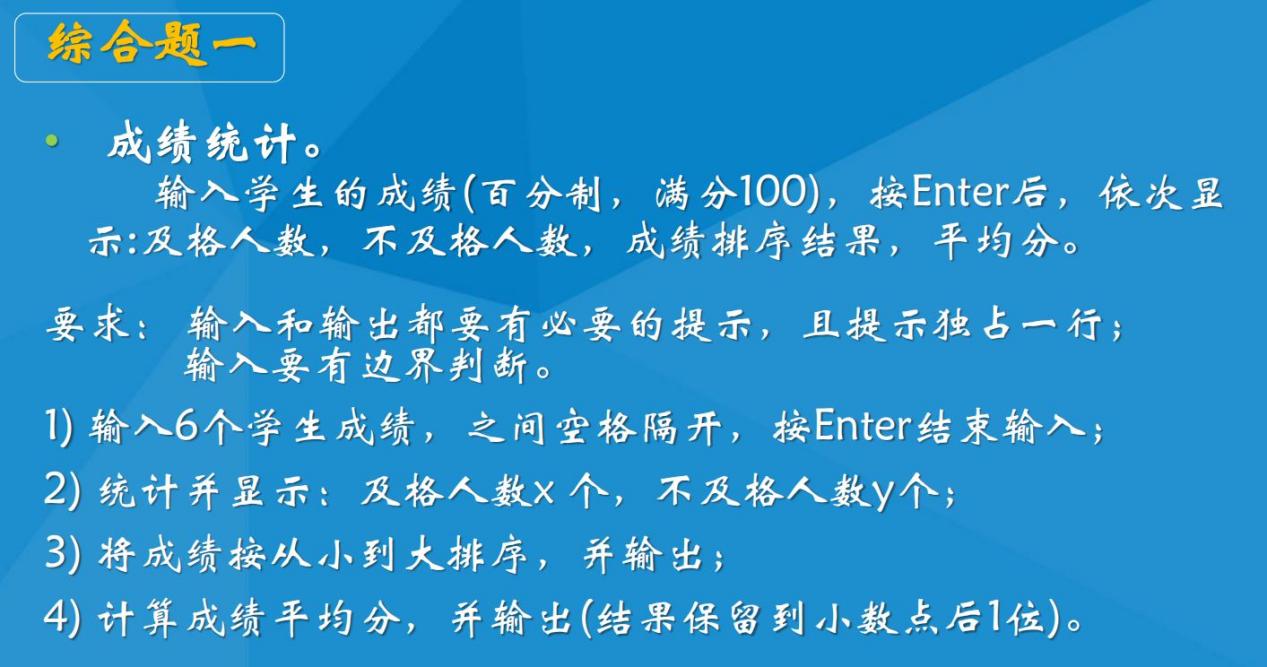
源代码..............................................................................................................................................20

调试..................................................................................................................................................25

心得体会..........................................................................................................................................26

# 第一题

## 一、题目：



**二、写代码前的构思：（加粗的是最后采用的）**

**总体分为三个部分，输入，计算，输出**

**继续再进一步可以分为成绩依次输入，第二问计算、第三问计算、第四问计算，各自输出。**

**--------------------------------------------------------------------**

输入：

输入部分一开始我采用21h号中断的1号功能一位一位的输入，并以此来分别对数字、空格、回车和其余字符进行处理，数字计算后输入遇空格时将之前算好的数存入开辟的储存单元，回车结束，其余字符报错。但是后来我发现这样子的方式导致输入的时候最后要加一个空格才能接收到最后一个成绩，而且只要中途输入了不符合标准格式的空格，就会导致后面所有成绩输入失败，所以最终我还是采用了21h的9号功能直接接收字符串。

**在data段我开辟了stu DB 50,?,50 dup('$')储存单元作为9号功能的缓冲区，地址偏移量前两位分别为开辟单元数量与接收到的字符数量；另外开辟储存单元SCORE DB 6 dup(0)，用来存放计算完成的6个成绩。**

**为了使程序简洁美观，我将所有功能都写成了子程序。**

**输入INPUTSTU子程序，21h中断9号功能将键盘输入接收进stu缓冲区，从缓冲区偏移量2即第一个字符开始首先和回车字符与空格字符比较，如果相等就分别跳转至相应位置开始将接收到的成绩一位位由ascII码转换为数字本身，如果不等就继续和数字0和9的ascII码比较来判断是否是数字，不是就报错；对数字的处理是将每个成绩的第一个数减30h后暂存在某寄存器，如果接受下一位，就把前一位乘10并与新的数字相加，遇到空格就标志成功接收了一个成绩，将其存入score相应位置，并加到提前开辟的sum字储存单元中作为总成绩，同时每接收到一个成绩就将其与60相比，同时处理标志量，为后续问题做准备；如果接收到空格，代表最后一个成绩也接收成功，此时处理后跳转至结束。**

**--------------------------------------------------------------------**

计算：

**计算部分，在输入时我们已经将获取到的数字以及他们的总和、及格与不及格成绩计算并储存完毕，所以第二问的计算相当于在输入时就顺便完成。**

**第三问需要编写排序子程序BUB，因为排序数据较少所以直接采用冒泡排序排序后输出，原理是score中储存的成绩两两比较，如果前一个大于后一个，就将两者在score中的位置互换。**

**第四问平均分，获取在输入时将获取到的总成绩，首先将其与600比较，特殊情况直接输出100.0，其余情况根据数学算法除6或除10，从十位开始到小数点后位数一位一位计算并输出。**

**--------------------------------------------------------------------**

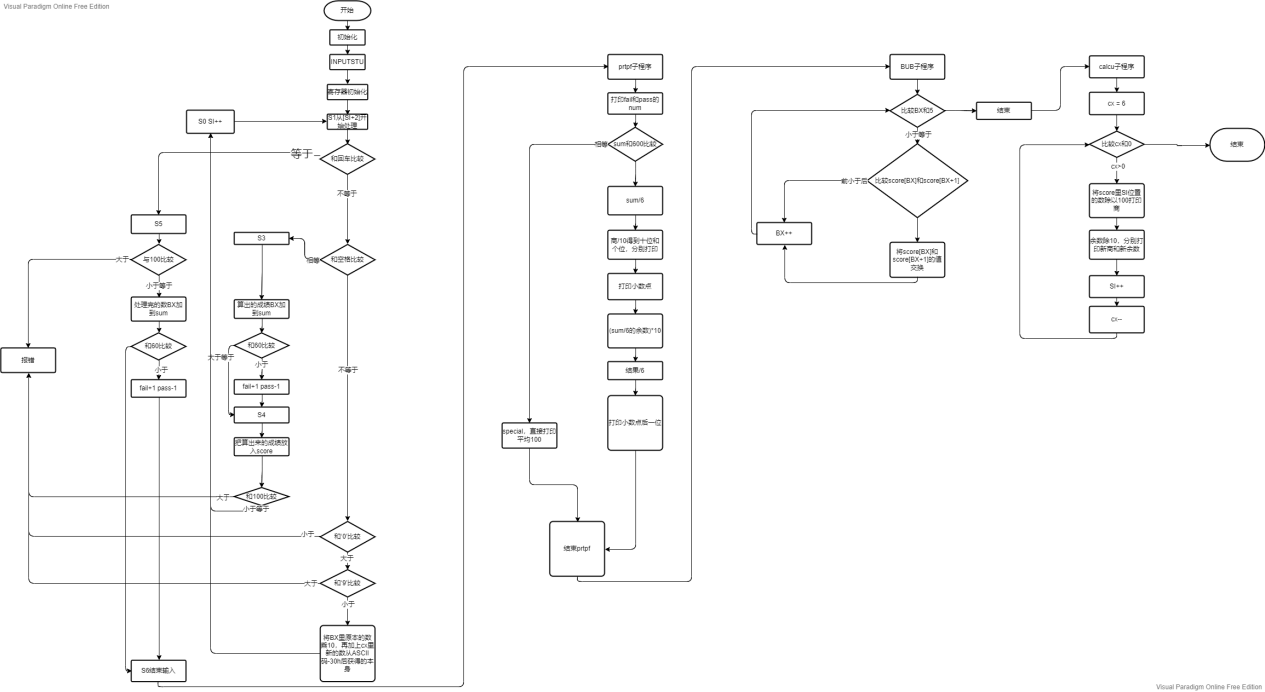
输出：

**按照题目顺序输出，每道题首先输出data端中写好的相关提示string**

**第二问直接在输入部分结束时输出提示词与输入中算好的pass和fail两个储存单元中的数**

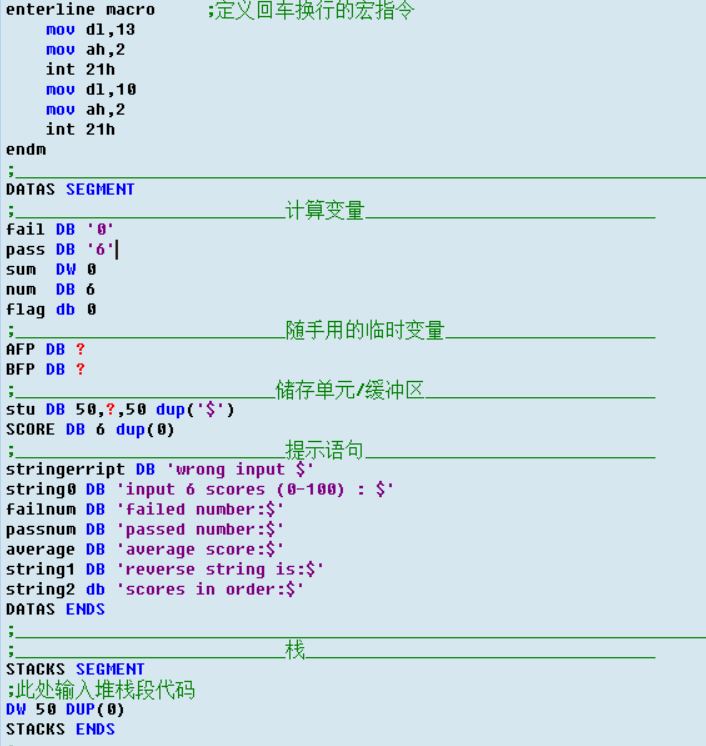
**第三问编写子程序calcu来输出，主程序中循环调用calcu程序6次输出6个数，用除100除10等的方式将score中的一个成绩每一位分开，然后一位一位输出**

1. **流程图：**

****

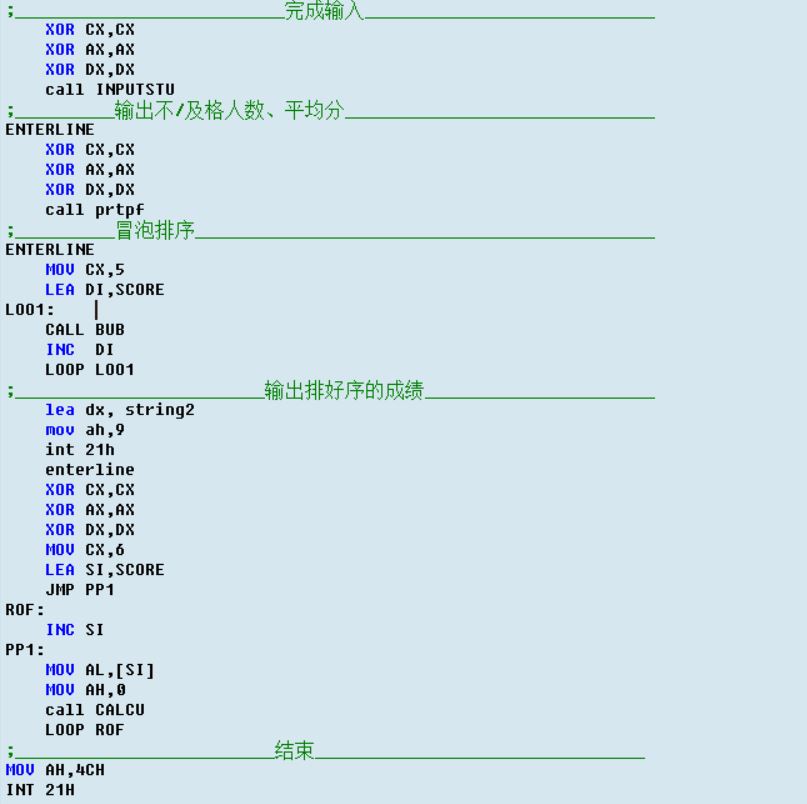
1. **程序源代码：**

代码段以外的段以及宏指令

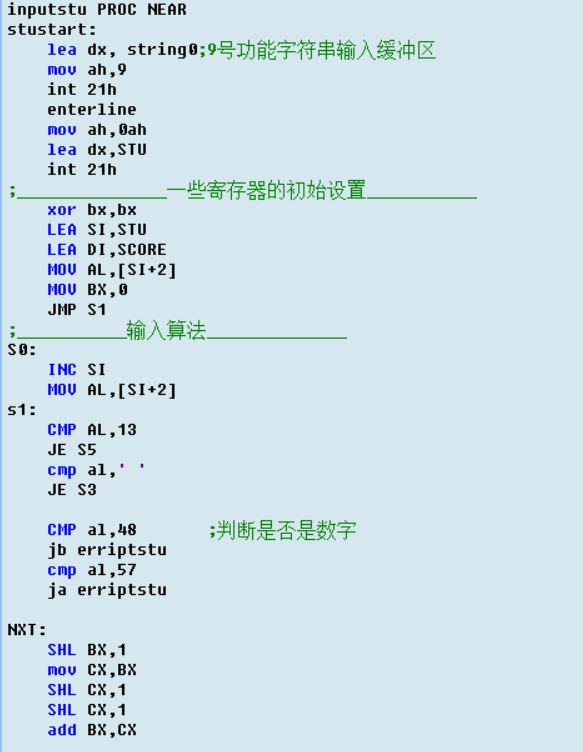


代码段的初始化

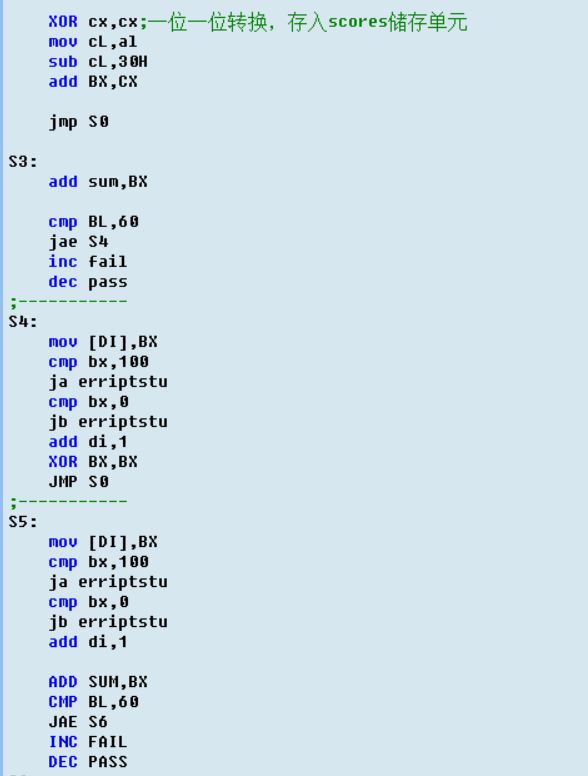
代码段主体



INPUTSTU子程序部分

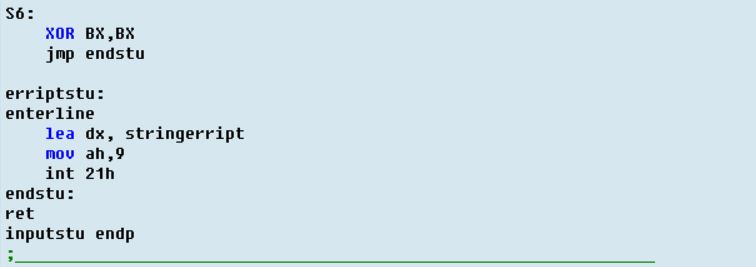


与回车、空格和1、9的比较



进入S3表示接收到了空格，可以完成一个成绩的计算，并将其加入sum、与60比较计算fail和pass，之后顺序进入S4，一个输入的成绩是否超过100的边界判断

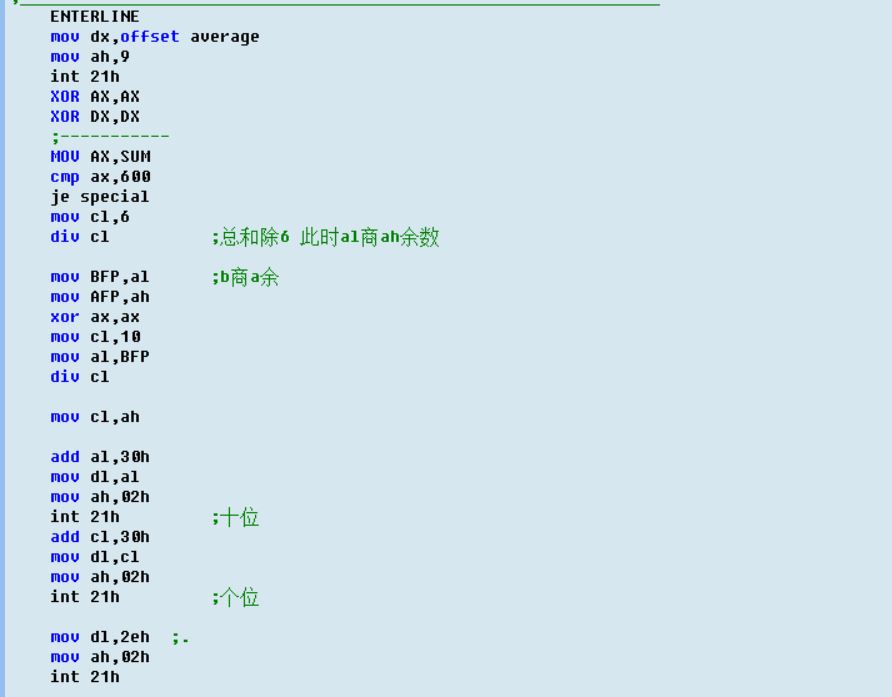
进入S5表示最后一个成绩可以完成计算，并且遇到了回车要结束输入了，与100比较进行边界判断，并将bx里的数加入sum、与60比较计算fail和pass，之后结束



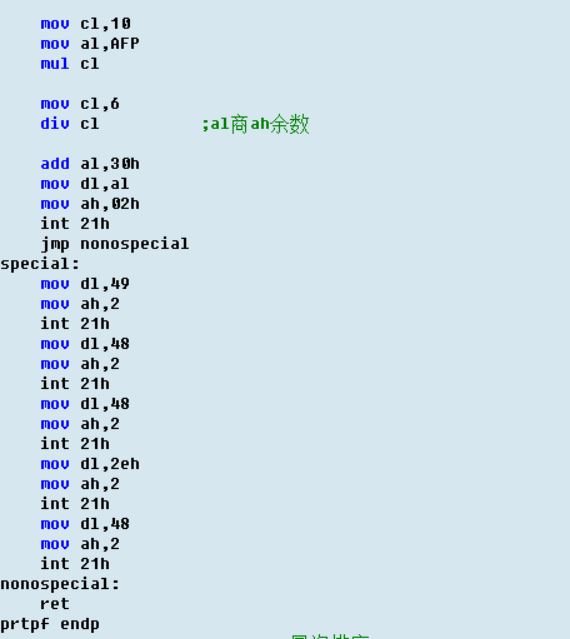
Prtpf子程序



开头是一个输出，因为INPUTSTU子程序已经完成了第二问的计算所以在这里直接输出

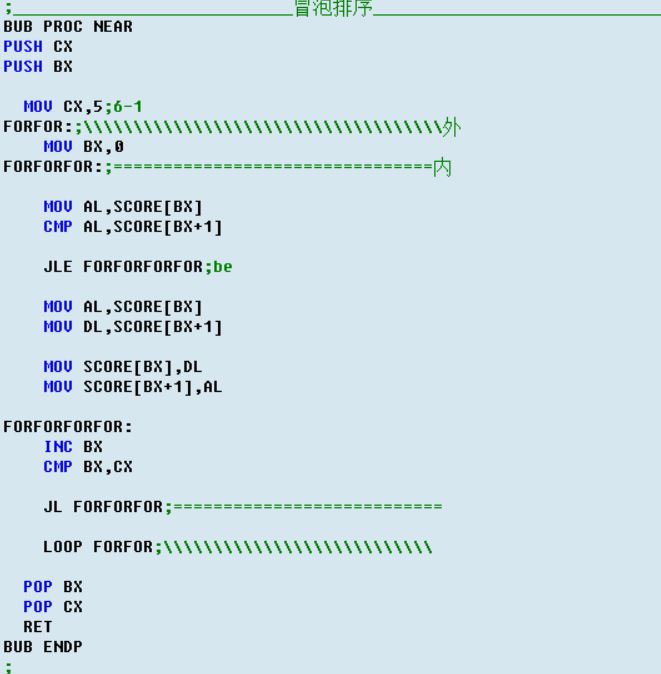


之后进入平均成绩的计算，输出提示后进行边界判断，之后总和除6，商BFP里是整数部分的数，余数AFP是小数部分要用到的值；正数部分正常除10打印十位个位，再打印一个小数点



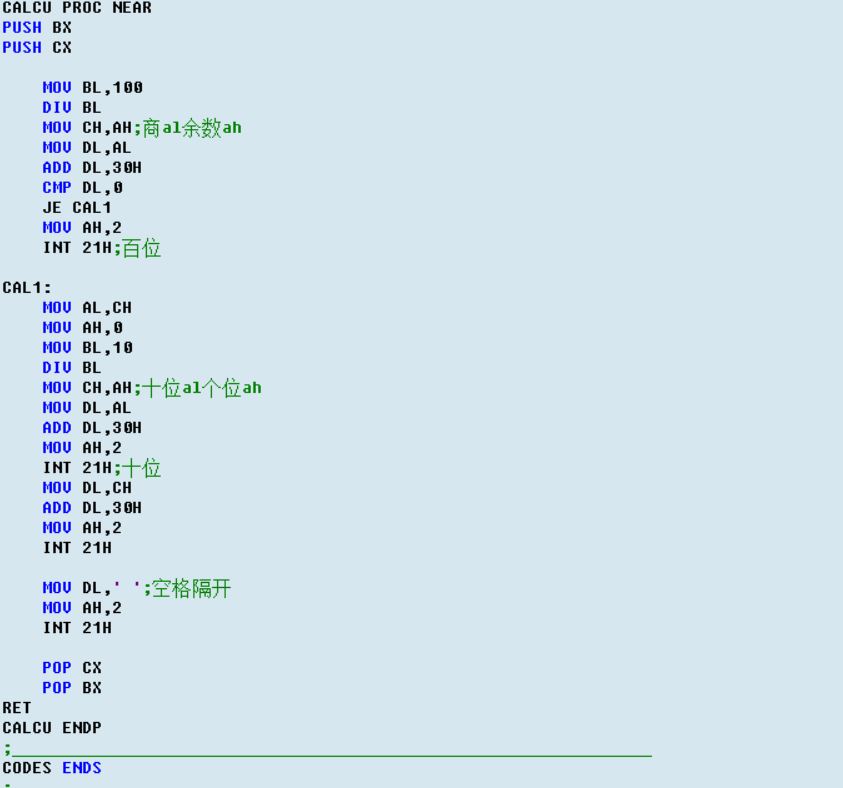
小数部分↑乘十除6获取小于1的小数点后第一位数，接着再加30h获得ascII码（没有实现四舍五入）

BUB子程序部分



简单经典的冒泡排序，双重循环，从每个数开始与后面的数两两比较，不满足前小后大就交换

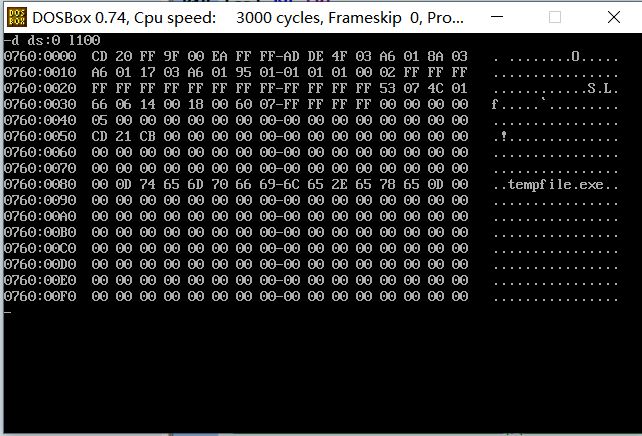
Calcu子程序



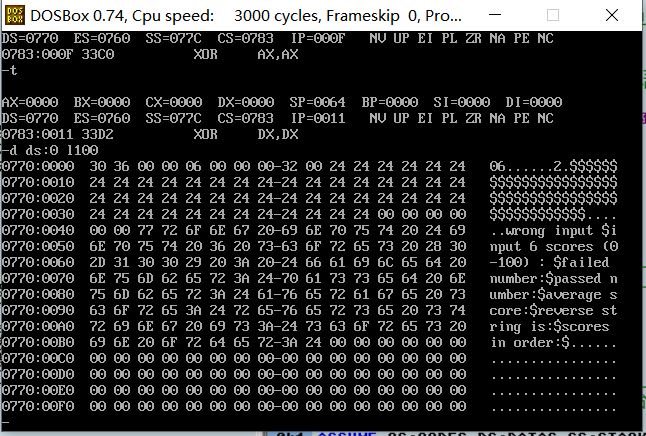
除100除10获得百位十位个位输出，很简单普通的输出

1. **调试debug：**

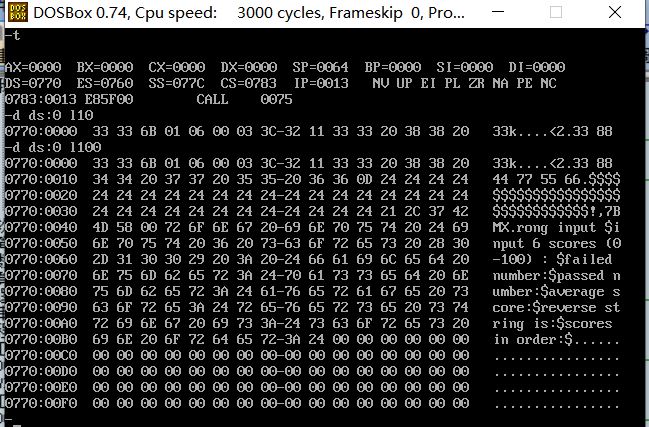
输入部分：



↑程序运行前



↑程序开始运行，子程序INPUTSTU调用之前，可以看到代码段初始的一些数据，比如fail是0 pass是6 等等



运行后

可以看到相应位置stu储存单元已经存入了输入的成绩字符串33 88 44 77 55 66，

以及pass和fail计算后都变成了33h（ASCII51即3），

pass和fail后就是sum的字，由00 00变成了6b 01即16bh（10进制363（33+44+55+66+77+88））

一堆$的stu后面6个字节是score，由......变成了‘！，7BMX’即排好序的10进制ASCII码‘33 44 55 66 77 88’

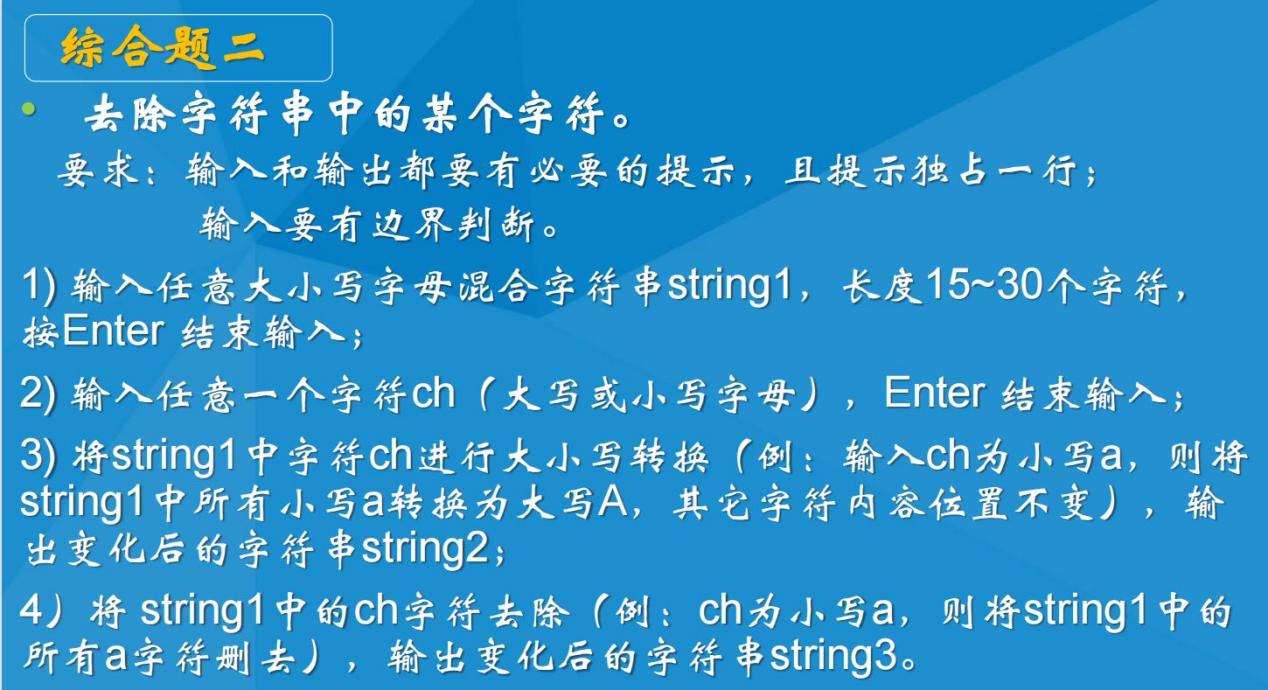
1. **心得体会：**

这一道题让我深刻地感受到写一道题之前清晰的思路和充分的准备的重要性。在题目要求逐渐复杂后，尤其是需要非常多比较、跳转，需要时常处理数字进行计算的时候，在别的地方把思路进程规划一下，再一点一点或者一个模块一个模块地完成封装，这样就会轻松很多；相反，如果直接一股脑写代码，甚至直接全写在代码段，那么写着写着就会遗忘、错失很多细节，比如比较的边界是哪里啊，跳来跳去哪个是干什么的等等，然后返回去改的时候又会导致其他地方也全要调整，这样就会增加很多工作量，更别说如果不是一次性写完，要写好几次的时候，就会遗忘更多，写之前还得先复习一遍之前写的。

这一道综合题让我第一次在汇编中编写了c语言中常见的循环，让我感觉到了汇编语言独特的魅力。虽然c语言写冒泡排序确实非常简单，但是汇编写出来的排序也让人格外有成就感。

另外这道题的代码写的时候遗失过两次，都是masm保存操作不熟悉，没有保存上就关掉了，导致重写了两次，虽然越写越熟练，但是心态都不好了，之后再写的时候我每次都专门把代码粘贴到额外的txt保存，顺便还留下了每个版本的代码的记录可以随时查找、返回，我认为这是一个新养成的好习惯，将来可以多尝试类似github的方式。

# 第二题

一、题目

**二、写代码前的构思：**

主要分为三个部分，输入，转换，输出

其中第一第二问在输入部分完成，第三问单独子程序CHANGE完成，第四问子程序DELETE中完成

第一问和第二问的输入均由21h中断的9号功能实现，将输入的字符串接受至提前开辟好的缓冲区储存单元中。在输入完成后利用9号功能自动计算储存在缓冲区偏移量1的字符数量来进行边界判断和输入正确判断。

第三问我们知道字母大小写的ASCII码之间差了20h，所以我们只要不做处理直接按原来顺序打印字符串中和输入的字符ch不同的其他字符，和输入字符ch相同的再区分大写字母+20h打印，小写字母-20H打印就行了

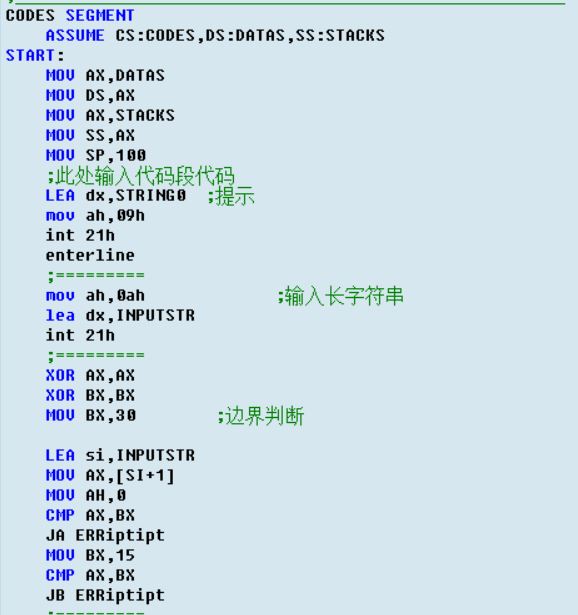
第四问我们直接一位一位打印字符串，打印的时候一位一位判断，只要和ch相同就跳过不打印，最后就得到了去掉ch的输出。

## Untitled三、流程图：

**四、程序源代码**

****

宏指令定义、代码段的储存单元以及栈段（未用到）

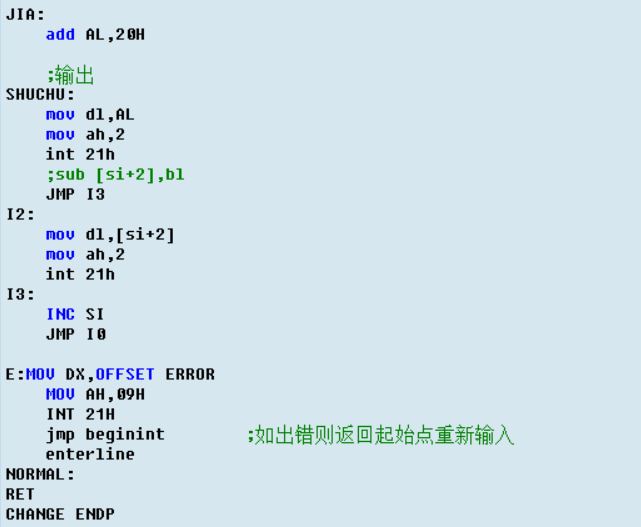
↑代码段初始化，提示打印、输入字符串及其边界判断

****

↑子程序的调用以及不同种类报错信息的打印

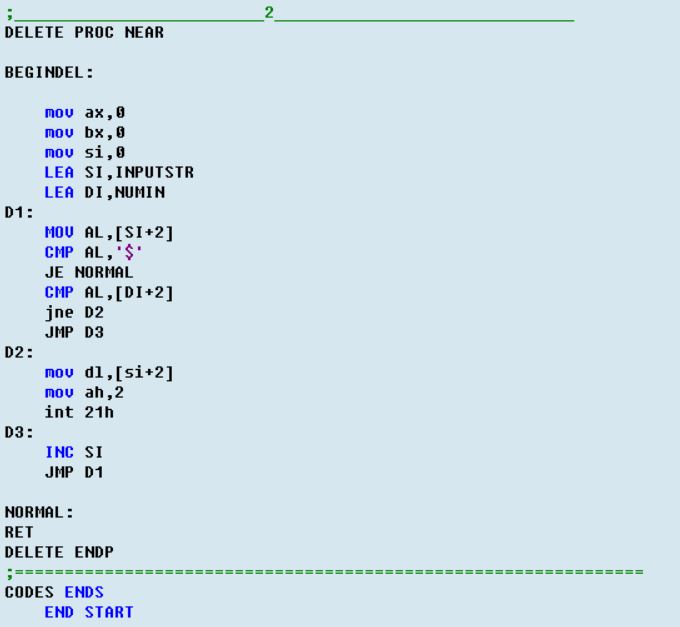
****

首先接收输入的ch进入numin，然后遍历字符串，在遇到$之前一位一位遍历，每次遍历都首先将当前字符与ch比较，不相等直接跳转进入I2打印，相等的话判断它是大写还是小写，分别跳转进入JIA和JIAN进行处理

****

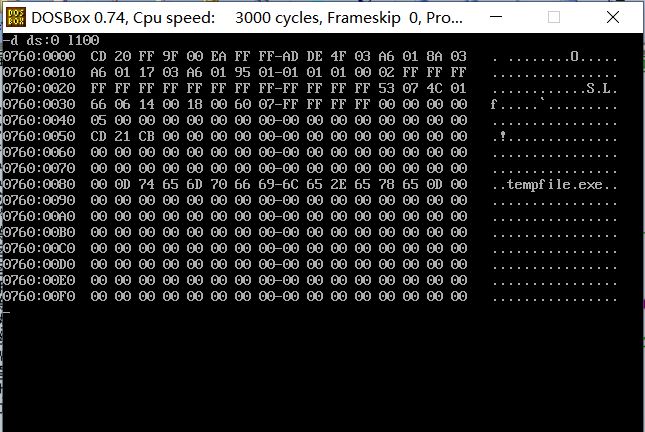
SHUCHU是打印和ch相同字符的输出位置，I2是打印ch不相同字符的输出位置

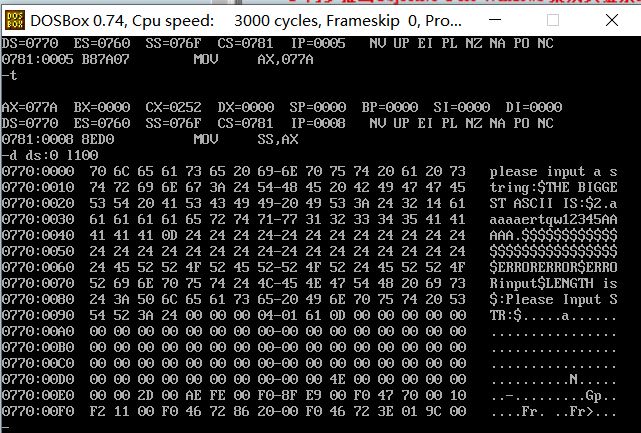
I3增加SI选到下一位字符后返回继续循环，直到遇到$标志遍历结束

****

和CHANGE相同的方法，遇到$结束标记之前一位一位遍历字符串，比较ch，遇到相同的跳过直接到下一位，其余打印

**五、调试debug**

****

****

上图为运行前后的储存单元变化，输入字符串为‘aaaaaertqw12345AAAAA’，字符ch为‘a’。这一道题较简单而且主要在输出前控制ax里面的内容来控制打印内容，感觉debug能看的不多

**六、心得体会**

这一题相比于第一题虽然简单，但是也有很多边界判断以及跳转，比如大小写那里，我一开始想的是分开精确判断大于‘A’小于‘Z’以及大于‘a’小于‘z’，避开其他字符尤其是‘Z’和‘a’中间那些字符，但是后来仔细一想发现不需要那么多比较那么复杂，只需要少数几次比较找出要跳转的特殊情况就可以了。

在老师检查的时候我也听了一下其他人的代码思路，发现这道题有很多不同的解法，包括不同的输入方式，不同的计算方式，输出方式相对来说比较大同小异，我这种直接打印需要打印的，跳过不需要打印的字符的写法只是其中一种算法，也让我认识到了开阔思路的重要性。