****

**汇编程序设计**

**学生成绩管理系统**

**题目要求**

一个学生的信息包括姓名、班级、学号、成绩，其中成绩需要精确到1位小数。实现以下功能：   
（1） 可以录入学生成绩（十进制形式）；   
（2） 可以按要求（如学号或成绩）进行排序显示；   
（3） 可以统计平均成绩；   
（4） 可以统计不及格成绩、60~70、70~80、80~90、90~100各分数段的人数。

**主要数据结构：**

姓名、班级、学号、成绩各用一个数组来保存，学生数小于等于100，相同下标的代表同一个学生的信息。   
姓名要求长度小于等于10个字符，班级和学号都是讲用户输入的ASCII码转十进制储存在一个字中。   
成绩处理稍微特殊一点，由于有一位小数，我在存的时候是把这个数扩大10倍再存的。及时没有小数也会乘以10，在输出的时候只要看最后一位是不是0就可以判断是不是整数了，小数就多输出一个小数点。在排序做比较和计算平均数，统计各分数段的时候这样也很方便，只需要在输出的时候除10并保留一位小数就行了。   
SORTED数组中保存的是排序后的下标，在排序之后学生信息的顺序还是录入时候的数据，这样避免了大量的数据移动。只是在写程序的时候容易犯错。

NAME\_ARR DB 100 DUP (10 DUP (?))

CLASS\_ARR DW 100 DUP (?)

ID\_ARR DW 100 DUP (?)

SCORE\_ARR DW 100 DUP (?)

SORTED DW 100 DUP (?);保存排序后的下标

主要使用的变量：

NUMBER DW ?;The number of students

COUNT0 DW 0;不及格

COUNT1 DW 0;60~69

COUNT2 DW 0;70~79

COUNT3 DW 0;80~89

COUNT4 DW 0;90~100

其他变量和数组

NOTICE DB 'Please input your choice: 1.Logging Data; 2.Sort and Output; 3.Get Average; 4. Statistic of Ranges; 5.Exit',0DH,0AH,'$'

ERR DB 'ERROR!',0DH,0AH,'$'

NOTICE1 DB 'Please input the NAME of the student:(ENTER for end)',0DH,0AH,'$'

NOTICE2 DB 'Please input the CLASS of the student:',0DH,0AH,'$'

NOTICE3 DB 'Please input the ID of the student:',0DH,0AH,'$'

NOTICE4 DB 'Please input the SCORE of the student:',0DH,0AH,'$'

NOTICE5 DB 'Do you want to log another one? Y/N',0DH,0AH,'$'

NOTICE6 DB 'CHOOSE: 1.Sort by scores;2. Sort by IDs',0DH,0AH,'$'

NOTICE7 DB '0~59:',0DH,0AH,'$'

NOTICE8 DB '60~69:',0DH,0AH,'$'

NOTICE9 DB '70~79:',0DH,0AH,'$'

NOTICE10 DB '80~89:',0DH,0AH,'$'

NOTICE11 DB '90~100:',0DH,0AH,'$'

TABLE DW CASE1,CASE2,CASE3,CASE4,CASE5,DEFAULT;用来当做switch使用

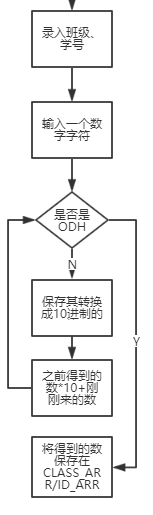
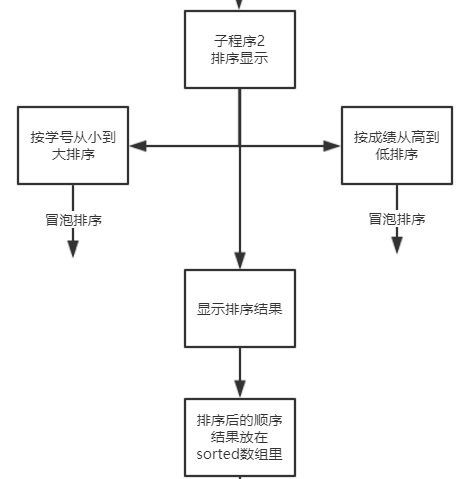
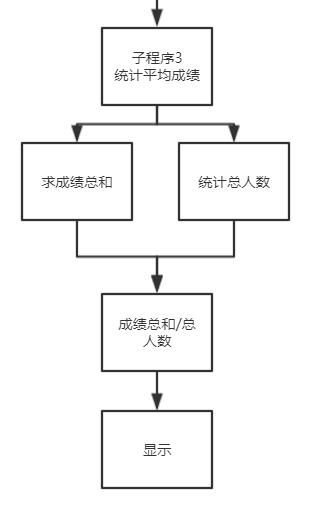
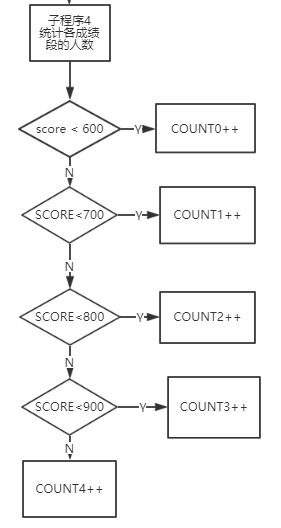
BUFFER DB 10 DUP (0),'$';用来辅助输出数字的ASCII码

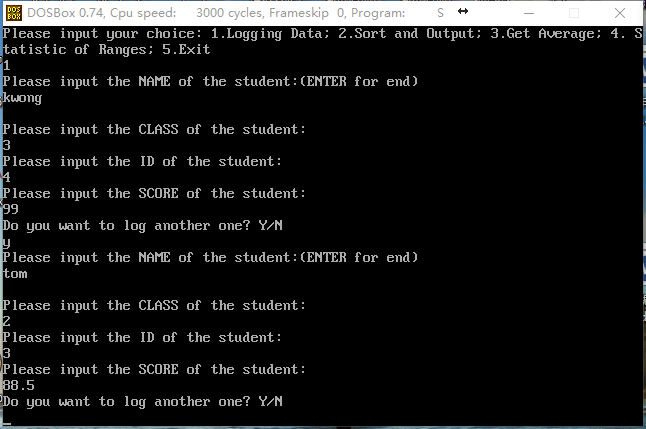
BUFREAR EQU OFFSET BUFFER+10

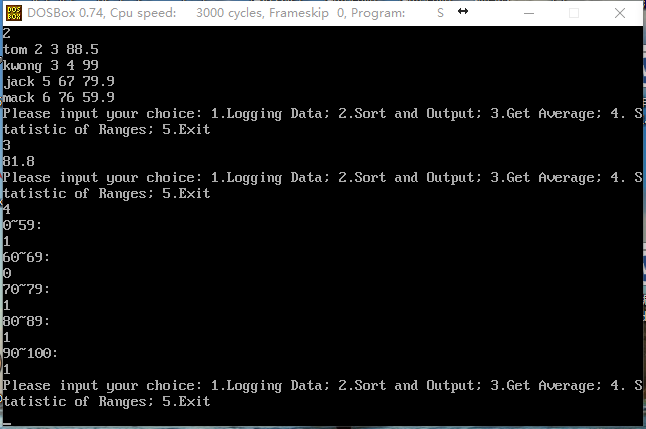
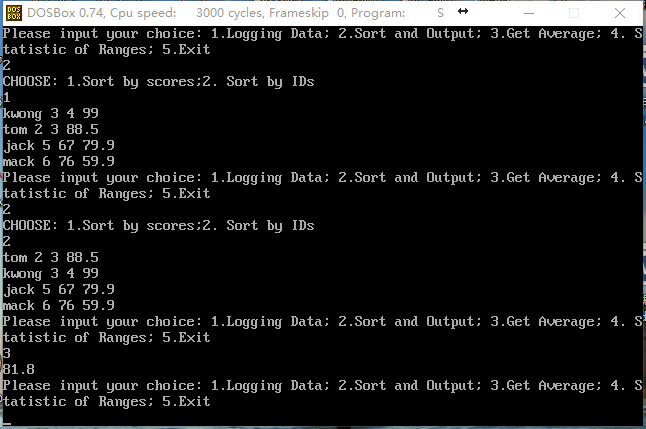
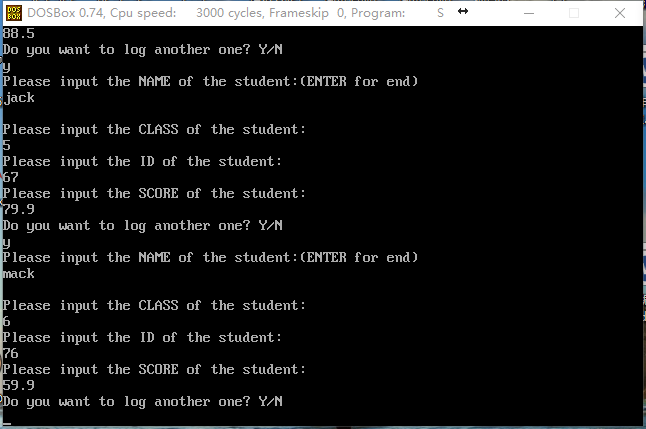
TYPE\_NAME DB 10

**程序结构说明**

START函数：主要是提示用户选择操作类型，   
1.Logging Data; 2.Sort and Output; 3.Get Average; 4. Statistic of Ranges; 5.Exit   
用户输入相应的数字即允许相应的子程序。   
子程序结构如下：

* 1：LOGDATA 输入学生信息。调用四个子程序分布录入姓名、班级、学号、成绩。录入后NUMBER增加1   
  + 1.1：LOGNAME 输入姓名，保存在NAME\_ARR中，用户输入完后在末尾加‘$’保存
  + 1.2：LOGCLASS 输入班级，保存在CLASS\_ARR中  
    
  + 1.3：LOGID 输入学号，保存在ID\_ARR中
  + 1.4：LOGSCORE 输入成绩，保存在SCORE\_ARR中,所有成绩都扩大10倍保存，这样一位小数也变成了整数，而原来是整数的数最低位为0，通过这一点区别整数和带一位小数的成绩
* 2：SORT 排序，选择按成绩排序还是按学号排序   
  
  + 2.1：SORTSCORE 按成绩排序，采用冒泡排序算法，排序结果保留在SORTED数组中
  + 2.2 : SORTID 按学号排序，冒泡排序，排序结果保留在SORTED数组中
  + 2.3 : SHOWSORTED 将排序后的结构输出，通过读取SORTED数组中的数字得到一个学生的信息输出。   
    - 2.3.1： NAMEOUT 输出姓名字符串
    - 2.3.2： DECOUT 十进制转ASCII码输出，用于输出学号或班级等十进制存储的数，此函数在所有输出十进制数时都会被其他子程序调用
    - 2.3.3： SCOREOUT 输出成绩，由于成绩有小数，所以需要特殊处理
* 3：GETAVERAGE 计算平均分数，SUM(SCORES)/NUMBER。输出调用SCOREOUT  
  
* 4：STATISTIC 统计各分数段人数，遍历各分数，先后与60，70，80，90比较确定区间  
  
* 5：其他辅助程序   
  + 5.1： CRLF 回车换行
  + 5.2： SPACE 空格
  + 5.3： DEBUG 调试使用，输出所有学生数据
  + 5.4： TAG 调试使用，就是在需要调试的地方输出一个‘！’

运行截图：   




**记录的BUG**

1. 函数没有全部出栈，无法RET
2. 循环 DIV CL时，记得把AH置0，否则永远除不尽，因为余数放在AH
3. 如下代码，INC 使得SI 加一，之后再左移，相当于乘2，所以SI的公式变成了 SI = (SI+1)\*2, {0,2,6,14,30,…..}

LEA BX, SORTED

XOR SI, SI

XOR DX, DX

LP0:

MOV [BX+SI], DX

INC SI

MOV DX, SI

SHL SI,1

LOOP LP0

1. 变量是 DW还是DB一定要牢记，MUL和DIV时注意8位与16位的区别，

MOV DX, 10

MOV AX, BX

MUL DL

DIV CX

程序会死掉！！！   
改成

MOV DX, 10

MOV AX, BX

MUL DX

DIV CX

**实验总结：**

通过此次汇编程序设计，让我发现了自己很多的不足，比如对于指令的不熟悉，对于CPU内部结构的不熟悉，以及一些指令需要注意的地方，比如MUL和DIV都是隐含寻址，操作数默认在AX中。还有存储中的字和字节的区别，在循环存储数据的过程中十分重要，如果在以字为单位的数组中每次地址只增加1，则会导致每次覆盖前一个写入的数据。

这次实验花费了大约两天的时间，编码时间其实也就半天，但是调试花了很多时间，首先是因为汇编程序调试不是很方便，其次自己对程序运行过程不是很熟，再之时程序太长自己没有足够的耐心一步一步的调试。后来慢慢掌握了一点调试的技巧才加快的进度。

这次实验让我对于汇编程序以及CPU的内部结构进行了更加深入的了解，也让我意识到细心的编码的重要性，好的编码习惯能够节省不少调试时间。